IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 18, 2019, accepted October 19, 2019, date of publication October 28, 2019,
date of current version November 13, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2949782

A Comprehensive Survey on Secure Outsourced Computation and Its Applications

YANG YANG[1,2,3,4,5], (Member, IEEE), XINDI HUANG[1,5], XIMENG LIU[1,5], (Member, IEEE), HONGJU CHENG[1], (Member, IEEE), JIAN WENG[3], XIANGYANG LUO[6], AND VICTOR CHANG[7]

[1]College of Mathematics and Computer Science, Fuzhou University, Fuzhou 350108, China
[2]Guangxi Key Laboratory of Cryptography and Information Security, Guangxi 541004, China
[3]Guangdong Provincial Key Laboratory of Data Security and Privacy Protection, Guangzhou 510632, China
[4]Fujian Provincial Key Laboratory of Information Processing and Intelligent Control, Minjiang University, Fuzhou 350108, China
[5]University Key Laboratory of Information Security of Network Systems, Fuzhou University, Fuzhou 350108, China
[6]State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou 450001, China
[7]School of Computing, Engineering and Digital Technologies, Teesside University, Middlesbrough TS1 3BX, U.K.

Corresponding author: Yang Yang (yang.yang.research@gmail.com)

This work was supported in part by National Natural Science Foundation of China under Grant 61872091, in part by the Guangxi Key Laboratory of Cryptography and Information Security under Grant GCIS201721, in part by the Opening Project of Guangdong Provincial Key Laboratory of Data Security and Privacy Protection under Grant 2019B030301004-13, and in part by the Open Fund Project of Fujian Provincial Key Laboratory of Information Processing and Intelligent Control, Minjiang University, under Grant MJUKF-IPIC201906.

ABSTRACT With the ever-increasing requirement of storage and computation resources, it is unrealistic for local devices (with limited sources) to implement large-scale data processing. Therefore, individuals or corporations incline to outsource their computation requirements to the cloud. However, data outsourcing brings security and privacy concerns to users when the cloud servers are not fully trusted. Recently, extensive research works are conducted, aiming at secure outsourcing schemes for diverse computational tasks via different technologies. In this survey, we provide a technical review and comparison of existing outsourcing schemes using diverse secure computation methods. Specifically, we begin the survey by describing security threats and requirements of secure outsourcing computation. Meanwhile, we introduce four secure techniques (i.e., secure multi-party computation, pseudorandom functions, software guard extensions, and perturbation approaches) and their related works. Then, we focus on the theories and evolution of homomorphic encryption, as well as the applications of the basic operations and application-specific tasks. Finally, we discuss the security and performance of existing works and give future directions in this field.

INDEX TERMS Secure outsourced computing, privacy preserving, homomorphic encryption, secure outsourced machine learning, data processing.

I. INTRODUCTION

Cloud computing, as a booming technology, delivers computing services (e.g., servers, storage, databases, analytic and intelligence) over the internet. With the rapid growth of data volume and computing scale, individuals or corporations are incapable or unwilling to afford the heavy storage or computation burdens. In this case, cloud service providers (CSPs) with almost unlimited storage space or computing power become an excellent choice. Using cloud computing technology, resources (e.g., CPU and storage) are available on-demand for users' terminals [1]. Generally, users are able to upload their large-scale data processing requirements to CSP, and then wait for the requested computational results, which could save abundant local memory space or computation overhead. Nowadays, cloud computing has found a tremendous number of fields for practical application, such as education [2], internet of things (IoT) [3], healthcare [4], and workflow scheduling [5]. A lot of large information technology (IT) companies are competing to build more powerful, stable, and reliable cloud service providers [6], such as Amazon Web Service (AWS), Microsoft Azure, Google Cloud, and AliCloud. Obviously, it is a win-win transaction for both the clients and the service providers.

Despite considerable benefits for the users, outsourcing computation still has an inevitable obstacle: the issue of

The associate editor coordinating the review of this manuscript and approving it for publication was Wei Yu.

security and privacy. The data outsourced to the cloud may be valuable or sensitive, and the users lose control of their data during the process of outsourcing computation. However, the server may be semi-trustful or even malicious in some cases, which means it may be motivated to deduce the users' private information for curiosity or profit. This is a huge threat to users' privacy, thereby becoming a non-negligible concern of outsourcing computation. For example, in a cloud-assisted e-healthcare system [7], large-scale historical medical data of patients and medical decision-analytic models are outsourced to the cloud platform (CP) for the follow-up functions of diagnosis and treatment. When a new patient consults the disease state, (s)he sends the symptom data to the cloud. Depending on the decision model, CP efficiently executes the diagnosis process and returns the results to the patient. Apparently, both the patient's symptom information and the final diagnosis results have high sensitivity, and are unacceptable to be disclosed to the CP or other unauthorized parties. Moreover, the historical medical data and medical decision-analytic models are considered as an asset of the providers, which are also required to be confidential, either to the CP or patients. Without adequate security and privacy protections, healthcare providers and patients will hesitate to adopt the outsourcing computation solutions. Besides, the correctness of the computation result is another challenge for secure outsourcing. It is possible that the server returns an incorrect result (for time saving, profits gaining, or due to malicious attacks), which makes the computation inaccurate or even misleads the users. Thus, the research on achieving data verifiability/checkability is a hot topic.

So far, many effective methods are proposed to address these concerns. Aiming at different application purposes, researchers continuously improve secure outsourcing schemes for various computation tasks. Security and efficiency are two important tools for evaluating secure computation schemes, which are always hard to satisfy at the same time. The tradeoff of security and efficiency is a commonly used method in many works. The technologies used for privacy-preserving outsourcing computation are continuously optimized or combined for higher security levels or better performance. Among them, secure multi-party computation (MPC) [8] and perturbation operations [9] are competent algorithms for secure outsourcing, which provide satisfactory performance in many application scenarios. Nevertheless, a majority of these works achieve a relatively low-security level, which is unacceptable for specific computation tasks with high privacy requirements. Instead, this survey focuses on a powerful cryptographic tool: homomorphic encryption (HE) technique, which has the advantage of high security and privacy level. It is well known that, in many cases, the overhead of HE-based outsourcing computation is comparatively higher than the other methods, especially the utilization of fully homomorphic encryption (FHE). However, with the improvement of HE-based algorithms, many HE-based schemes for large-scale computations show good performance, which are expected to be applied in the near future.

In this survey, we mainly introduce HE-based research achievements for secure outsourcing computations. We consider two aspects: secure outsourcing of fundamental functions for dealing with common mathematical operations; the other one, based on the former, is for application-oriented secure computation outsourcing, where many practical applications like machine learning or biometric computations are securely executed by cloud. In summary, this survey aims to provide a particular perspective for the overview of the existing secure outsourcing solutions based on HE algorithms. To evaluate these schemes, we compare their security and efficiency performances. The design framework of outsourced computation and the applications are summarized in Fig. 1, where we focus on secure computation outsourcing, secure technologies, homomorphic encryption, outsourced fundamental functions, outsourced application-specific tasks, security and efficiency comparison, and future research directions.

The remainder of this paper is organized as follows. In Section II, we describe the constructions and critical points of secure computation outsourcing, including system models, security threats, and general requirements. In addition, we compare our survey with several other surveys with similar topics. Next, several commonly used technologies for secure computations are introduced in Section III. Then, in Section IV, we discuss the classifications, evolutions, and related works of HE technique. Section V and Section VI cover the secure outsourcing computations for fundamental and application-oriented algorithms, respectively. In Section VII, we analyze the security level as well as the performance of HE-based secure outsourcing schemes. In Section VIII, we further present open issues and challenges for secure outsourcing computation. Finally, we draw our conclusions in Section IX.

II. SECURE COMPUTATION OUTSOURCING

In this section, we first describe the general system models for secure computation outsourcing in a lot of related works. Then, we show common security threats and two general requirements of the outsourcing computation. Finally, we compare related surveys (similar but having different emphases) with our survey. For ease of reference, Table 1 lists definitions of abbreviations and notations used in Tables throughout this survey.

A. SYSTEM MODEL

In general, the computation outsourcing system mainly contains three entities: user (data owner or request user) and a cloud platform (CP) and cloud service provider (CSP) [10], [11]. Due to the constrained capability of personal devices, the client outsources the burdensome computational task to the cloud. For security concern, the client first protects the privacy of his computational task through encryption algorithm and then uploads the encrypted request to the CP.

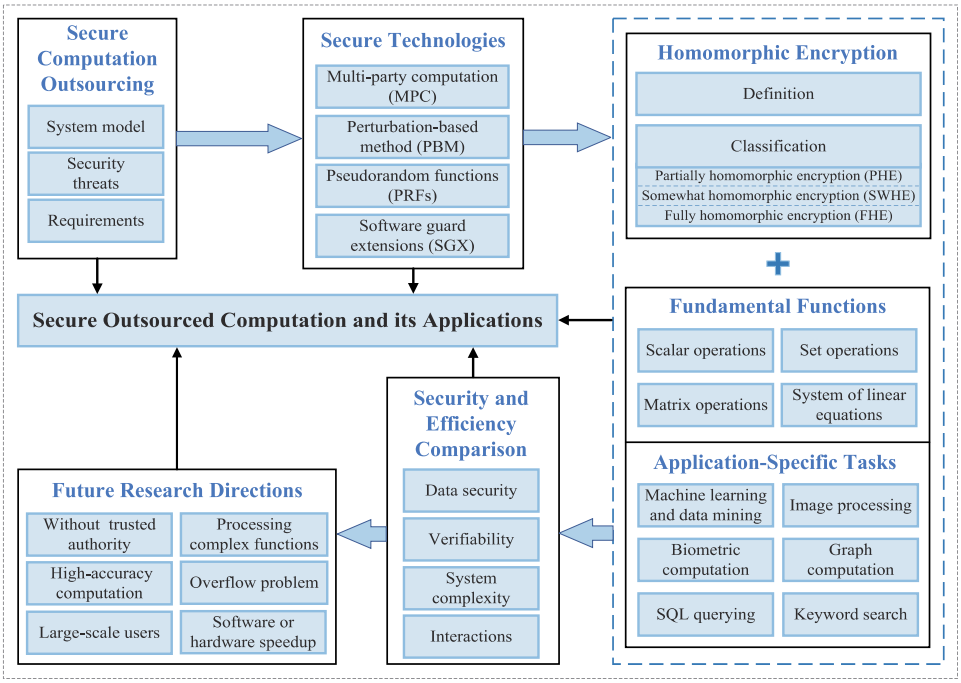


FIGURE 1. Road map of secure outsourced computation and its applications.

TABLE 1. Abbreviations and notations used in Tables.

Abbreviation	Definition	Notation	Definition
HE	homomorphic encryption	k	the size of an input set
PHE	partially homomorphic encryption	λ	security parameter of the adopted OT protocol
SWHE	somewhat homomorphic encryption	m	the dimension of a square input matrix
FHE	fully homomorphic encryption	m_1	the row dimension of the first non-square matrix
SS	secret sharing	m_2	the column dimension of the first non-square matrix
PBM	perturbation-based method	m_3	the column dimension of the second non-square matrix
GCs	garbled circuits	l	security parameter of the adopted HE scheme
MPC	multi-party computation	r	the batch number of inner products in a single ciphertext
SE	standard symmetric encryption	p	the iteration number of protocol
OT	oblivious transfer	q	the number of Map slots
PRFs	pseudorandom functions		
PRPs	pseudorandom permutations		
ORE	order-revealing encryption		
OPE	order-preserving encryption		
DP	differential privacy		

After that, the CSP interacts with the cloud storage to execute the required computational task. The final result, also in the encrypted domain, is returned to the client. Using the decryption key, the client recovers the result. Note that handling a computation task may need several rounds of interactions between the client and the CP.

The model above only considers a simple situation of computation outsourcing systems. In practical applications, the system model can be much more complicated: except for the request user, the model may also contain data owner(s) for providing the original data. For example, in an online medical system, diagnosis should be made based on historical medical data, which are provided by numerous data owners. On the other hand, the model may involve multiple (typically two) cloud service providers to solve the problem independently or jointly. Moreover, third types of authorities (e.g., attribute authority (AA), third-party auditor (TPA), and key generation center (KGC)) may be introduced to distribute secret keys, or implement other auxiliary functions. The system architecture is shown in Fig. 2.

B. SECURITY THREATS

Although providing a great convenience for users, outsourcing computation is still not a completely-satisfying technique. The primary reason is the appearance of security and privacy concerns during data treatments, such as the reveal of data contents or user privacy. Thus, we should identify and solve the security threats of the outsourcing computation, thereby minimizing the risks of destroying data security and privacy.

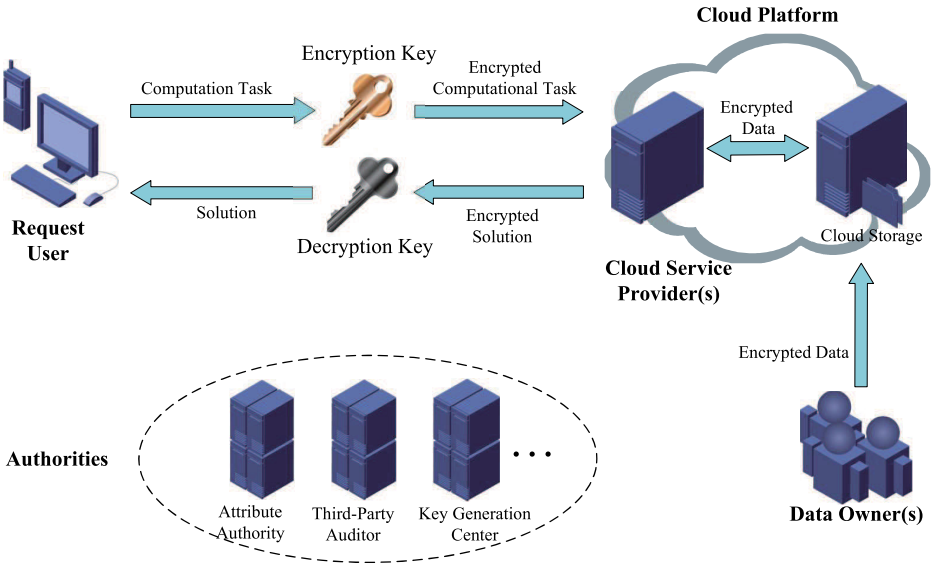


FIGURE 2. System model.

Generally, two types of threats are considered in the environment of cloud computing. One is from external attackers, including the threats of remote software or hardware attacks. The attackers utilize different kinds of techniques (e.g., network eavesdropping or malware attacks) to access unauthorized data or intrude cloud servers. The other security threat is from internal participants. After a user submits his computational task, the data and computational process will be out of his control. So the ''morality'' of cloud servers plays a vital role in data security and privacy. However, most cloud servers are unable to be deemed as trustful, which means the servers may solve the computational tasks incorrectly or try to learn what should not be known. Depending on the server's behavior, people usually classify the adversarial models into two levels: the first is called ''honest-but-curious'' or ''semi-honest'' model and the second is ''malicious'' model. In the honest-but-curious model, servers execute the operations complying with the required computational processes, yet still curious about the sensitive information of users. On the other hand, apart from the curiosity of users' privacy, malicious servers may go against the requested computations and return incorrect results to save their computing power or achieve other intentions for benefits. These two threats (i.e., external and internal attacks) significantly destroys data confidentiality and integrity of users.

C. REQUIREMENTS FOR SECURE COMPUTATION OUTSOURCING

In this subsection, we discuss the requirements for a powerful, or say satisfying, secure computation outsourcing system, which can be roughly divided into two types: security and efficiency requirements. The former, directed against the security threats mentioned in Subsection II-B, is intended to protect the security and privacy of users' data. Note that the security of information from servers is also a significant potential issue in many computation outsourcing schemes, yet we will not discuss much in this survey. The efficiency requirement refers to the cost of computation and communication for completing the outsourced tasks.

The evaluation of outsourcing computation security is made up of three main factors, which are data confidentiality, data integrity, and data access controllability. As mentioned before, data confidentiality refers to the ability to prevent users' data from revealing to the cloud or unauthorized parties. And the concept of data integrity means guaranteeing data completeness and correctness, which is also called data verifiability/checkability. Access controllability is the property that authorizes the access permission only to valid users, which restricts the group of users to obtain the computational results or defines the permitted data sources from the servers or data owners. This kind of fine-grained access strategy provides a flexible access control (like in [12] and [13]), thus having been of great significance both in the theory and reality.

Another essential requirement for outsourcing computation is the efficiency performance, particularly in the HE-based schemes. It is just because the users are intolerant of high computation cost that they choose to outsource their heavy tasks to the cloud. For this reason, designing high-efficiency outsourcing schemes is beneficial to both users and service providers. The major factors to measure the efficiency are the overhead of computations before/during/after the outsourced processes (including data encryption/decryption, processing, verification, etc.) and the communication cost during data transmission.

However, in an outsourcing computation scheme, achieving both requirements for great effects is always impractical. In general, better security requires additional operations to support, which naturally reduces the efficiency.

Conversely, an excessive pursuit of high efficiency often leads to insufficient security guarantees. Thus, a feasible, or rather say a good-performance, computation scheme is always the one realizing best tradeoffs between the security and the efficiency. It has also been an inspiration to design the algorithms in the research.

D. COMPARISON WITH OTHER SURVEYS

We go through other related surveys regarding secure outsourcing computation or especially the ones based on HE technique, and compare them with ours. The security issues and their countermeasures of outsourcing computation were investigated in [14]–[16]. And the works like [17]–[20] presented a detailed overview of HE technique, from its evolutions, classifications to implements and more. While for practical outsourced applications, no or very few specific schemes were referred to in the surveys [17]–[20]. Scheme [21] and [22] mainly introduced verifiable computing techniques and some related practical designs. In [23] and [24], secure outsourcing schemes for a limited number of functions (e.g., scientific computations) were discussed, and these surveys consider different techniques including HE algorithms. Another survey [25] focused on secure outsourcing schemes based on HE technology, still just covering certain computational tasks. Like our survey, [26] provided a wide scope of outsourced schemes for specific computational tasks, including fundamental and application-specific functions. The difference is that we collect and analyze the schemes based on HE schemes, while in the survey [26] this line of research was not the emphasis.

III. PRACTICAL TECHNOLOGIES FOR SECURE OUTSOURCING COMPUTATION

Several practical secure technologies have been raised and widely applied to protect sensitive information in many outsourcing schemes. In this section, we will list four secure techniques: secure multi-party computation, pseudo-random functions, software guard extensions, and perturbation approaches. We briefly explain their theories, progresses, and practical applications. Since the secure outsourcing computation based on homomorphic encryption is the main research contents in this survey, we will later introduce HE technique separately in Section IV.

A. SECURE MULTI-PARTY COMPUTATION

One solution for secure computation across different parties is called multi-party computation (MPC) in which multiple parties jointly compute a function over their inputs while keeping individual input private (the detailed definitions were referred to in [8]). MPC is a popular topic in cryptography, which has been studied for more than twenty years since Yao's millionaire protocol [27]. Yao's millionaire problem describes such a scenario: supposing there are two numbers a (for Alice) and b (for Bob), the goal is to get the relationship between these two values without revealing the actual values to the counterparts. The idea of garbled-circuit (GC) based MPC was first described by Yao [28] to generate a general function of MPC. In the design, one party A first creates a ''garbled circuit'', and sends the circuit to the other party B. Then B evaluates the circuit with his inputs and returns the result to A. By exchanging some information, both parties will know the result but no information about the other side. Unfortunately, his paper did not provide the details on how to construct this general circuit. Later, highly efficient garbled-circuit techniques were designed in the two-party case for saving the running time and memory space [29], [30]. Despite the success of the two-party case, multi-party secure computation progress has been much slower than the two-party setting. Beaver, Micali, and Rogaway [31] designed a construction for constant-round multiparty secure function evaluation (consisting of n parties, $n \geq 2$, each of whom possesses a private input x_i, $1 \leq i \leq n$). The n parties want to collaboratively evaluate a function $f(x_1, \ldots, x_n)$ without revealing their private values. Ben-Efraim *et al.* [32] showed that via a multiparty garbled circuit, the constant-round secure multiparty computation can be achieved with good performance for the case of semi-honest adversaries. Later, the work presented a new way of constructing a garbled circuit that can be evaluated with only a constant number of operations per gate for a large number of parties. Due to the natural characters of secure data processing across different parties, a large number of references use garbled-circuit method to design protocols for real-world applications, such as biometric identification [33], private linear branching programs [34], privacy-preserving remote diagnosis [35], and face recognition [36]. However, these schemes still suffer from very high computation and multiple round communication complexities [37].

Another approach to MPC is the secret-sharing based protocol which generates random shares using secret sharing (SS) technique and distributes the shares to different parties, and the parties jointly compute objective functions interactively. Secret sharing (first designed by Shamir [38] and Blakley *et al.* [39]) enables a division and a distribution of confidential information to a certain number of shareholders, where the decryption can be performed jointly only if enough parties are gathered. Ben-Or *et al.* [40] and Chaum *et al.* [41] proposed protocols for securely evaluating any function. They both designed solutions for computing addition and multiplication (*XOR* and *AND*) on values in (verifiable) secret shared form, and with the results remaining secret shared. As these primitives are complete, any function can be evaluated gate by gate. General secret sharing based MPC protocols tend to be less efficient than special-purpose protocols for two reasons. First, the circuits are generally quite large. Second, the multiplication sub-protocol is rather inefficient as it requires substantial interactions. Thus, a line of research has focused on developing efficient MPC protocols for specific functions. Damgård *et al.* [42] presented generic protocols for comparison, equality test, and bit-decomposition operations based on common secret sharing mechanisms. Later, Nishide and Ohta [43] constructed more efficient protocols for the tests of interval, equality, and comparison of

shared secrets without relying on bit-decomposition protocol (though it seems essential to such bit-oriented operations). More efficient MPC protocols for specific functions were constructed, such as secure multi-party product [44], scalar product [45], sorting [46], matrix factorization [47], and set intersection [48]. These protocols have been used for privacy-preserving multi-party data mining [49], cooperative scientific computations [50], database query [51], geometric computation [52], etc.

Although the secret-sharing based MPC is promising, it requires multiple parties to store certain redundant data and requires pairwise secure channels between servers. Very recently, fully homomorphic encryption (FHE) [53], [54] has been used to reduce the round complexity (e.g, reduced to two-round) in MPC [55], [56]. Unfortunately, one of the biggest drawbacks of fully homomorphic cryptosystems is the system complexity. The other approach is to use indistinguishability obfuscation (IO) [57], [58] to achieve two-round MPC protocols [59], [60]. Although IO has the power to broaden the scope of cryptography dramatically, how to construct practical IO is still an open research problem.

There concludes to several drawbacks of the solutions of MPC. The first is the online requirement: all the parties are required to be online simultaneously while performing the secure MPC. The second is the multiple communication rounds: MPC requires at least two rounds of interaction for each party for each function. The third one is the local storage overhead: each party needs to store its own data (even needs to store other parties' shares in the secret-sharing based MPC). The last is about the local computation overhead: all the parties are required to pre-process the data before the computations. For example, the parties should first use random numbers to randomize the data for secret sharing or ''garbled'' the circuit before performing the secure computation.

B. PSEUDORANDOM FUNCTIONS

Pseudorandom function (PRF) technique [61] is another practical cryptographic method for secure computation in the delegation setting. In fact, the notation PRF refers to a pseudo-random function family with the definition as:

Definition 1: Let $F : \{0,1\}^\kappa \times \{0,1\}^m \rightarrow \{0,1\}^l$ be a family of efficient, keyed functions. For $k \in \{0,1\}^\kappa$, the function $f_k : \{0,1\}^m \rightarrow \{0,1\}^l$ is defined as $f_k(x) = F(k,x)$. We say F is a pseudorandom function (PRF) if for every probabilistic polynomial time (PPT) adversary $\mathcal{A}$, $|Pr[\mathcal{A}^{F(k,\cdot)}(1^\kappa) = 1] - Pr[\mathcal{A}^{f_\kappa}(1^\kappa) = 1]|$ is negligible, where k is uniformly chosen from $\{0,1\}^\kappa$ and f_κ is randomly chosen from all possible functions mapping from m-bit strings to l-bit strings.

In other words, a PRF is a computable keyed function $f_k(\cdot)$ ($k \in \{0,1\}^\kappa$), whose values are indistinguishable from random values in the defined function range.

The oblivious PRF (or OPRF) [62] is defined as an evaluation functionality $\mathcal{F}_{OPRF} : (k,x) \rightarrow (\perp, f_k(x))$, corresponding to a PRF function $f_k(\cdot)$. Using oblivious pseudo-random functions, the client evaluates a keyed, pseudo-random function on his input with the server holding the key. Finally, the client obtains the result obliviously while the server will know nothing.

PRF technique has been used for secure computation in many proposed outsourcing schemes. Benabbas *et al.* [63] used closed-form efficient PRFs to achieve verifiable delegation of polynomials and database queries efficiently. Fiore and Gennaro [64] proposed an extending scheme for polynomial evaluation and matrix-vector multiplication with an improved PRF algorithm. Wang *et al.* [65] employed a random transformation based approach where all the matrices and vectors for sampling and secret transformation are generated by a PRF with random seeds. The sharing of the private matrices and vectors is simplified to shares of the random seeds. Kamara *et al.* [66] introduced a verifiable delegated private set intersection (PSI) scheme, secure against several adversarial models. In the protocol, clients jointly generate the key to PRF. We can refer to more PRF-based works for different application domains, such as [67], [68] for linear algebra, [69] for pattern matching, and [70] for data search. Moreover, oblivious PRF has also attracted the attention of researchers. For example, based on the oblivious PRF algorithm, Freedman *et al.* [62] described a secure keyword search scheme, and Hazay and Lindell [71] designed a protocol for outsourcing private set intersection function.

In addition to PRF and OPRF, another member of the pseudo-random family is pseudo-random permutation (PRP), which is an indistinguishably secure encryption scheme of permutations over operating domains $\{0,1\}^m$. The function is bijective where the input domain is equivalent to the output domain. We can also find several practical applications based on PRPs. Evdokimov and Günther [72] designed an efficient scheme supporting secure operations on the outsourced database. In the work, the search query is protected using pseudo-random permutations. Ma *et al.* [73] realized secure outsourcing modular exponentiation computations by the randomization ability of the PRP technique. The graph encryption schemes in [74] employed the PRP and other cryptographic algorithms to support secret approximate shortest distance queries. The graph encryption schemes are provably secure under the semi-honest model.

C. SOFTWARE GUARD EXTENSIONS

Recently released, software guard extension (SGX) [75], [76] is a security extension of Intel processor technique. Unlike software defense, SGX implements a trusted execution environment combining both secure hardware and software. In the design of SGX, a protected memory container, called enclave, is reserved as a trusted execution environment (TEE) to hide sensitive information (e.g., secret codes or data) from privileged modules like operating system (OS), virtual machine (VM) scheduler or management engine (ME). Even the root or high-priority programs cannot access or modify the contents inside the enclave. The data is available in plaintext within the enclave module and is protected when written to the system memory (i.e., RAM). SGX also provides a remote attestation mechanism with proof

to verify the integrity of the newly generated enclave by remote entities. The efforts of SGX guarantee the confidentiality and integrity of sensitive data and computations on an untrusted cloud, even though other system parts are attacked or compromised.

SGX technique can be applied to secure outsourcing frameworks, where the computations are executed securely inside the TEE. Chen *et al.* [77] proposed a genetic testing framework which leverages SGX to achieve secure storage and computation on an untrusted cloud. Before data outsourcing, the data owner and the enclave first conduct an attestation procedure to prove their integrities and authenticities. Then the data is encrypted and sent to the cloud server along with a message authentication code (MAC). The enclave seals the sensitive data for answering further queries from data users. When a user queries a command, he will also attest the remote enclave and establish a secure channel with the enclave. Receiving the query, the data within the enclave will be unsealed for query operations. Finally, the encrypted result will be sent back to the authorized data user. Throughout the process, the security and integrity of the data are ensured. A multi-party machine learning scheme [78] was achieved based on trusted SGX-processors. In the design, each party independently establishes a secure channel with the enclave and sends encrypted data to the enclave. The enclave runs target functions on the whole data set securely and returns the encrypted result to each party. Scheme [79] ran a secure SGX-based MapReduce algorithm [80]. Only the core algorithms are running inside the enclave, thereby minimizing the performance overhead. With a combination of HE and SGX techniques, Sadat *et al.* [81] presented a solution for secure genome-wide association studies (GWAS). The data is encrypted using Paillier cryptosystem [82] and then put into statistical tests within a secure enclave. Besides these works, more SGX-based schemes are popular in cloud computing and applied in practical fields, like healthcare [83], machine learning [84], [85], data analysis [86], location-based services [87], and many others.

Providing an alternative solution for achieving privacy-preserving and verifiability in the outsourcing schemes, Intel SGX, however, is known to be vulnerable to certain software and physical attacks. For example, the host OS may be controlled by an adversary, thereby possibly leaking sensitive data from side-channels [88] (including the attacks like cache attack [89], [90], branch shadowing attack [91], controlled-channel attack [92], etc.). Besides, the compromised OS may launch DoS (denial of service) attack to disrupt the functions of the enclave. Except for the security threats, integrity property cannot always be guaranteed, like in the event of system shutdown [93]. In addition to these defects, some problems of insufficient performance (for instance, due to a limited enclave page cache) are also to be resolved. Although several secure SGX-based solutions have been designed to settle or partially settle the problems of security and performance. More "perfect" SGX-based schemes for different applications are required in the future.

D. PERTURBATION-BASED APPROACHES

Data perturbation is another major technique for preserving privacy in outsourcing computation. Generally, by performing linear algebra operations or other conversion operations, data owners transform the data in certain ways to distinguish and conceal the original information and then outsource the perturbed data to the server. The perturbation methods include swapping values between records [94], [95], randomization (e.g., adding noise [96]), geometric perturbation [97], rotational perturbation [98], [99], replacing the original database by a sample from the same distribution [100], [101], etc.

We first introduce three fundamental perturbation-based approaches for protecting private matrices: matrix addition, matrix multiplication, and matrix's row and column permutations. Suppose a private matrix $\mathbf{M} \in \mathbb{R}^{m\times n}$ whose elements should be hidden. Matrix addition is to perform an addition between the matrix $\mathbf{M}$ and a randomly generated matrix $\mathbf{L} \in \mathbb{R}^{m\times n}$:

$$\mathbf{M}_1 = \mathbf{M} + \mathbf{L}$$

The original matrix $\mathbf{M}$ cannot be recovered when the random matrix $\mathbf{L}$ is unknown to the observers. Note that if the matrix after the transformation has a dense structure, a large number of unnecessary computations will be introduced. Thus, finding a practical random matrix that leads to both privacy and sparsity is rather significant. Choosing a diagonal matrix $\mathbf{A} \in \mathbb{R}^{m\times m}$ where the non-zero elements are generated by a pseudorandom function and another diagonal matrix $\mathbf{B} \in \mathbb{R}^{n\times n}$ with the random non-zeros positive, matrix multiplication-perturbation method can be computed as:

$$\mathbf{M}_2 = \mathbf{AMB}$$

However, during this transformation, zeros of the original matrix are still retained. To further hide the structure of the original matrix (i.e., the positions of the elements), matrix permutation was introduced. Matrix permutation disrupts the order of elements in the private matrix by randomly permuting the rows and columns of $\mathbf{M}$, expressed as:

$$\mathbf{M}_3 = \mathbf{DME}$$

where $\mathbf{M}_3 \in \mathbb{R}^{m\times n}$ is the permuted matrix, and $\mathbf{D} \in \mathbb{R}^{m\times m}$ and $\mathbf{E} \in \mathbb{R}^{n\times n}$ are the pseudorandom orthogonal permutation matrices. The user recovers the original matrix by performing:

$$\mathbf{M} = \mathbf{D}^{\mathrm{T}}\mathbf{M}_3\mathbf{E}^{\mathrm{T}}$$

where $\mathbf{D}^{\mathrm{T}}$ is the transposed form of $\mathbf{D}$, and $\mathbf{E}^{\mathrm{T}}$ is alike. The formula holds with the orthogonality property of the permutation matrices, which is $\mathbf{D}^{\mathrm{T}}\mathbf{D} = \mathbf{I}$ and $\mathbf{E}^{\mathrm{T}}\mathbf{E} = \mathbf{I}$ ($\mathbf{I}$ is the identity matrix). Moreover, combining both matrix multiplication and matrix permutation, the transformation can be formed as follows:

$$\mathbf{M}_4 = \mathbf{PMQ}$$

where $\mathbf{P} = \mathbf{DA}$ and $\mathbf{Q} = \mathbf{BE}$.

Matrix perturbation technique is widely used in many practical outsourcing computations. We illustrate with several proposed schemes. Duan *et al.* [102] proposed a secure and verifiable outsourcing scheme for nonnegative matrix factorization. The input matrix is obscured by performing permutation operations, and the two permutation matrices are generated by Knuth shuffle algorithm [103]. Yang *et al.* [104] employed a retrievable data perturbation method for privacy-preserving cloud computing. In the work, private data is protected by adding a noise matrix with the property that the perturbed data have the same mean and covariance as the original one. Lin [105] constructed a privacy-preserving kernel k-means clustering outsourcing scheme. To encrypt a set of vector instances $\mathbf{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m\}$, a linear transformation for all instances is performed as: $\boldsymbol{z}_i = \mathbf{M}\boldsymbol{x}_i (i = 1, \ldots, m)$, where $\mathbf{M} \in \mathbb{R}^{m \times m}$ is an invertible random matrix.

Geometric data perturbation (GDP) is a combining technique including multiplicative transformation ($\mathbf{R}$), translational transformation (Ψ) and noise additive operation (Δ):

$$G(\mathbf{X}) = \mathbf{RX} + \Psi + \Delta$$

$\mathbf{X}$ is the original matrix and $G(\mathbf{X})$ is the perturbed one. $\mathbf{R}$, Ψ, and Δ are the multiplicative, translation, and additive matrices respectively. The integration of these sub-transformations shows well utility and privacy guarantees during computations [106], [107].

Lastly, we briefly describe a special perturbation method called permutation technique, which is based on a permutation function to disorder the original elements without changing the values. Matrix permutation (which we introduced previously), permuting the rows and columns of a matrix, is a common case. A permutation function can be expressed as $\pi(i) = p_i$ $(i = 1, \ldots, n)$, where the independent variable i is the original sorting label and $\pi(i)$ is the rearranged label. In other words, the element labeled by i at first will be replaced by the one labeled by p_i. For example, suppose we have a private matrix $\mathbf{A}$ (in which $\mathbf{A}[i, j]$ is the element locating at the ith row and the jth column of $\mathbf{A}$). An effective permutation operation can be performed like: $\mathbf{B}[i, j] = \mathbf{A}[\pi_1(i), \pi_2(j)]$, where $\pi_1(\cdot)$ and $\pi_2(\cdot)$ are two defined permutation function. The elements in matrix $\mathbf{A}$ are thereby rearranged and reconstructed into a permuted matrix $\mathbf{B}$. Permutation-based approaches have been applied in many specific outsourcing schemes, such as linear algebra [108], [109], image processing [110], and data mining [111].

Due to the limited space, we will not give introductions to other types of perturbation methods (please refer to [9] or [112] for more details). In perturbation-based schemes, randomness of the random values or permutations is deemed as the secret key of users. Compared with cryptography-based techniques, perturbation methods usually lead to lower computational complexities due to their relatively simple operations. However, the ability for privacy protection is generally inferior to the methods based on cryptography. For example, some important information cannot be preserved as the limitations of linear transformations. Moreover, the quality of the random components and functions also matters to the performance of the transform algorithms.

IV. HOMOMORPHIC ENCRYPTION

To provide better privacy protection for outsourcing computation, researchers have proposed many cryptographic algorithms. However, traditional encryption schemes such as AES (advanced encryption standard) [113] require a recovery of the encrypted data before computation. That is, before processing the data, the cloud server should first decrypt the encrypted data using the secret key, and then perform specific functions on the plaintexts. In this case, the user's sensitive information will be exposed to the cloud server. The raise of homomorphic encryption (HE) technique well resolves the concern. HE allows arithmetic operations to be directly performed on the encrypted data without decryption in advance. The computed result matches the encrypted result (by the same encryption algorithm) operating on plaintexts.

In this section, we first introduce the foundations of HE theory. Then, we list classical categories of HE algorithms and briefly demonstrate their corresponding evolutions and implements. Lastly, we summarize the benefits and disadvantages of HE technique. A comparison of homomorphic encryption and other secure techniques (introduced in Section III) is listed in Table 2.

A. DEFINITION AND BASIC FUNCTIONS OF HOMOMORPHIC ENCRYPTION

Generally, an HE scheme can be defined as follows:

Definition 2: An encryption scheme is called homomorphic over an operation "Θ_m" if the following equation holds:

$$\mathrm{E}(m_1)\Theta_c\mathrm{E}(m_2) = \mathrm{E}(m_1\Theta_m m_2), \forall m_1, m_2 \in M$$

$\mathrm{E}(\cdot)$ is the encryption algorithm and $\mathrm{E}(x)$ is the corresponding ciphertext of the message x. M is the set of plaintexts. Operator "Θ_m" or "Θ_c" denotes some operations over the domain of plaintexts or ciphertexts, respectively. If the operation "Θ_m" is an addition operation, then we say this encryption scheme satisfies additive homomorphism. Likewise, if "Θ_m" is a multiplication operation, the property is known as multiplicative homomorphism.

An HE scheme is primarily composed of four operations: *KeyGen*, *Enc*, *Dec*, and *Eval*. The first three functions are much the same as the ones of the traditional encryption schemes. *Eval* is an HE-specific function that executes certain calculations on the ciphertexts. We take the asymmetric HE scheme as an example to illustrate the algorithm processes:

-*KeyGen*(λ). Given the security parameter λ, the system generates a public and secret key pair: $\{pk, sk\}$. Note that pk is public while sk should be preserved secretly by the decryptor.

-*Enc*(pk, m). The algorithm takes as input the public key pk and a message $m \in M$ (M is the domain of plaintexts) and outputs the corresponding ciphertext c of m.

TABLE 2. An comparison of homomorphic encryption and other secure techniques.

Secure technique	Cryptographic or Non-cryptographic	Storage overhead	Requires interaction	Strengthes	Weaknesses
MPC	Cryptographic	High	Yes	Achieving secure computation among multiple users without third parties	Online requirement and storage burden of users
PRFs and PRPs	Cryptographic	Low	No	Easy to perform; providing efficient verification	Application scenario limited
SGX	Cryptographic	High	No	Hardware security: trusted execution environment	Vulnerable to certain software and physical attacks
PBM	Non-cryptographic	Low	No	More efficient; easy to perform	Application scenario limited; lower security level
HE	Cryptographic	High	No	More secure and general	Relatively inefficient

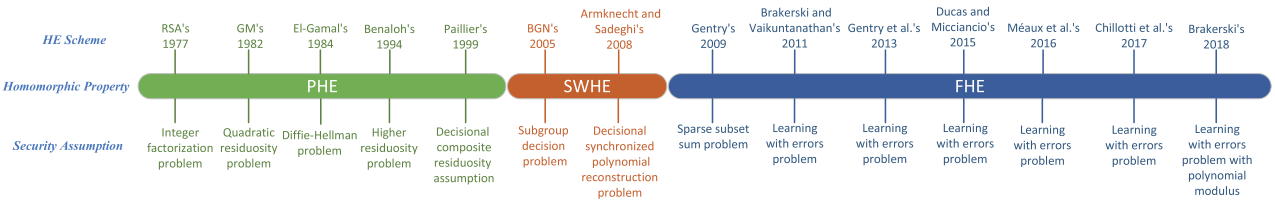


FIGURE 3. Time-flow of some remarkable HE schemes and their security assumptions.

-$Dec(sk, c)$. The ciphertext c can be decrypted using the secret key sk. After the decryption operations, message m is restored.

-$Eval(pk, f(\cdot,\cdot), c_1, c_2)$. $Eval$ performs the supported function $f(\cdot,\cdot)$ (like an addition operation in additive-homomorphism schemes) directly over two ciphertexts (c_1, c_2) using pk, and outputs the encrypted result $f(c_1, c_2)$. The computational result over the corresponding plaintexts of (c_1, c_2) can be correctly obtained by decrypting $f(c_1, c_2)$.

Here, we take ElGamal cryptosystem [114] as an example to illustrate the concrete construction of the above operations.

-$KeyGen(\lambda)$. Given a security parameter λ, construct a cyclic group G with order n and generator g (such that $|g| = \lambda$). Then choose a random value $x \in \mathbb{Z}_n^*$ and compute $h = g^x$. Then, output public/private key pairs:

$$\{pk; sk\} \leftarrow \{(\mathrm{G}, n, g, h); x\}.$$

-$Enc(pk, m)$. The algorithm takes as input the public key $pk = (\mathrm{G}, n, g, h)$ and a message $m \in \mathrm{G}$. It chooses a random $y \in \mathbb{Z}_n^*$ and computes $h_1 = h^y$. The ciphertext of m is calculated as:

$$c = (c', c'') = (g^y, mh_1) = (g^y, mg^{xy}).$$

-$Dec(sk, c)$. The ciphertext $c = (c', c'')$ can be decrypted using the secret key $sk = x$. Calculate $s = (c')^x = g^{xy}$. Then, m can be recovered by

$$c'' \cdot s^{-1} = mg^{xy} \cdot g^{-xy} = m.$$

-$Eval(pk, mul(\cdot,\cdot), c_1, c_2)$. Take as input the public key $pk = (\mathrm{G}, n, g, h)$, the ciphertexts $c_1 = Enc(pk, m_1)$ and $c_2 = Enc(pk, m_2)$, and multiplication function $mul(\cdot,\cdot)$. It computes the encrypted product of m_1 and m_2 as

$$Enc(pk, m_1) * Enc(pk, m_2) = c_1 * c_2 = (g^{y_1}, m_1 h^{y_1}) * (g^{y_2}, m_2 h^{y_2}) = (g^{y_1+y_2}, m_1 m_2 h^{y_1+y_2}) = Enc(pk, m_1 m_2),$$

where the symbol "$*$" denotes component-wise multiplication. It is easy to deduce that ElGamal cryptosystem satisfies multiplicative homomorphism.

B. HOMOMORPHIC ENCRYPTION SCHEME CLASSIFICATION AND RELATED WORKS

Like the conventional cryptography, depending on the types of keys used to encrypt and decrypt data (either by using the same pair of keys or a different pair), HE schemes are known as symmetric HE or as asymmetric HE. Due to the consideration of the poor practicability, key management overhead, and security flaws of symmetric encryption, fewer symmetry-based HE schemes (like [115], [116]) were proposed. The researchers nowadays prefer using asymmetric algorithms to implement HE schemes, such as [82], [117].

On the other hand, according to the homomorphic ability, homomorphic encryption can be broadly categorized into partially homomorphic encryption (PHE), somewhat homomorphic encryption (SWHE), and fully homomorphic encryption (FHE). Since the operations of addition and multiplication are functionally complete over finite sets, it is sufficient to employ these two basic operations to construct arbitrary functions for homomorphic evaluation. The three classifications for homomorphic computations are in terms of the characteristics of executing addition and multiplication operations, which will be discussed later. The remarkable PHE, SWHE, and FHE schemes (as well as their security assumptions) are summarized in the time-flow in Figure 3 and are explained with greater detail in this subsection.

1) PARTIALLY HOMOMORPHIC ENCRYPTION (PHE)

Partially homomorphic encryption (PHE) allows only one type of operations (i.e., addition or multiplication) on the ciphertexts with an unlimited number of times. Accordingly, PHE can be divided into two types: additively homomorphic encryption and multiplicatively homomorphic encryption.

Additively homomorphic encryption (AHE) only allows performing unlimited times of additive calculations over the ciphertexts. Goldwasser-Micali (GM) cryptosystem [118] was the first probabilistic provably secure public-key encryption scheme based on quadratic residuosity which allows the addition modulo 2 (i.e., exclusive-or operation) of the plaintext over the ciphertexts. Benaloh's homomorphic encryption function [119], originally designed for electronic voting and relying on prime residuosity, prefigured the first attempt to exploit the plain resources of this theory. Later, Paillier [82] proposed a new additively homomorphic cryptographic building-block based on composite residuosity which has been widely used for keyword search on the remote encrypted data [120], privacy-preserving aggregation [121], etc. There have been numerous AHE-based schemes proposed for outsourcing computation. For example, Samanthula *et al.* [122] designed a secure interactive multiplication, comparison protocol between two parties. As this AHE scheme only supports integer operations, Liu *et al.* [10] extended the integer-based outsourced calculation to support rational numbers in the twin-cloud environment and can be used for processing sensitive health data. The AHE-based scheme is efficient, however, the non-colluding twin-cloud architecture and interactive communications of the two servers are still necessary.

Despite a bunch of AHE schemes [123], [124], multiplicatively homomorphic encryption (MHE) cryptosystems are also attractive, which allow multiplicative operations for unlimited times on the ciphertext space. Rivest *et al.* [125] introduced the first wildly used public-key encryption scheme called RSA (Rivest-Shamir-Adleman) which satisfies multiplicative homomorphism. Another famous MHE scheme is called ElGamal cryptosystem [114], which is based on a well-known open hard problem called discrete logarithm. Although many practical MHE schemes like [126], [127] were proposed, there exists an inherent drawback in MHE schemes: it cannot store the key element 0 due to its algebraic structure.

2) SOMEWHAT HOMOMORPHIC ENCRYPTION (SWHE)

Somewhat homomorphic encryption (SWHE) performs finite steps of homomorphic operations for both addition and multiplication on the ciphertexts. For example, Boneh-Goh-Nissim (BGN) cryptosystem [128] supports a limited number of additively homomorphic operations and only once multiplicatively homomorphic operation. Armknecht and Sadeghi [129] proposed an SWHE scheme based on a coding theory problem. It allows arbitrary times of additions and a fixed number of multiplications, working over arbitrary finite domains. However, the scheme is symmetric and the ciphertext size grows exponentially with the expected total number of encryptions. Although SWHE supports additions and multiplications simultaneously, the allowed number of operations is limited, and it can only be used for small-scale programs/circuits.

3) FULLY HOMOMORPHIC ENCRYPTION (FHE)

As an ultimate solution for secure computation, fully homomorphic encryption (FHE) allows arbitrary operations (i.e., addition and multiplication) with an unlimited number of times over ciphertexts. Gentry and Boneh [130] gave the first generation of FHE, constructed an SWHE scheme, and made it bootstrappable. Specifically, the scheme can perform at least one more homomorphic operation after evaluating its own homomorphic decryption circuit over the encrypted domain. However, according to Gentry and Halevi [131], a basic bit operation based on the idea of [130] requires 30 minutes, which is unrealistic performance. Although several refinements and optimizations [132], [133] were subsequently proposed, these schemes are still impractical for real-world applications in terms of both the ciphertext size and the running time. The second generation of FHE was first constructed by Brakerski and Vaikuntanathan [134], based on the hardness of the learning with errors (LWE) problem. In a separate work, Brakerski *et al.* [135] built a new efficient tool to reduce ciphertext noise. Furthermore, Gentry *et al.* [54] presented a new approximate eigenvector method to make homomorphic addition and multiplication run more efficiently. To store data more efficiently, Smart and Vercauteren [136] outlined a technique to allow the packing of several ciphertext values in a single ciphertext and operate on the values in a single instruction multiple data (SIMD) fashion. To implement the second generation of FHE scheme, Halevi and Shoup [137] built a library called HElib to implement the BGV (Brakerski-Gentry-Vaikuntanathan) cryptosystem and bootstrapping method [138]. Gentry *et al.* [139] also implemented the AES-encryption circuit under HElib. Although these encryption schemes are almost practical, it still requires 182.71 seconds to perform the bootstrapping procedure. Ducas and Micciancio [140] constructed the third generation of FHE scheme with efficient bootstrapping, and Chillotti *et al.* [141], [142] gave two optimal methods for bootstrapping to make the FHE scheme practical. With the advantages of both block ciphers and stream ciphers evaluation, Méaux *et al.* [143] designed an efficient FHE scheme which leads to low-noise of ciphertexts. Brakerski [144] presented a quantum FHE (QFHE) scheme, with the supporting functions to the ones computable in quantum polynomial time. The error of the homomorphic evaluation can be only exponentially small at any polynomial quantum circuit. In the work of [145], the authors constructed a threshold FHE (ThFHE) scheme with the security assumption of learning with errors. Moreover, a general framework for threshold cryptosystems was also given from ThFHE. Due to the natural advantage of FHE (all the computations can be executed in a single semi-trusted server), a lot of applications are designed based on FHE algorithms, such as association rule mining [146], private information retrieval [147], and clinical decision support system [148]. However, the high computation and storage overhead of FHE is still a barrier for wide-scope applications of FHE-based outsourcing computation.

4) SUMMARY

Homomorphic encryption is a popular cryptographic tool used in cloud computing, owing to the operability on encrypted data at the servers. The processes of data transmission and computation can be accomplished without decryption, and the used cryptosystems always provide advanced security strengths. This is the major reason why abundant outsourcing schemes employ homomorphic encryption for privacy-preserving, instead of other secure technologies. According to the characteristics of supporting operations, HE schemes are generally categorized into three types: PHE, SWHW, and FHE. Although supporting only one kind of operation, PHE-based algorithms have received massive attention and been used in many realistic applications for its relatively low computation and storage overhead. SWHE executes both the addition and multiplication operations but for a limited time, still suitable for some finite application domains. FHE-based schemes perform arbitrary functions for arbitrary times, while producing higher computation and storage overhead. Therefore, more refinements and improvements should be achieved to reduce the heavy burden of FHE-based schemes.

To sum up, in the meantime of providing stronger privacy protection, there are three main drawbacks of HE schemes. 1) High computation cost: for most of PHE and SWHW schemes, it requires several modular exponent arithmetics for designing the secure computation protocol. Moreover, FHE algorithms need an essential technique called bootstrapping to reduce the noises from the ciphertext, which will significantly decrease the efficiency of the whole system. 2) Large storage overhead: the storage cost for ciphertexts will be expanded for several times compared with the original plaintext, and the storage needs to be expanded for hundred or even thousand times for most of the FHE scheme (even the same for some PHE and SWHW schemes). 3) Trusted authority (TA) is required: TA is in charge of generating and distributing the public/private key for all the parties in the system. At least one TA is required for HE-based schemes.

V. SECURE OUTSOURCING OF FUNDAMENTAL FUNCTIONS

Generally speaking, the scope of outsourcing computation can be categorized into fundamental and application-specific functions. Fundamental functions are considered as some simple operations for resolving basic arithmetic problems, like the addition of two scalars or matrices. They are also used as building blocks of complex tasks. In this section, we give an overview of existing HE-based outsourcing computation schemes for fundamental functions, including scalar operations, set operations, matrix operations, and systems of linear equations. The schemes' security levels and performance evaluations are analyzed in Section VII.

A. SCALAR OPERATIONS

Scalar operations, always seen as the most common elementary operations, are the building blocks of general operations. The calculations on scalars can be roughly classified into basic and blend arithmetic operations. The basic scalar operations include single calculations on scalars, like addition, subtraction, multiplication, division, comparison, sorting, square root, exponentiation, greatest common divisor (GCD), etc. Blend arithmetic operation is the calculation of polynomials made up of several operations (i.e., addition, multiplication, exponential, or other operators). We discuss existing HE-based outsourcing schemes for secure storing and processing scalars, from the perspective of basic and blend arithmetic operations.

Considering secure basic arithmetic operations, several efficient outsourcing schemes were designed for integer numbers. For example, Gong *et al.* [149] proposed a secure integer arithmetic framework, supporting the operations of addition, multiplication, complement, etc. In the protocols, the sensitive values are protected due to fully homomorphic properties. Based on Paillier cryptosystem, the protocol of [150] calculated a secure quotient over two encrypted positive integers, with the encrypted result in a fixed-point format. The computational complexity grows linearly with respect to the desired quotient precision. Liu *et al.* [151] built secure circuits for commonly-used unsigned and signed integer computations (e.g., integer multiplication and comparison). In the work, the operations are performed on encrypted operands by FHE algorithms. Moreover, the authors also adopted SIMD technique to support batch computations, thereby largely reducing the burdens of storage and computation.

However, as many specific application fields require the data with higher precision, integer processing cannot satisfy their actual needs. Since standard encryption algorithms only support integer values (in finite domains), some effective schemes were elaborately designed to complete secure non-integer computations, which cover different types of basic operations. Liu *et al.* [152] proposed a framework for privacy-preserving outsourced calculation on floating-point numbers (FPNs). In the framework, real numbers are expressed or approximated by FPNs, which are encrypted and outsourced with the specific format and constant-length. The authors first constructed a secure calculation toolkit for common integer operations (e.g., multiplication, comparison, or modular calculation). Taking these integer protocols as sub-protocols, several protocols for FPN operations were also introduced (i.e., equivalence testing, sorting with absolute value, addition, multiplication, and comparison of FPNs). The framework uses Paillier cryptosystem with partial decryption (PCPD) as a solution to enable direct computations on the encrypted data. To reduce leaking risk of private key, in the cryptosystem the private key is split into two partial shares and distributed to the CP and CSP parties respectively. After the interactions between the CP and CSP, the required computations are completed securely without revealing the sensitive values to unauthorized parties. Also, the framework can handle FPN exceptions (such as overflow and underflow). Moreover, the interaction between the clients and the cloud server is kept to the minimal: the client only needs to send a computational query to the cloud platform,

and then receive the result in a single round. However, since the Paillier encryption only supports homomorphic additions, the work adopts a twin-server model to construct multiplicatively homomorphic protocols. Hence, a higher security assumption is required. Similarly, the authors further designed a framework [10] to support basic computations on integer and rational numbers. In the work, the rational number is expressed or approximated in a fraction format, whose numerator and denominator are separately encrypted and handled.

Arita and Nakasato [153] constructed the first FHE-based secure protocols for encrypted fixed-point number operations (i.e., addition and multiplication). Nevertheless, the work requires a large space for key and ciphertext storage, and cannot calculate complex functions on the FPNs. To remove these limitations, Bai *et al.* [154] designed privacy-preserving protocols for floating-point number addition and multiplication, with low ciphertext expansion ratio and small public key size. Moreover, the calculations can be generalized to analytic functions (e.g., exponential function and logarithmic function) by utilizing Taylor series. The precision of the calculations is almost the same as the unencrypted case. Scheme [155] realized parallel FHE algorithm for floating-point number operations, based on the MapReduce environment. Moreover, to provide stronger security, the order of ciphertexts is disrupted to eliminate the relevance between the child ciphertext and key pair. Basilakis and Javadi [156] also designed a parallel solution, which securely performs binary operations (i.e., comparison, addition, and subtraction operations) over real numbers. By using the packing SIMD technique, the system's overall performance can be significantly accelerated.

When multiple data providers are involved to outsource their data to the same cloud server, they should be distributed with individual keys to avoid multi-tenancy related attacks. However, achieving secure calculations under multiple keys while protecting individual data is another difficulty. By adopting a new cryptographic method called distributed two trapdoors public-key cryptosystem (DT-PKC), Liu *et al.* [157] built a privacy-preserving outsourced calculation toolkit of integer numbers for the scenarios involving multiple data providers (DPs). When a request user (RU) sends a computational query correlated to the data from DP a and DP b (we call the encrypted data $[x]_a$ and $[y]_b$, respectively), the server will perform the necessary homomorphic computations and finally return the encrypted result $[f(x, y)]_{pk_\sigma}$ to the user, where $f(\cdot, \cdot)$ is the requested function and pk_σ is a joint public key associated with different DPs and the RU. If DP a (DP b) allows the RU to access the result, DP a (DP b) will partially decrypt the encrypted result and send partial decrypted ciphertext WT_a (WT_b) to the user. Then the user executes the second stage of partial decryption using his own secret key and gets the result in plaintext. The toolkit can also be extended to store and process real numbers. Across large-scale multiple encrypted domains, a secure framework for outsourcing functional computations was proposed by Liu *et al.* [158], called as POFD. In the work, a user defines a function and obtains the result of the function over encrypted data from different data providers. Neither the function nor the input/output will be revealed to unauthorized parties. In the paper, two versions of POFD are presented: basic POFD and enhanced POFD (the latter achieves a higher security level). Supporting multiple encrypted domains, [159] also gave protocols for securely non-integer processing, based on the cryptosystem of [157]. Furthermore, the authors continued to construct secure algorithms for reinforcement learning, which can be applied for decision-making in diagnosis systems.

Researchers have also designed secure solutions to handle blend arithmetic operations. In [160], Gai *et al.*and Qiu proposed a solution for blend arithmetic processing on additions and multiplications over real numbers. The solution removes all the parentheses and transforms the formula into a set of binomials joined by addition operators. Hence, the blend arithmetic operations can be solved using familiar algorithms, like homomorphic multiplications and additions. Liu *et al.* [161] designed an efficient privacy-preserving outsourced functional computation framework over public data. The work adopts switchable homomorphic encryption [162] with partially decryption (SHED) as the core cryptographic algorithm. Two coding methods (i.e., message pre-coding technique and, message extending and coding technique) are introduced to transform the values into the input domain of SHED. Yu *et al.* [163] proposed a verifiable scheme for outsourced function evaluation over ciphertexts, using the properties of FHE scheme. To achieve data confidentiality and verifiability, especially two additional entities are introduced: a trusted authenticator (TA) and a public auditor proxy (PAP). When the client requests for the evaluation result, the TA will check the client's certificate and then re-encrypts the computational result into another lightweight one if the verification is satisfied. Interacting with the TA, the PAP checks the correctness of the encrypted result homomorphically. If the result is validated, it will be returned to the client.

B. SET OPERATIONS

The set, a commonly used data structure, is served as a container for different objects. The major operations on sets include set intersection, set union, and set difference, which have been served as building blocks in many specific applications, such as data mining, graph algorithms, and recommendation services. In this subsection, we mainly discuss outsourcing schemes for set intersection, set union, and their variant (i.e., set-intersection and set-union cardinality). Since the records inside a set are often sensitive for users, the security of set elements and set-operation results is required to be guaranteed.

1) SET INTERSECTION

Set intersection is the operation to obtain the intersecting elements of involved sets. Assume there are n parties ($n \geq 2$), each holding a private set S_i ($1 \leq i \leq n$). The desired

set-intersection output is written as $S_1 \bigcap S_2 \bigcap \ldots \bigcap S_n$, where the elements included are the ones exist in all n sets simultaneously. In the delegating environments, one or more clients obtain the intersection result while their individual sets are kept private. The server executes the set intersection operations efficiently and learns nothing from the processing.

As proposed Freedman *et al.* [48] discussed secure two-party set-intersection protocols in the semi-honest and the malicious adversary model. In the protocol, the client's set is $X = \{x_1, \ldots, x_{n_C}\}$, while the cloud server's set is $Y = \{y_1, \ldots, y_{n_S}\}$. Under the semi-honest setting, the client first defines a polynomial $P(y) = (x_1 - y)(x_2 - y) \cdots (x_{n_C} - y) = \sum_{k=0}^{n_C} \alpha_k y^k$, where the roots (i.e., $x_1, x_2, \ldots, x_{n_C}$) are his input set members. Then he encrypts the $n_C + 1$ coefficients of the polynomial using a semantically-secure homomorphic encryption scheme. The encrypted coefficients $\{\text{Enc}(\alpha_0), \ldots, \text{Enc}(\alpha_{n_C})\}$ are then sent to the server. For each value y_i ($i \in \{1, \ldots, n_S\}$) in the set Y, the server does the following calculations. By exploiting HE properties, the server computes $\text{Enc}(P(y_i)) = \text{Enc}(\sum_{k=0}^{n_C} \alpha_k y_i^k)$, and chooses a random value r_i to computes $\text{Enc}(r_i P(y_i) + y_i)$. Apparently, if $y_i \in \{x_1, \ldots, x_{n_C}\}$, then $\text{Enc}(r_i P(y_i) + y_i) = \text{Enc}(r_i \cdot 0 + y_i) = \text{Enc}(y_i)$; otherwise, even decrypting $\text{Enc}(r_i P(y_i) + y_i)$, the value of y_i is still unknown. Afterwards, the server permutes the ciphertexts $\text{Enc}(r_i P(y_i) + y_i)$ ($i \in \{1, \ldots, n_S\}$) randomly and sends it back to the client. Receiving the result, the client decrypts all the ciphertexts, compares with his own private set, and obtains the intersection $X \bigcap Y$. To reduce the degree of the polynomials, the protocol also employs the balanced allocation method [164]. Specifically, the idea is to map the elements to corresponding bins whose size upper-bounds are limited. Based on the protocol against semi-honest parties, the authors further extended the protocol for malicious models. Based on the idea of [48], Dachman-Soled *et al.* [165] described a robust protocol for set intersection, with verifiability in the malicious setting. The algorithm also employs a Shamir secret sharing technique to share the server's set through a k-degree polynomial, in which k is the security parameter. To verify the correctness of final results, the server and the client jointly run a cut-and-choose protocol on the server's set. Finally, the client correctly obtains the elements of the server's set which is also in his own set.

Apart from privacy and correctness requirements, efficiency is also an important factor to be considered in the outsourcing schemes. Utilizing leveled FHE scheme, Chen *et al.* [166] constructed a private set intersection protocol considering both security and practicability in the semi-honest model. By combining various optimizations (e.g., batching and hashing techniques), the communication and computation cost is largely reduced. Supposing the smaller set with the size N_s and the larger one with N_l, the communication complexity of the proposed scheme is $O(N_s \log N_l)$. With the multiplicatively homomorphic property of RSA cryptosystem, Yang *et al.* [167] presented an efficient set intersection protocol, secure in the semi-honest model. The protocol assumes that two different parties (i.e., A and B) hold their individual private sets, which are encrypted and outsourced to the cloud. When one party A tries to obtain the intersection result of their sets, he sends a requesting signal to the other party B. If B agrees to engage the set intersection, he will send a permit message and some necessary information to the cloud. Due to homomorphic properties, the cloud server operates on the encrypted sets and returns the intersection result (which is also in the encrypted form) to A. Finally, the party A decrypts the result and recovers the set intersection without learning B's private set. The computation at the clients only involves several simple modular multiplications. In the case of multiple clients (each holding a secret set), the cloud will execute the set intersection operations among the involved encrypted sets after receiving permit messages from the owners.

Instead of polynomial representation, Ruan *et al.* [168] expressed the sets as bit vectors, and the set-intersection operation is thereby transformed into vector operations. Supposing one input set S is selected from an n-element set $X = (x_1, x_2, \ldots, x_n)$, the vector-representation corresponding to S is $(v_1, v_2, \ldots, v_n)$, where $v_i = 1$ if $x_i \in S$ and $v_i = 0$ otherwise ($1 \leq i \leq n$). If two input sets are denoted as $A = (a_1, a_2, \ldots, a_n)$ and $B = (b_1, b_2, \ldots, b_n)$, then the set intersection (with vector representation) can be computed by multiplying the elements of both sets at the same position:

$$\begin{aligned} A \bigcap B &= (a_1, a_2, \ldots, a_n) \bigcap (b_1, b_2, \ldots, b_n) \\ &= (a_1 b_1, a_2 b_2, \ldots, a_n b_n) \end{aligned}$$

The process can be securely performed by the cloud server due to PHE characteristics. Based on GM cryptosystem, Zhu *et al.* [169] constructed another set-representation form based on Bloom filter. The protocol allows multiple clients to outsource their sets and obtain the set-intersection result without revealing their private sets.

2) SET UNION

Set union is another fundamental set operation. Assuming we have n parties each holding an input set S_i ($1 \leq i \leq n$), the set-union operation is to obtain the items which appear in at least one participant's input set without knowing anything else, written as $S_1 \bigcup S_2 \bigcup \ldots S_n$. There has been relatively little work done for set-union computation by homomorphic cryptosystems.

Based on AHE scheme, Frikken [170] introduced a privacy-preserving set-union solution secure in the semi-honest model. Like most set-operation algorithm, the work employs polynomials to represent set elements, where the polynomial coefficients are encrypted by HE algorithm. In the protocol, one participant P_1 (i.e., the client) first encrypts his polynomially-represented set $f(\cdot)$ as $\text{Enc}(f(\cdot))$ ($\text{Enc}(\cdot)$ is the adopted encryption algorithm), where $f(x) = 0$ if and only if x belongs to P_1's set. The encrypted polynomial is then outsourced to the other party P_2 (i.e., the server). After that, P_2 computes the tuples $(\text{Enc}(f(s_i) \cdot s_i \cdot r_i); \text{Enc}(f(s_i) \cdot r_i))$

homomorphically for each element s_i in his own set (the random value r_i is uniformly chosen), and sends them to P_1 in a random order. After that, P_1 decrypts the tuples and obtains the elements s_i which appear in P_2's set but not in P_1's (i.e., $f(s_i) \cdot r_i \neq 0$), denoted by a set X. The result of set union equals to the elements of X along with the original members of P_1's set. The computation and communication complexities of the proposed scheme are $O(n^2)$ and $O(n)$ (n is the size of the input sets), respectively. With linear complexities of computation and communication, [171] achieved private set union based on Bloom filter and AHE scheme. The framework assumes two parties (i.e., a client and a server) each holding a private set, whose size is informed to the other side. The client's set is transformed into an encrypted bloom filter and processed homomorphically for union operations at the server. The input and output privacy is guaranteed for the semi-honest model. By adding the authorization from a trusted third party, the protocol also achieves input security against malicious adversaries.

3) SET-INTERSECTION AND SET-UNION CARDINALITY

In this subsection, we cover existing privacy-preserving outsourcing schemes of computing set-intersection and set-union cardinality. The protocols for the cardinality of set intersection or set union are to compute the size of the intersection or union result of all involved private input sets. Assume that there are n data owners with private sets $S_1, S_2, \ldots, S_n$, respectively. The client will learn the value of $|S_1 \bigcap S_2 \bigcap \ldots \bigcap S_n|$ (or $|S_1 \bigcup S_2 \bigcup \ldots \bigcup S_n|$) but not the actual elements of the intersection set or the union set. The contents of the private sets are unknown to the computing service provider and the unauthorized parties.

Freedman *et al.* [48] sketched a protocol of cardinality set-intersection, which is slightly modified from the set-intersection protocol for the semi-honest setting in the work (stated in Section V-B.1). After the computation, the client learns the cardinality of the final intersection set, while the server learns nothing. Scheme [168] (stated in V-B.1) also introduced a secure set-intersection cardinality protocol, with the sets represented as bit vectors. If two sets are denoted as vectors $A = (a_1, a_2, \ldots, a_n)$ and $B = (b_1, b_2, \ldots, b_n)$, the computation of set-intersection cardinality is equivalent to the inner product of both vectors as: $|A \bigcap B| = |(a_1, a_2, \ldots, a_n) \bigcap (b_1, b_2, \ldots, b_n)| = a_1b_1 + a_2b_2 + \ldots + a_nb_n$. In the protocol, supposing one cloud server owns the set A and one client owns the set B, the server performs the transformed vector operations homomorphically over its set and the client's encrypted set. Finally, the client decrypts the returned result and obtains set-intersection cardinality between their sets. In [171] (stated in Section V-B.2), the authors extended the steps of the original set-union algorithm to compute the cardinality of set union or set intersection with linear computation and communication complexities. Taking advantage of the set property $|A \bigcap B| = |A| + |B| - |A \bigcup B|$, one of the cardinalities of set intersection and set union can be obtained when the other cardinality value is known.

With FHE properties, Tajima *et al.* [172] designed another protocol for secure outsourced private set intersection cardinality computations. In the work, two data owners transform their private sets into bloom filters and encrypt the bloom filters using the public key generated by the client, and then send them to the cloud. In the query stage, the cloud operates on the encrypted bloom filters homomorphically and sends the resulting bloom filters to the client. The client recovers the result using his secret key and obtains the intersection cardinality of the input sets. The authors described two secure solutions, with the burdens at the client of $O(N_s)$ and $O(N_s/L)$ (N_s is the size of the smaller set and L is the slots of a single ciphertext), respectively. In the second solution, the server aggregates the resulting bloom filters to decrease the number of returned ciphertexts. Thus, the computation cost of the client is lightened by delegating more workloads to the server.

C. MATRIX OPERATIONS

Linear algebra, a branch of mathematics, has become a widely used tool in various application fields (especially in scientific and engineering fields). Nowadays, numerous applications involve large amounts of data, which is arranged in the matrices with large dimensions. Undoubtedly, computing on such large-scale matrices is a heavy burden for weak devices. Hence, outsourcing expensive matrix operations is desirable to most individuals. In this subsection, we mainly discuss secure delegation schemes of matrix multiplication and several other matrix operations, and analyze their properties of efficiency, privacy, and verifiability.

1) MATRIX MULTIPLICATION

Matrix multiplication is an operation between two matrices, which is frequently used as a building block no matter in specific applications or other matrix operations. In the outsourcing schemes for matrix multiplication, supposing matrix $\mathbf{A}$ and $\mathbf{B}$ are input matrices, after the computation by the server side, the client will finally obtain the multiplication result from $\mathbf{C} = \mathbf{AB}$ with minimum overhead. The server will never know the original input matrices nor the final multiplication result. When operating on the ($n \times n$)-dimension matrices, the best known theoretical upper bound for matrix multiplication is $O(n^{\omega})$ ($\omega \cong 2.38$). In practice, however, the computation complexity often closes to $O(n^3)$.

Outsourcing schemes for matrix multiplication with verifiability have been studied in many works. Benjamin and Atallah [173] designed protocols for secure outsourcing matrix multiplication to two cloud servers. Each input matrix is randomly split into two shares and outsourced to both servers, respectively. During intermediate stages of the protocol, when dealing with matrix multiplication between $n \times n$-matrices $\mathbf{X}$ and $\mathbf{Y}$, the computation can be securely executed

as:

$$\mathrm{Enc}(\mathbf{X}\cdot\mathbf{Y})[i,j] = \mathrm{Enc}(\sum_{k=1}^{n}\mathbf{X}[i,k]\mathbf{Y}[k,j]) = \prod_{k=1}^{n}\mathrm{Enc}(\mathbf{X}[i,k])^{\mathbf{Y}[k,j]}$$

$\mathbf{M}[i,j]$ denotes the element locating at the i-th row and the j-th column of the matrix $\mathbf{M}$. The symbol $\mathrm{Enc}(\cdot)$ represents the homomorphic encryption algorithm. Except for guaranteeing the input and output privacy, the work realizes checkability regarding the returned matrix. Specifically, any possible corruption behavior can be detected by the client with a high probability, even when the servers collude. After receiving the result matrix $\mathbf{C}$ (i.e., $\mathbf{C} = \mathbf{AB}$), the cheating-detection mechanism will be carried out: the client first generates a random $(n\times 1)$-dimension column vector $\boldsymbol{v}$. After that, he calculates $\boldsymbol{x} = \mathbf{C}\boldsymbol{v}$ and $\boldsymbol{y} = \mathbf{A}(\mathbf{B}\boldsymbol{v})$. If the vector $\boldsymbol{x}$ is not equal to $\boldsymbol{y}$, then the servers are proved to be dishonest. Maintaining strong checkability, Atallah and Frikken [174] proposed an improved solution utilizing only one server by the combination technique of extending Shamir's secret sharing and semantically-secure AHE scheme. Scheme [175] analyzed the verifiability of the delegating homomorphic matrix multiplication, based on different HE schemes. In the work, it is proved that if the adopted HE scheme satisfies two properties (i.e., associativity and distinctiveness), the computational result can be verified with $O(n^2)$ complexity.

With a novel packing method, Duong *et al.* [176] proposed an efficient scheme for secure matrix multiplication. In the protocol, the entries of input matrix are packed into a single polynomial and encrypted using SWHE scheme [177]. In the protocol, the multiplication between two matrices only requires one homomorphic multiplication operation over the ciphertexts. As [176] only supported multiplication operation between two matrices, an advanced work [178] for multiple-matrix multiplication was further presented, based on BGV cryptosystem. In addition, the method was implemented under HElib, showing great efficiency performance. Lu *et al.* [179] also presented a secure matrix multiplication protocol with higher efficiency. To reduce the overhead of computation and communication, several optimizations were introduced. On the one hand, with the Chinese-remainder-theorem (CRT) packing, an efficient packing technique is designed to compute a batch of inner products at the cost of a single homomorphic operation. On the other hand, a precomputed table is constructed at the beginning of the protocol and reused multiple times by the client, thus largely reducing the client's workload. The scheme was proved to work well even with a high concurrency.

2) OTHER MATRIX OPERATIONS

In the following, we briefly introduce several outsourcing schemes for other matrix operations, which are also computation-intensive. Based on the secure delegating protocol for matrix multiplication described Mohassel [175] further proposed efficient and constant-round constructions for several other matrix operations, such as matrix inversion or minimal polynomial of a matrix. The work also gives secure solutions to test matrix singularity and compute matrix rank and determinant, which are reduced to computing matrix minimal polynomial (based on the ideas of [180]). By adopting a weave ElGamal encryption scheme, Chen *et al.* [181] designed secure and efficient outsourcing protocols for certain matrix operations over $\mathbb{Z}_p$. In the protocols, the cloud performs the functions of Gaussian elimination and its related algebraic computations (i.e., Gaussian-Jordan elimination, matrix determinant, linear system solver, and matrix inversion) without learning any non-zero element of the matrix. The encrypted matrix can be transformed into a row/column echelon form through certain elementary row/column operations. Moreover, the linear system solver and matrix inversion protocols enable the clients to verify the returned results.

Finding the matrix's eigenvalues and eigenvectors is also a crucial sub-task in many scientific and engineering computations. In general, the direct computation consumes $O(n^3)$ complexity (operated on a matrix with $n \times n$-dimension), which is not practical when n is large. Moon *et al.* [182] and [183] presented efficient iteratively-processing solutions offloading expensive computations to the cloud. The aim of [182] is to find the largest eigenvalue and its corresponding eigenvector of an encrypted matrix. The confidentiality of private matrices is guaranteed by Paillier cryptosystem, and the computational result is verifiable. [183] also lightens the client's cost and hides the real values of the involved matrices. Through secure iterations between the cloud and the client, the protocol focuses on finding the approximate top-k eigenvectors and the corresponding eigenvalues of an encrypted matrix.

Matrix factorization is to decompose an original large matrix into the product of several small matrices, the algorithms including triangular factorization, QR factorization, singular value decomposition (SVD), etc. It is a popular method used in recommendation systems to apply the problems to much smaller matrices. Due to this property, people have designed several secure matrix factorization schemes for recommendation functions. Nikolaenko *et al.* [184] were the first to propose a solution realizing matrix factorization over encrypted data, using a combining approach of AHE and GCs. Given a matrix containing users' item-rating pairs and some unobserved vacant elements, the recommendation system and the cloud server predict other unrated entries by executing matrix factorization cooperatively. However, the protocol leaks the number and profile of the rated items. Hence, attackers may infer the preference of the users intentionally. The computational complexity of the protocol is $\Theta(Mlog^2M)$ (where M is the number of ratings), which needs to be further reduced. Achieving an enhanced security level and lower complexities, Kim *et al.* [185] realized privacy-preserving matrix factorization using AHE and FHE schemes as cryptographic methods. Besides, the authors also consider the

security of rated items by adding fake ratings. Moreover, a novel data structure which enables parallel computations is adopted for improved performance. The computation and communication overhead of the work is linear to R/S (R and S are separately the numbers of ratings and the slots the FHE scheme supporting).

D. SYSTEM OF LINEAR EQUATIONS

A system of linear equations (SLE) is a collection of two or more linear equations with the same set of variables. For simplification, the SLE problem is often expressed as the matrix form of $\mathbf{A}\boldsymbol{x} = \boldsymbol{b}$, where $\boldsymbol{x}$ is an unknown vector, with given conditions: a matrix $\mathbf{A}$ and a constant vector $\boldsymbol{b}$. The solution of SLE is to determine the variables of vector $\boldsymbol{x}$ satisfying all the equations simultaneously (if the system is solvable). SLE problem is a fundamental block in linear algebra and works well in modern application fields, such as engineering, computer science, and physics. However, many practical SLE problems are large-scale, which means figuring out the solutions is often a time-consuming and storage-exceeding task. Therefore, it urges customers with weak devices to outsource their expensive SLE tasks to the cloud. The issues of privacy and verifiability in the outsourcing schemes are analyzed in related references.

Considering the matrix form of an SLE problem, denoted as $\mathbf{A}\boldsymbol{x} = \boldsymbol{b}$, we further discuss the solutions on condition that the matrix $\mathbf{A}$ is non-singular or not. Kiltz *et al.* [180] introduced two PHE-based solutions aiming at the non-singular and general case, respectively. When matrix $\mathbf{A}$ is a non-singular square matrix, $\mathbf{A}^{-1}$ can be computed through matrix inversion. The problem is then transformed to determine $\boldsymbol{x} = \mathbf{A}^{-1}\boldsymbol{b}$ in a privacy-preserving way, which can be calculated by $\mathrm{Enc}(\mathbf{A}^{-1})\mathrm{Enc}(\boldsymbol{b}) = \mathrm{Enc}(\mathbf{A}^{-1}\boldsymbol{b}) = \mathrm{Enc}(\boldsymbol{x})$, where Enc is an encryption algorithm of the adopted HE scheme. By decrypting $\mathrm{Enc}(\boldsymbol{x})$, the client learns the vector $\boldsymbol{x}$ privately. Considering the case of general matrix $\mathbf{A}$, the authors adopted a transform-based approach [186] to convert the problem into the non-singular case. The proposed schemes are valid if the SLE problem is solvable. By utilizing AHE scheme, Wang *et al.* [187] proposed a different iterative approach for securely outsourcing SLE problems. The algorithm is to seek successive approximations to the solution continuously until the required accuracy is reached, with only $O(n)$ local computation overhead of each round. The work also raises an efficient cheating-detection mechanism, which allows the clients to verify the correctness of previous-iteration answers with high possibility.

VI. APPLICATION-ORIENTED SECURE COMPUTATION OUTSOURCING

In this section, we give a further overview of existing outsourcing schemes for specific applications, which cover a wide range of practical fields. Like Section V, we only focus on the underlying techniques based on HE.

A. MACHINE LEARNING AND DATA MINING

Machine learning (ML) is an essential part of the field of artificial intelligence (AI), which involves multiple disciplines like probability theory, statistics, convex analysis, and algorithmic complexity theory. It usually requires a certain amount of training data set to build ''knowledge'' (or say ''model'') and applies the trained model to predict new data. Data mining is a process of transforming the original information into valuable structures or rules. The aim is to discover patterns or interesting trends from large-scale datasets. Nowadays, the techniques of machine learning and data mining are served as powerful tools in real life. For example, many business companies have relied on such algorithms to make decisions.

According to the types of training data, machine learning and data mining tasks can be generally categorized into three subclasses: supervised learning, unsupervised learning, and semi-supervised learning. The data of the training set of supervised learning is all labeled, thereby guiding the machine to find the relation between the features and labels. Classical algorithms of the supervised learning include decision trees, naïve Bayesian classification, regression analysis, support vector machine (SVM), k-nearest neighbor (KNN), etc. Unsupervised learning uncovers the inner relationships and patterns by learning from unlabeled training samples. The most studied and used algorithm of such learning tasks is clustering analysis, like k-means clustering. Association rule mining (ARM) is another commonly used method of unsupervised learning. Combining supervised learning and unsupervised learning, semi-supervised learning method makes use of both labeled and unlabeled training samples to improve prediction performance.

Usually, the size of the training data or the model is large, leading to a considerable burden of computation and storage for users. A realistic strategy is to outsource the data and the training/predicting/analyzing tasks to the cloud. The data samples (whether for training or testing) are often sensitive, like personal images or financial information, and the trained models (analyzing structures) are seen as the owner's property. Thus, the issues of data security cannot be ignored. Apart from this, efficiency performance and the correctness of the predicting (analyzing) results are also crucial for the outsourcing schemes. We collect delegating schemes for classic machine learning and data mining algorithms and analyze them in terms of security and efficiency features. The model of outsourced machine learning and data mining tasks is shown in Fig. 4. A summary of the papers surveyed in this subsection is given in Table 3.

1) REGRESSION ALGORITHMS

Regression analysis is a set of algorithms for estimating the relationships among variables. Roughly speaking, given the independent variables, regression analysis aims to estimate the conditional expectation of the corresponding result.

Linear regression is one of the most popular techniques to learn predictive models. In linear regression, the variable

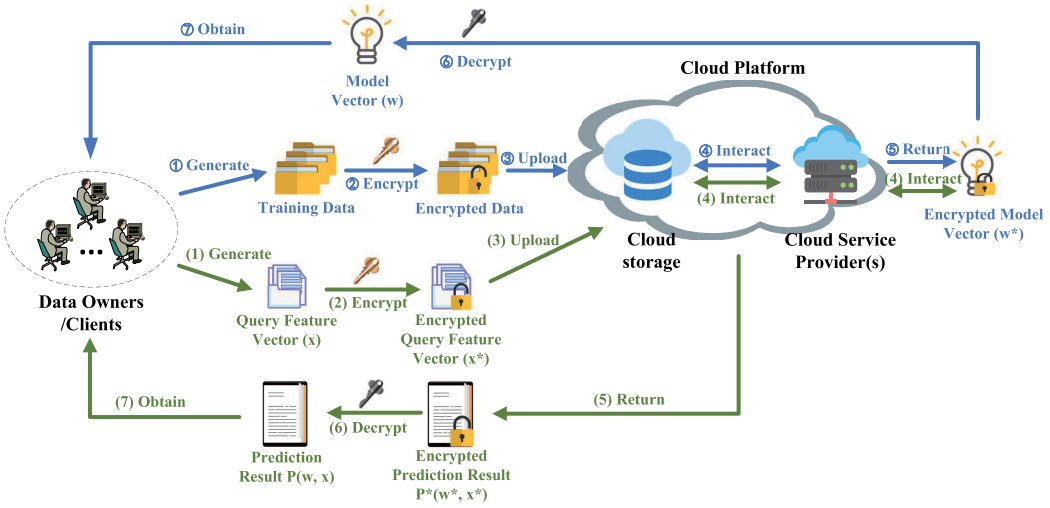


FIGURE 4. The framework of secure outsourced machine learning and data mining tasks. It contains learning phase (labeled with blue) and prediction phase (labeled with green). Assume multiple data owners (or clients) in the system share the same key pairs of HE scheme. In the learning process, the data owners upload encrypted data to the cloud database(s). The encrypted model vector can be obtained by the interactions between the cloud storage and cloud service provider(s). In the prediction process, the data owners upload encrypted query feature vectors to the cloud database(s), which can securely compute their corresponding prediction results based on the trained model parameters.

relation is modeled by training data using a linear approximation function. The linear predictor function $f(\cdot)$ can be expressed as $f(\boldsymbol{x}) = w_1x_1 + w_2x_2 + \cdots + w_nx_n + b$, in which the weights $w_1, \cdots, w_n$ and the constant b are the model parameters, and $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)$ is the input variable. In order to alleviate the over-fitting problem of linear regression, its variants (e.g., ridge and lasso regression) are introduced as improved versions of linear regression by adding a regularization term in loss function for more accurate estimates. Using a secure technique combining both HE scheme and differential privacy, Aono *et al.* [188] proposed a privacy-preserving scheme for linear regression. Receiving the encrypted training data from a client (using his public key), the cloud server performs necessary computations on the encrypted data and returns the encrypted model parameters to the client. Finally, the client recovers the trained model by the decryption. In the protocol, the coefficients of the model's cost function are randomized by adding noises from Laplace distribution, and then be encrypted by an HE scheme (revised from [227]). Moreover, the protocol is also extended for secure ridge and lasso regressions. Nevertheless, the use of differential privacy sacrifices some accuracy of the system results. Morshed *et al.* [189] also developed a secure scheme for linear regression on encrypted data and applied the model for disease prediction. The work is based on a multi-core framework, enabling parallel computations to largely reduce the running time. Another secure linear regression outsourcing framework [190] made use of a vector HE scheme to lower computation and communication overhead. While in the scheme, only the features of training data are encrypted, and the corresponding objective values remain unhidden.

Nikolaenko *et al.* [191] presented a practical system for privacy-preserving ridge regression. The scheme uses a hybrid approach in which HE scheme is applied to calculate the linear processing and Yao's GCs method to handle the heavy non-linear part. By only utilizing linear HE (LHE) scheme, Giacomelli *et al.* [192] also realized secure ridge regression with lower computation and communication complexities. Compared with previous work [191], the work abandons the use of Yao's protocol, and only adopts LHE properties to compute the system of linear equations $\mathbf{A}\boldsymbol{w} = \boldsymbol{b}$, where matrix $\mathbf{A}$ and vector $\boldsymbol{b}$ are the encrypted known parameters, and the vector $\boldsymbol{w}$ is the model parameter which can be securely calculated. Hu *et al.* [193] designed an efficient multiplication protocol over encrypted real numbers utilizing Paillier encryption. Based on the multiplication protocol, the authors further presented a lightweight and privacy-preserving ridge regression scheme. In the solution, the ridge regression training problem is transformed into a system of linear equations, securely solved through Gaussian elimination and Jacobi iterative method.

Logistic regression, another subclass of regression analysis, is a powerful method used to classify data. Given the input vector $\boldsymbol{x}$, the regression output $f_L(\boldsymbol{x})$ is a discrete value indicating the classification result, i.e., $f_L(\boldsymbol{x}) = g(f(\boldsymbol{x}))$. $f(\boldsymbol{x})$ is the linear function just like the expression of the linear regression, and $g(\cdot)$ is usually the sigmoid function $g(z) = \frac{1}{1+e^{-z}}$. A secure outsourcing scheme called homomorphism-aware logistic regression via function approximations was given in [194]. The storage and computation overhead is $O(Nd^2)$ at the server side, while light-weight $O(d^2)$ at the client (N is the number of records and d is the dimension of each record). However, when

TABLE 3. An overview of the surveyed machine learning and data mining literature.

Task		Scheme	Secure techniques	Threat model	Verifiability
Regression analysis	Linear or ridge regression	[188]	PHE+DP	semi-honest	No
		[189]	SWHE	semi-honest	No
		[190]	Vector HE	semi-honest	No
		[191]	Linearly HE+GCs	malicious	Yes
		[192]	Linearly HE	semi-honest	No
		[193]	PHE	semi-honest	No
	Logistic regression	[194]	PHE	semi-honest	No
		[195]	SWHE	semi-honest	No
		[196]	Approximate HE	semi-honest	No
		[197]	Approximate HE	semi-honest	No
		[198]	SWHE+SGX	semi-honest	No
Classification algorithms	Support vector machine	[199]	PHE+SS+GCs	semi-honest	No
		[200]	PHE	semi-honest	No
		[201]	PHE	semi-honest	No
		[202]	PHE	semi-honest	No
		[203]	PHE	semi-honest	No
	Other algorithms, like naïve Bayesian and decision trees	[204]	FHE	semi-honest	verify the learned model probabilistically
		[205]	PHE	semi-honest	No
		[206]	PHE+OT	semi-honest	No
		[207]	FHE	semi-honest	No
		[208]	PHE	semi-honest	No
		[7]	PHE	semi-honest	No
		[209]	PHE	semi-honest	No
Artificial Neural networks		[210]	FHE	semi-honest	No
		[211]	FHE	semi-honest	No
		[212]	PHE+SS	semi-honest	No
		[213]	FHE	semi-honest	No
		[214]	PHE	semi-honest	No
Data mining	K-means clustering and its variants	[215]	PHE	semi-honest	No
		[216]	FHE	semi-honest	No
		[217]	PHE+FHE	semi-honest	No
		[218]	FHE	semi-honest	No
		[219]	FHE	semi-honest	No
		[220]	FHE	semi-honest	No
		[221]	FHE	semi-honest	No
		[222]	Approximate HE	semi-honest	No
	Association rule mining	[223]	PHE	semi-honest	No
		[224]	PHE	semi-honest	No
		[225]	SWHE	semi-honest	Yes
		[226]	PHE	semi-honest	No

the data dimension rises, the result of the proposed system tends to show a lower accuracy than the scheme without cryptosystem. Bonte and Vercauteren [195] constructed a secure homomorphic logistic regression learning scheme on the encrypted data. The work adopts a light-weight iterative method which simplifies the standard Hessian method. Another privacy-preserving scheme for logistic regression learning was proposed by Kim *et al.* [196], based on the cryptosystem of approximate homomorphic encryption [228]. To find the local extremum of the function, the protocol uses the Nesterov's accelerated gradient method [229] with a better convergence rate (compared with the typical gradient descent method). Moreover, the work also introduces an encoding method to reduce the storage burden. Precisely, the method is to encrypt a matrix-represented training dataset into a single ciphertext. Similarly, Cheon *et al.* [197] utilized a new ensemble gradient descent method to reduce the iteration number of logistic regression learning. Through a hybrid cryptographic framework combining both SWHE and SGX techniques, Sadat *et al.* [198] presented a scheme for performing regression analysis, achieving a better balance between security and efficiency. In the protocol, some complex calculations (which cannot be efficiently handled by existing HE schemes, like the matrix inversion and division) are performed in the plaintext form inside secure hardware at the server side. Moreover, the accuracy quality of the final result is guaranteed.

2) CLASSIFICATION ALGORITHMS

Classification is to identify the category of a new sample, based on the sorting model learned from the empirical data set. There are multiple outstanding classification algorithms, such as logistic regression (which was discussed before), support vector machine (SVM), decision trees, etc. Classifiers are widely used in numerous fields, like computer vision, statistics, and biometric identification.

Graepel *et al.* [204] presented solutions to realize confidential machine learning on encrypted data based on leveled FHE scheme. The work focuses on solving binary classifications, like linear means (LM) classifier and Fisher's linear

discriminant (FLD) classifier, using low-degree polynomial approximations. Bost *et al.* [205] constructed secure homomorphic protocols for achieving three classification algorithms: hyperplane decision, naïve Bayesian classification, and decision trees. Moreover, the authors also analyzed a more general classification function using AdaBoost technology [230]. Gao *et al.* [206] proposed a secure scheme for constructing naïve Bayesian classifier. In the work, by combining a novel ''double-blinding'' technique, the AHE scheme, and oblivious transfers, the privacy of both the client and the server can be protected. Moreover, most computations of the protocol are executed offline by the server, thus demanding less overhead of online computation and communication for both parties.

By employing an optimized ring LWE-based variant of HE scheme, Khedr *et al.* [207] introduced a secure scheme for practical Bayesian application: encrypted Bayesian spam filters [231]. Using Bayesian rule, the proposed spam filter determines whether the given email is spam by computing over its included encrypted words. Based on the hardware-assisted PHE-based technique, Bian *et al.* [208] also proposed a secure email filter system based on naïve Bayesian filter. To reduce the number of homomorphic operations, two optimization approaches are adopted. The first one is the weight-embedding technique, which simply embeds rounded exponent weight (instead of a fixed-point pattern) into the result. The second is the batch filtering technique, allowing for packing more bits (from the binary-decomposed words) into one ciphertext. Through the hardware implementation, one email with an average-length can be classified in 0.5 second. Another realistic example based on naïve Bayesian classification was proposed in [7], achieving a patient-centric clinical treatment system in a privacy-preserving way. By utilizing an elaborate AHE scheme, the naïve Bayesian classifier is trained using historical clinical data in the encrypted form. After that, the trained model can be applied to homomorphically predict k most possible diseases for a new patient. Adapted from the work [7], Alabdulkarim *et al.* [209] also implemented a secure clinical decision system utilizing decision tree algorithm. The simulation experiment shows that the proposed scheme achieves better results than the one using naïve Bayesian algorithm [7].

SVM method constructs a hyperplane or a set of hyperplanes in a high (or infinite) dimensional space to divide samples. The algorithms efficiently perform linear classification and are further extended for non-linear classification using kernel tricks. Laur *et al.* [199] proposed privacy-preserving protocols for the evaluation of kernel matrix and the processes of kernel-based classifier's training and prediction, using cryptographic techniques of AHE, secret sharing, and secure circuit evaluation. After the training phase, each client holds a secret share of the trained model. While in the testing phase, all of the clients need to collaboratively perform the classification using their individual model shares. The computation requires the participation of all the client parties, hence only suitable for certain particular scenarios. González-Serrano *et al.* [200] proposed a privacy-preserving scheme for learning SVM over multiple distributed data. The work utilizes an AHE scheme of Bresson, Catalano, and Pointcheval (BCP) cryptosystem [232] with multiple keys. The algorithm is robust against finite word-length effects. Based on the privacy-preserving outsourced calculation framework of [157], Sun *et al.* [201] also constructed secure protocols for SVM training and predicting over multiple encrypted domains. To protect data privacy, two servers interact multiple rounds with each other to perform the tasks on the multi-key encrypted data. Scheme [202] was the first to realize privacy-preserving SVM protocols for two-class and multi-class classification. In the framework, the server trains the classifier and the client uploads his sample to get the corresponding classified label as a service. Compared with the conventional approach (without using cryptosystem), the result of the proposed scheme is accurate to an equal degree. Applying the SVM algorithm to practice, Liu *et al.* [203] proposed an outsourced drug discovery framework based on their designed secure SVM protocols. In the scheme, the decision model is trained using multiple drug formulas, where a secure sequential minimal optimization algorithm is adopted to refresh the model parameters. By the privacy-preserving computations at the cloud server, users determine the activity property of given chemical compounds from the trained model.

3) ARTIFICIAL NEURAL NETWORKS

Machine learning based on artificial neural networks (ANNs) has dramatically pushed the advance of AI. Simulating the nervous system in the biological brain, the framework of ANN contains a set of connected units or nodes. Deep neural network (DNN) with multiple hidden layers between the input and output layers is one kind of ANNs. Recurrent neural network (RNN) and convolutional neural network (CNN) are two practical examples of DNNs. To substantially cut down the burden of clients, training and applying ANN models are usually performed at cloud servers. For privacy requirements, the servers should never know any private data during executions.

Scheme [210] and [211] achieved privacy-preserving prediction of neural networks over encrypted data. In the protocol, the client sends the encrypted sample feature (by HE algorithm) to the cloud for the prediction from the trained model. Both the input data and the prediction result are confidential to the cloud. Ma *et al.* [212] proposed the first fully non-interactive (between the cloud servers and the client) neural network prediction scheme. In the protocol, the trained model is split into two random parts using secret sharing technique, which are separately sent to two non-colluding servers. Due to additive privacy homomorphism, the servers apply neural networks interactively on the client's input data (encrypted by an AHE scheme) and return the encrypted prediction shares to the client. Finally, the client decrypts and recovers the corresponding prediction of his data sample by combining the result shares. The computation and

communication overhead at the client's side is independent of the size of the neural network model.

The works [210], [211], and [212] only focused on the prediction stage, assuming the neural network model was previously trained. Considering the training phase of neural networks, CryptoDL [213] was designed to run the well-known CNN algorithms on encrypted data securely. To Break the limitations from HE algorithms, the protocol approximates the activation functions with low-degree polynomials and trains the CNN model using the approximation polynomials. The trained model is then implemented over encrypted data for prediction. Tang *et al.* [214] proposed a distributed deep learning scheme with security guarantees and high accuracy. In the protocol, data requesters outsource the encrypted gradients to the data service provider for a new round of updates of the model weights and request the newly updated values. A new party (i.e., key transform server) is adopted to re-encrypt the encrypted gradients. Meanwhile, the data service provider makes the re-encrypted gradients additively homomorphic and performs the updating computations on the ciphertexts. At last, each data requester obtains the updated weights and decrypts them. In the algorithm, extra communication cost is added, yet still tolerable.

4) CLUSTERING AND ASSOCIATION RULE MINING

Clustering is a process of performing grouping tasks for a collection of data. The items in the same group are similar and are relatively different from other groups'. The algorithms are available to handle numerous data of real-world fields, like in commerce, biology, social network, etc. We discuss the proposed HE-based schemes on k-means clustering (as well as its variants). Depending on the similarity measure, k-means clustering runs a series of iteration procedures to group the analyzed data into k clusters. Samanthula *et al.* [215] proposed a secure k-means clustering scheme under multi-user setting. In the system architecture, multiple users (i.e., data owners) outsource their encrypted data to obtain the combined clustering result, and two non-colluding cloud servers are in charge of the total clustering computations. The servers generate new clusters iteratively based on HE properties until the termination condition is satisfied. The main contributions of the proposed work are two folds. On the one hand, an efficient transformation method was developed to enable clustering processing to operate over encrypted integers (instead of fractions). On the other hand, the servers achieve a secure comparison of Euclidean distances (between data records and current clusters) based on the order-preserving property. Similarly, in [216] the encrypted distance values are also comparable due to the use of order-preserving encryption (OPE) together with trapdoor information. Different from [215], the system model only contains one data owner and one cloud server. In each k-means clustering iteration, the data owner generates the new trapdoor information from updated cluster centers. Hence, the clustering processes require certain computation and communication burdens for the data owner. An extending homomorphic scheme of [216] was designed by Liu *et al.* [217], involving two data owners holding horizontal partitions of the dataset. In each clustering round, the server allocates each encrypted data record to the nearest cluster, and the two data owners compute new cluster centers and uploads the encrypted clusters to the server for the next clustering operation. The iterative processes will terminate if the assignments of data records have not changed.

Based on an LWE-based homomorphic cryptosystem, Theodouli *et al.* [218] introduced a privacy-preserving k-means clustering framework, and also described three user-server interactive algorithms for computing new clusters. Receiving the encrypted distances (between data records and current clusters), the client decrypts the distances and compares the distances to obtain the individual minimum for each data record. By the comparison results, the new clusters are computed. In the first version of the proposed algorithms, the new clusters are found by the user, while incurring plenty of computation and storage burdens for the user. In the second version, the client sends unencrypted cluster identifiers which have minimum distances to the server. The server homomorphically computes the new clusters with the assistance of the client. The design decreases the client's computational and storage complexities, while with some information leakage. To solve the privacy flaw, the authors further presented the third version, by loading more computations to the server to protect the confidentiality of the cluster identifiers. The three algorithms achieve different tradeoffs between the security and the consumed resources of the client and the server. In practical applications, the above schemes [216]–[218] still require nonnegligible computation burdens for the users, whether for computing the trapdoor information or decrypting all distance values. To reduce the participation amount of the users, Almutairi *et al.* [219] proposed a homomorphic k-means solution by employing a structure called updatable distance matrix (UDM) for storing the information of data records. During the algorithm processes, the server needs to update the UDM for further clustering in each iteration. For updating the UDM, the server calculates the encrypted differences between the clusters of the previous and current rounds and sends them to the user. The user only needs to decrypt the encrypted differences and organize them into a shift matrix, which is then uploaded to the server for UDM update. The computation and storage overhead of the proposed scheme is lower, compared to the previous works. The authors of [220] introduced an FHE-based k-means clustering protocol also with a low client workload. In the secure solution, to lighten the clients burden, the comparison operations (which are initially performed by the client) are completed by an additional entity-a trusted and auditable server.

The variations of k-means algorithm also have outstanding clustering performance and can be securely applied in the cloud computing environment. Zhang *et al.* [221] employed the BGV cryptosystem to design a privacy-preserving weighted possibilistic c-means algorithm. Compared with k-means clustering, this variant algorithm considers the

TABLE 4. An overview of the surveyed image processing literature.

Task		Scheme	Secure technique	Characteristics or underlying algorithms
Image feature extraction	SIFT (scale-invariant feature transform)	[233]	PHE	Image contents can be deduced by the cloud server.
		[234]	PHE	
		[235]	SWHE	Preserve the robustness and distinctiveness of the original SIFT scheme.
		[236]	PHE	Protect the confidentiality of the location of image keypoints.
	SURF (speeded-up robust features)	[237]	PHE	Execute interest points extraction, accurate keypoint localization, and descriptor extraction over the encrypted image domain.
		[238]	SWHE	Preserve the robustness and distinctiveness of the original SURF scheme.
	HOG (histogram of oriented gradients)	[239]	SWHE	Two solutions based on single-server and two-server models, respectively.
		[240]	Vector HE	More efficient.
		[241]	FHE	Higher accuracy.
Content-based image retrieval		[242]	PHE	Image color, texture and shape features for similarity measurement.
		[243]	PHE	Discrete wavelet transform (DWT) histograms for similarity measurement.
		[244]	FHE	Achieve fine-grained access control.
Image watermarking		[245]	PHE	Walsh-Hadamard transform (WHT) based method.
		[246]	PHE	Discrete wavelet transform (DWT) based method.
		[247]	PHE	Discrete wavelet transform (DWT) and discrete cosine transform (DCT) based method.
		[248]	PHE	Discrete wavelet transform (DWT) and discrete cosine transform (DCT) based method.
		[249]	PHE	Adaptively scanned wavelet difference reduction (ASWDR) compression [250] based method.

membership of every point and achieves soft clustering. However, the exponential and division operations of the algorithm cannot be directly evaluated using FHE scheme. Hence, the scheme adopts Taylor theorem to approximate the complex functions to the polynomials containing additions and multiplications. A homomorphic version of the mean-shift clustering algorithm (which is a non-parametric clustering technique) was designed by Cheon *et al.* [222]. During the mean-shifting, the protocol executes over several randomly sampled points instead of all the points, which reduces the computation complexity of quadratic to linear. To enable the efficient utilization of homomorphic cryptosystem, non-polynomial kernels of the original algorithm are replaced by a polynomial kernel function.

Association rule mining (ARM) is a method to discover valuable correlations of certain dimensions amongst a large collection of data. Once a rule $X \rightarrow Y$ (X and Y are respectively the antecedent and consequent of the correlation rule) satisfies the threshold of both the support and confidence parameters, then it can be considered as an ''interesting'' association rule. Based on the ElGamal encryption scheme, Liu *et al.* [223] proposed a secure outsourcing ARM scheme over encrypted data. In the protocol, data owners encrypt their data and send them to the cloud server. If an initiator launches an ARM request, the server will homomorphically perform necessary computations for computing frequent itemsets by interacting with the involved data owners. After that, the data owners query whether a given rule holds or not. The authors first designed an algorithm involving only one data owner, then extended it to support the scenario for multiple data owners. The original data and the miming results are confidential to the server. Yi *et al.* [224] adopted a N-server computing framework ($N \geq 2$) for collectively accomplishing ARM tasks based on partially homomorphic property. In the framework, the data owner outsources its encrypted data to a database and chooses several data mining servers for the service of ARM computations. For achieving different security levels, the authors described three secure ARM solutions for protecting the privacy of the involved items, transactions, and database, respectively. The security features can be realized when at least one server behaves honestly. Supporting vertically partitioned databases, an efficient solution for ARM tasks was presented by Li *et al.* [225] through the use of symmetric HE scheme. However, the scheme requires the users to stay online during the processing and reveals some privacy of the raw data. Removing these limitations, Liu *et al.* [226] proposed a homomorphic outsourcing ARM scheme with multiple encrypted keys. The protocol protects data confidentiality using BCP cryptosystem [232].

B. IMAGE PROCESSING

With the size and the number of images rapidly growing, outsourcing the tasks of image processing is a promising choice. Since users' image data (no matter the original images or the processed images) is usually sensitive, achieving efficient image computations which satisfy the security requirements has become a popular research direction. We discuss existing outsourcing schemes for three specific image processing tasks (i.e., image feature extraction, content-based image retrieval, and image watermarking) and draw the corresponding algorithm flowcharts (see Fig. 5-7).

1) IMAGE FEATURE EXTRACTION

Image feature extraction is a significant procedure through image analysis, processing, and recognition. It aims to extract useful features from original image data, as an expression or a description of the analyzed image. Image extraction algorithms have found extensive application scenarios, such as cloud-assisted e-healthcare system [251] and biometric system [252]. As follows, we analyze secure outsourcing schemes for three popular image feature extraction algorithms: scale-invariant feature transform (SIFT) [253], speeded-up robust features (SURF) [254], and histogram of oriented gradients (HOG) [255].

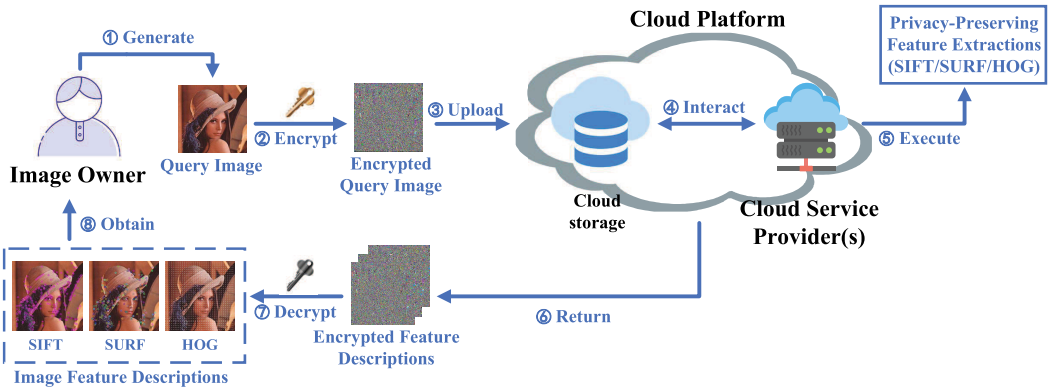


FIGURE 5. The framework of secure outsourced image feature extraction tasks. After privacy-preserving image processing for the task of SIFT (scale-invariant feature transform), SURF (speeded-up robust features), or HOG (histogram of oriented gradients), the cloud platform returns feature extraction results to the image owner.

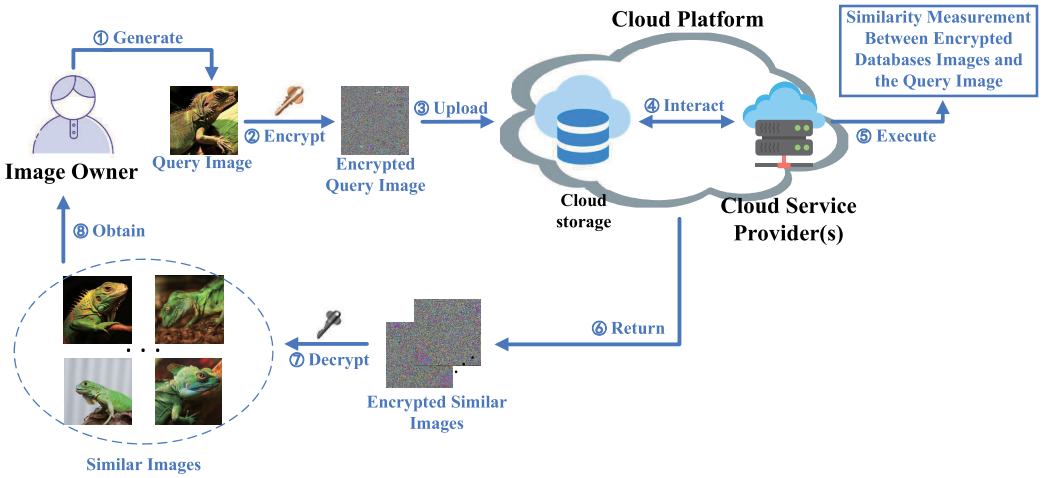


FIGURE 6. The framework of secure outsourced content-based image retrieval task. After privacy-preserving similarity measurement between image samples, the cloud platform returns image results similar to the query image submitted by the owner.

SIFT is an algorithm used to detect and describe local features in the image, with a powerful attack-resilient feature point detection mechanism. Hsu *et al.* [233] presented secure and robust outsourcing protocols for SIFT computation, achieving SIFT feature extraction and representation in the encrypted domain. The proposed algorithm contains four major parts: difference-of-Gaussian (DoG) transform, feature point detection, feature description, and descriptor matching, which are all executed in the ciphertext form by means of Paillier cryptosystem. Extending the work, Hsu *et al.* [234] further explored a similar HE-based secure SIFT outsourced scheme. The algorithm is secure against ciphertext-only attack (COA) and known-plaintext attack (KPA), based on the discrete logarithm problem and RSA problem. However, the works [233] and [234] introduce a large computation complexity and certain insecurities from the privacy perspective [256]. To remove these limitations, an advanced protocol was proposed in [235]. Instead of encrypting the initial image by Paillier cryptosystem, the work first additively splits the original image into two random shares and uploads the encrypted sub-images to two independent cloud servers. The comparison process is improved by an SWHE scheme together with the batch technique of SIMD. Besides, the privacy-preserving SIFT scheme well preserves important characteristics like the original SIFT scheme (without cryptosystem) concerning distinctiveness and robustness. Providing a stronger privacy, Li *et al.* [236] presented another secure SIFT feature extraction scheme, by means of Paillier cryptosystem with partial decryption (PCPD) of [152].

SURF is reckoned as an enhanced version of SIFT. It can be performed faster and be more robust against different image transformations than SIFT. The steps and theories of SURF algorithm are basically identical with the ones of SIFT, with some different details (such as scale space, feature point detection and direction determination, and feature descriptors) between them. For example, the SIFT algorithm

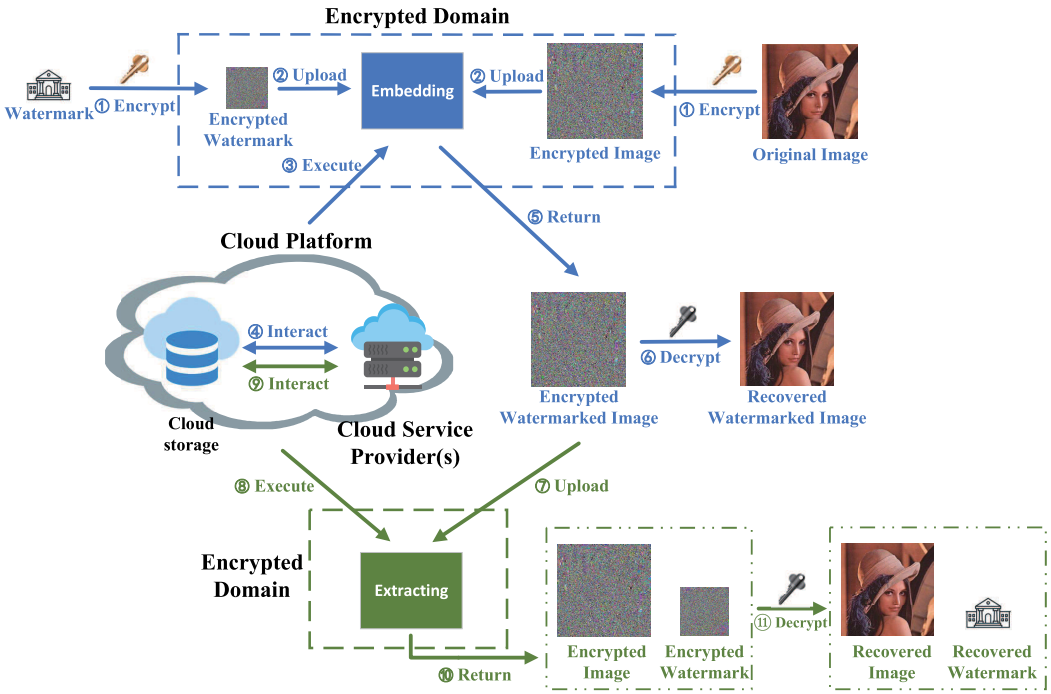


FIGURE 7. The framework of secure outsourced image watermarking tasks. It contains the processes of encrypted watermark embedding (labeled with blue) and extracting (labeled with green).

detects feature points by finding local extreme points in the scale space of DoG, while the SURF algorithm detects by computing determinant of the constructed Hessian matrix. Bai *et al.* [237] presented an outsourcing solution for SURF feature extraction, executed in the encrypted domain. The computations are supported by HE properties of Paillier cryptosystem. However, since the operations require multiple interactions between the client and the server, considerable communication overhead is generated. Apart from this, it also has a poor ability to preserve key characteristics of the original SURF. Motivated by these observations, Wang *et al.* [238] introduced a practical outsourcing protocol for SURF computations. The idea of the work is similar to [235], which employs two non-colluding servers to compute the encrypted feature descriptors of the input image jointly. In the algorithm, efficient interactive sub-protocols of multiplication and comparison operations are designed based on SWHE and SIMD techniques, which not only supports the secure computations but reduces the overall communication overhead.

HOG is another image feature descriptor widely used in computer vision and image processing fields, which is formed by calculating gradient orientation histograms of local regions. Wang *et al.* [239] designed secure outsourcing schemes for HOG computations. The work introduces privacy-preserving protocols for HOG computations in the encrypted domains under two different models: single-server and two-server settings. For the single-server model, the original image is encrypted utilizing SWHE integrated with SIMD technique, achieving both security and efficiency requirements. The encrypted feature descriptors are securely calculated and returned to the client. For the two-server model, the image is first split randomly into two shares, and the encrypted shares are then sent to two independent servers separately. After that, the two servers jointly compute their individual encrypted feature descriptors. In the last step, the client decrypts and recovers the feature descriptors combining both portions returned from the servers. Utilizing a more efficient homomorphic method (i.e., vector homomorphic encryption (VHE) [257]), Yang *et al.* [240] also proposed a privacy-preserving scheme for extracting HOG features. The encryption is performed directly on the image vectors, which can be well applied for image processing. Based on an FHE scheme, Shortell and Shokoufandeh [241] also designed a secure framework enabling SURF and HOG computations in the encrypted domain. In the protocols, SURF and HOG tasks are implemented over rational and fixed-point binary numbers, respectively. Experimental evaluations demonstrated that the proposed solutions [239], [240], and [241] reach comparable performance to the original HOG solution.

2) CONTENT-BASED IMAGE RETRIEVAL

CBIR (content-based image retrieval) is a practical image retrieval technique for the application of image analysis. Instead of exploiting the searching indexes like keywords, tags, or descriptions of the images, CBIR carries out the task of image retrieval directly based on the image contents. The similarity measure among the images is a decisive parameter in the algorithm. For practical considerations, the image data and CBIR tasks are usually outsourced to the cloud. Thus,

it is essential to protect the image contents and obtain the matching results as in the traditional CBIR method.

Zhang *et al.* [242] proposed a secure outsourcing image retrieval method based on the CBIR framework. Given an input image, to search for similar images in the cloud requires similarity measures between image features of the input and database image samples. The first step of the algorithm is to extract three kinds of features from the input image, including color, texture, and shape features. After that, the features are encrypted by Paillier cryptosystem for security and sent to the cloud. The cloud computes the similarities of the encrypted features between the input image and the image samples of the database, exploiting the properties of the HE scheme. After sorting the encrypted similarity results, the cloud returns the most similar images to the client. The retrieval performance is almost consistent with the conventional CBIR method. Supposing there are N images with M-dimension features in the database, the proposed algorithm implements $O(MN)$ times of multiplication and exponentiation operations. Unlike the work [242] which exploits local image features, the method in [243] was the first secure CBIR scheme over global image features, based on the wavelet transform [258]. The operations of image retrieval are performed by the cloud server on the encrypted domain, due to the additively homomorphic property of the Paillier cryptosystem. Supporting image sharing among multiple users, Zhang *et al.* [244] presented a secure and efficient outsourcing CBIR scheme with fine-grained access control. In the scheme, the users are only capable of searching for specific images which are authorized by the corresponding image owners. To avoid revealing the privacy of image data, the work uses a lightweight multi-level HE algorithm [259] as a building block to support the operations in the encrypted form. Moreover, the scheme is speeded up with efficient methods of distributed and parallel computation.

3) IMAGE WATERMARKING

A digit watermark is a signal typically indicating the copyright information, which is usually embedded in media data. Like traditional physical watermarks, digit watermarks are visible in certain conditions. Therefore, people have applied the technology to fields like copyright protection, source tracking, and content management on networks. When the task of image watermarking is outsourced, the confidential of the original and watermark images, as well as the watermarked images, should be guaranteed.

Since the images on the cloud are often encrypted for privacy, how to embed/extract encrypted watermarks to/from the encrypted images without decryption is a challenging issue. Zheng and Huang [245] raised one of the solutions, based on Walsh-Hadamard transform (WHT) [260]. Exploiting HE properties, the algorithm enables the encrypted watermark to be embedded over the encrypted input image. In addition, the authors proposed secure protocols for blind watermark extraction in both decrypted and encrypted domains. Subramanyam *et al.* [246] introduced a robust algorithm for watermarking encrypted, compressed JPEG2000 images based on HE scheme and discrete wavelet transform (DWT). Moreover, the work allows image watermarking detection in the compressed or decompressed domain. Guo *et al.* [247] presented another robust watermarking scheme by combining the discrete wavelet transform (DWT) and discrete cosine transform (DCT) methods in the encrypted domain. In the work, operations on the encrypted images are supported by Paillier cryptosystem. Based on the encrypted DWT-DCT domain, Priya *et al.* [248] also constructed a robust scheme for image watermarking computations (consisting of watermarking embedding and extraction). The watermark images are protected through the shuffling of the image pixel positions. Meanwhile, the original images and shuffled watermark images are encrypted by Paillier homomorphic cryptosystem. Likewise, based on additive privacy homomorphism, Mishra *et al.* [249] designed an image watermarking scheme in the compressed domain. It achieves robustness concerning selected image processing attacks.

C. BIOMETRIC COMPUTATION

Recent advances in biometric computations and increasing use of biometric data prompt the evolvement of medicine and biology. According to the application scenarios, different biometric identifiers are used. Specially, personal genomic data is used in the field of genome analysis. In addition, the tasks of biometric authentication are based on different biometric characteristics, like facial features, DNA, iris, and fingerprint. Obviously, the computation and storage burdens for large-scale biometric tasks are too heavy for local devices. Thus, the cloud servers are always responsible for implementing the biometric computations as well as storing biometric data. However, due to the sensitiveness of personal biometric data, the security issues (like the possible reveal or misuse of biometric data) cannot be ignored. In this subsection, we analyze the existing privacy-preserving outsourcing schemes for genome analysis and biometric authentication. A summary of the papers surveyed is given in Table 5.

1) SECURE GENOMIC DATA ANALYSIS

With the techniques of genetic testing and analysis continuously developing, various research on genomic data has attracted much attention from the public, like disease prediction and treatment. Nowadays, numerous researchers have designed secure outsourcing solutions for different genome analysis techniques via HE schemes.

Genome-wide association study (GWAS) technique is a common approach applied in genetics. It focuses on finding sequence variations at the genome-wide level to filter the single nucleotide polymorphisms (SNPs) related to particular diseases, thereby predicting some potential diseases. Zhang *et al.* [261] presented a framework to securely outsource GWAS task based on HE algorithms. Given two groups' genotypes containing several SNPs, the algorithm calculates the chi-square statistic χ^2 homomorphically

TABLE 5. An overview of the surveyed biometric computation literature.

Task		Scheme	Secure techniques	Threat model
Secure genome analysis	Genome-wide association study	[261]	FHE	semi-honest
		[262]	FHE	semi-honest
		[263]	FHE	semi-honest
		[264]	SWHE	semi-honest
		[265]	PHE+SGX	semi-honest
	Whole genome sequencing	[266]	PHE+FHE	semi-honest
	Edit distance or Hamming distance of genetic data	[267]	SWHE	semi-honest
		[268]	SWHE	semi-honest
	Pattern matching on genetic data	[269]	SWHE	semi-honest
Biometric authentication	Biometric authentication	[270]	PHE+GCs	semi-honest
		[271]	PHE+GCs	semi-honest
		[272]	SE+PBM	semi-honest single server
			SWHE+PBM	semi-honest two servers
		[273]	PHE	semi-honest
	Face recognition	[274]	PHE+PBM	semi-honest
		[36]	PHE+OT+GCs	semi-honest
		[275]	SWHE	semi-honest
		[276]	PHE+OT	semi-honest
		[277]	FHE	semi-honest
	Fingerprint or iris recognition	[278]	PHE	semi-honest
		[279]	PHE	semi-honest
		[280]	PHE+OT+GCs	semi-honest

between them. Suppose the symbol $O_{i,j}$ is the observed jth allele count from the case group ($i = 1$) or the control group ($i = 2$), and $E_{i,j}$ is the expected one. Typically, χ^2 can be securely calculated as:

$$\chi^2 = \sum_i \sum_j \frac{(O_{i,j} - E_{i,j})^2}{E_{i,j}}$$

Meanwhile, two main division protocols (i.e., errorless division protocol and approximation division protocol) are executed over homomorphically encrypted data in the algorithm. Moreover, parallel computation is also supported to improve the system efficiency. Another scheme for secure outsourcing GWAS computation was proposed by Lu *et al.* [262]. Exploiting the ring-LWE-based HE scheme, the cloud efficiently calculates χ^2 hypothesis testing at the client's request without learning any sensitive information. Extending the work [262], a general solution supporting more types of genomic hypothesis testing was developed in [263]. The protocol evaluates frequency tables to calculate typical GWAS-related statistics, based on the encrypted genomic data. Using the packing technique associated with FHE, the work presents a practical scheme for secure outsourcing χ^2 test, Hardy-Weinberg equilibrium (HWE), and linkage disequilibrium (LD) on genomic data. A cryptographic version of outsourced Fisher's exact test was designed by Poon *et al.* [264]. By using BGN's encryption algorithm, the test is securely executed on the encrypted genomic data. Utilizing a secure hybrid technique, which combines Paillier cryptosystem and Intel SGX, Sadat [265] also proposed a secure outsourcing GWAS scheme. The work performs four statistical tests: linkage disequilibrium (LD), Hardy-Weinberg equilibrium (HWE), Cochran-Armitage test for trend (CATT), and Fisher's exact test (FET) over federated encrypted genomic datasets.

Another popular analyzing method on genomic data, called whole genome sequencing (WGS), has also been analyzed in many related works. It refers to the process of analyzing structural differences between different individual genomes using the bioinformatics. Ziegeldorf et al. [266] proposed solutions for genetic disease testing based on the bloom filter. The authors employed two cryptographic methods (i.e., FHE and PHE) to design the secure algorithms ensuring data privacy. For the FHE-based scheme, the data owner first encodes the SNPs data in the patient database as bloom filters and uploads the encrypted ones (i.e., $\mathrm{Enc}(B_i)$, where $\mathrm{Enc}(\cdot)$ denotes the encryption algorithm satisfying fully homomorphism and B_i ($1 \leq i \leq n$) represents the bloom filter for each patient) to the cloud. In the online phase, the client transforms his query into a bloom filter (called Q) likewise. He then generates a key pair and consequently encrypts the query bloom filter, denoted by $\mathrm{Enc}(Q)$, and uploads it to the cloud. After that, the cloud executes the matching operation by computing $\mathrm{Enc}(B_i) \odot \mathrm{Enc}(Q)$, where $\odot$ is the FHE component-wise multiplication operation, and returns the aggregated results to the client. Finally, the client decrypts the results and reconstructs the matching list. While for the PHE-based scheme, the bloom filter is confused using a keyed hashing method, in order to speed up the matching process and compress the packing results. The FHE-based protocol was proved to be secure in the semi-honest setting while the PHE-based protocol makes a slight sacrifice of the access pattern privacy but achieves much improved performance.

A solution focusing on calculating edit distances on the encrypted genomic data was proposed by Cheon *et al.* [267]. For two given strings, the edit distance is the minimum number of single-character editing operations (i.e., insertion, deletion, and substitution) required to convert one string to the other. Receiving two encrypted genomic sequences,

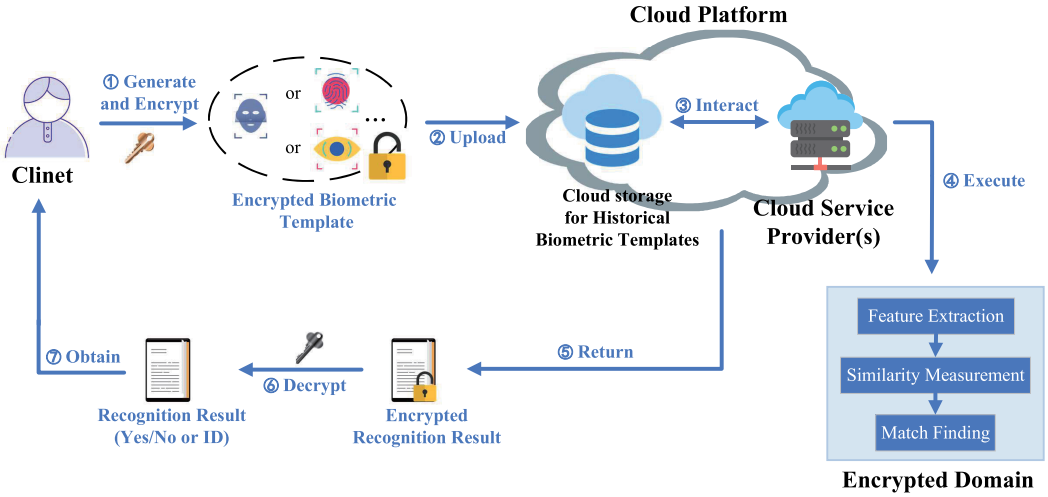


FIGURE 8. The framework of secure outsourced biometric authentication tasks. Given a certain encrypted biometric template of a client, the system carries out biometric authentication computations with the assistance of cloud storage and cloud service provider(s).

the proposed protocol calculates their edit distance in the encrypted form with an SWHE scheme. Besides, an optimized scheme was also presented to reduce the circuit depth in the algorithm. The key idea is to divide the edit distance matrix into sub-blocks, calculate the edit distance inside each block, and then integrate all the results. Kim and Lauter [268] also presented efficient protocols for outsourcing genomic testing computations. In addition to executing basic genomic tests for GWAS, the work considers secure comparisons of encrypted DNA sequences based on their Hamming distance and approximate edit distance. Particularly, the authors implemented the secure protocols by the utilization of the BGV scheme [139] and the YASHE scheme [281], respectively. By comparison, the BGV-based scheme has better performance than the YASHE-based scheme. Pattern matching of DNA sequences is a practical approach to search specific DNA sequences in a genome database. With the symmetric-key variant scheme of SWHE [177] and a packing method, Yasuda *et al.* [269] proposed privacy-preserving protocols for pattern matching and implemented the secure wildcards pattern matching (i.e., wildcard characters can be included in the queried pattern) on DNA sequences. The proposed protocol has a favorable performance with low communication complexity.

2) BIOMETRIC AUTHENTICATION

Biometric authentication is such a technique that utilizes the inherent physiological and behavioral characteristics of humans in the goal of identification or access control. By comparing the biometric identifiers between the target sample and the samples in the database, the system succeeds to identify the individual if the comparison result falls within a certain limit. In this subsection, We discuss several HE-based schemes for biometric authentication in the outsourcing environment. The factors should be considered include efficiency performance, accuracy, cost, as well as data privacy. Hence, researchers have devised efficient outsourcing schemes for diverse biometric authentication types, which protect users' sensitive information simultaneously. Fig. 8 shows the general system model of the related schemes.

Utilizing a hybrid approach of AHE and GCs, Chun *et al.* [270] presented a privacy-preserving scheme to outsource the tasks of biometric authentication. The work employs a cloud server to store the encrypted biometric data and another independent server to keep the decryption key. The two servers operate interactively during the protocol, and neither of them will learn the sensitive biometric information and the intermediate results. Given vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ (both with length m), Euclidean distance can be computed as:

$$ED(\boldsymbol{x}, \boldsymbol{y}) = \sqrt{\sum_{i=1}^{m} (\boldsymbol{x}(i) - \boldsymbol{y}(i))^2}$$

Hamming distance can be obtained from:

$$HD(\boldsymbol{x}, \boldsymbol{y}) = m - \sum_{i=1}^{m} (\boldsymbol{x}(i) \cdot \boldsymbol{y}(i))$$

Suppose $\boldsymbol{w}_i$ $(1 \leq i \leq n)$ and $\boldsymbol{u}$ are biometric feature vectors, where $\boldsymbol{w}_i$ represents the individual data in the cloud database and $\boldsymbol{u}$ is the query data. In the scheme, the servers calculate two similarity parameters (i.e., the Euclidean distance and the Hamming distance) between $\boldsymbol{w}_i$ $(1 \leq i \leq n)$ and $\boldsymbol{u}$ in the encrypted form due to additively homomorphic property. The calculated distance values are securely compared with a pre-defined threshold to filter the matching biometric samples. However, the proposed scheme is not practical due to the expensive communication cost between the two servers. Šeděnka *et al.* [271] introduced another outsourcing scheme for biometric authentication with scaled Manhattan and scaled Euclidean verifiers. The work first presents an algorithm based on GCs method, then modifies it to an AHE-based scheme for a higher security guarantee.

To improve authentication accuracy, the authors used the idea of principal component analysis (PCA), yet increasing the overhead of computation and communication.

For better efficiency performance, Hu *et al.* [272] described two different solutions for outsourcing biometric identification tasks respectively for single-server and two-server (assuming the two servers are non-colluding) models. The single-server protocol disguises the data using a symmetric-key encryption scheme and mathematical transformations. At the end of the protocol, the server ranks the Euclidean distances between the input record and database records, and returns the closest record to the client. While the two-server protocol achieves a higher security standard using a public-key SWHE scheme integrated with SIMD model. After homomorphically calculating the distances, the servers shuffle the indexes and return the permuted index with a minimum distance of the input record to the client. Thus, the real index of the result and its related distance are unknown to the servers. For the semi-honest model, the former protocol is secure under known-sample attack (KSA), and the latter one achieves the security under known-plaintext attack (KPA). Achieving both the requirements of data security and verifiability, Salem *et al.* [273] proposed a privacy-preserving biometric recognition system. Based on the property of additive homomorphism, the recognition process is operated on the encrypted features. Moreover, an additional task of real/fake biometric data detection is carried out by the client, which enhances the strength of integrity and correctness of the system result.

Face recognition is one of the widely used realistic methods for biometric authentication, also as a specific application in the field of image processing. Since the face images are highly private, the privacy issues involved during face recognition cannot be ignored. Researchers have designed privacy-preserving outsourcing schemes for face recognition, which can be applied in practice. Based on a strong cryptographic technology combining MPC and AHE, Erkin *et al.* [274] constructed a privacy-preserving protocol for the face recognition system. The work uses the idea of a standard Eigenface recognition algorithm from [282], operating on the encrypted face image data. In the scheme, the images are projected to a low-dimension face space spanned by the Eigenfaces. The Euclidean distances between features of the input image and the stored image samples are computed homomorphically to indicate the image similarities. After that, a series of comparison procedures are securely performed on these distances to find the most matching result. However, the protocol requires $O(\log N)$ rounds (supposing there exist N samples in the database) and expensive homomorphic operations. For this reason, an improved scheme with better communication and computation efficiency was presented by Sadeghi *et al.* [36], utilizing a hybrid approach which combines the techniques of AHE and GCs. The difference lies in that the client and the server jointly compare the distances by employing a more efficient protocol based on GCs method. Moreover, the scheme offloads most of the computation and communication workloads to a pre-computation phase. Yang *et al.* [275] also proposed an Eigenface-based face recognition scheme, based on an improved SWHE scheme. In the scheme, the clients obtain accurate online face recognition service in a privacy-preserving way.

Another secure homomorphic system for face identification, called SCiFI, was introduced by Osadchy *et al.* [276]. By using an elaborately-designed representation for images, the system is robust to common environmental changes such as illumination changes, occlusions, and shadows. Furthermore, the algorithm computes the Hamming distances between the image samples, which is superior to computing the Euclidean distances used in the Eigenfaces algorithm. The system of [277] achieved practical face matching with several optimization strategies to increase efficiency. On the one hand, the work adopts a more efficient FHE scheme (i.e., FV cryptosystem [283]) to lighten the computation burdens. On the other hand, a batch technique is used to assemble multiplication operations over multiple numbers into one single multiplication. Besides, the authors also applied the method of PCA (principal component analysis) for dimensionality reduction of the face templates. The experimental proof showed that the proposed scheme achieves a reasonable balance between matching accuracy and computational complexity.

Fingerprint and iris recognition are also the biometric authentication technologies frequently used in the systems of attendance, access control, criminal identification, etc. Meanwhile, several secure outsourcing schemes have been proposed for fingerprint and iris recognition tasks, based on HE algorithms. By exploiting PHE schemes and the fingerprint templates called Fingercode [284], Barni *et al.* [278] proposed a privacy-compliant system for outsourcing fingerprint recognition. Since the matching operations are performed homomorphically, the server will never learn the fingerprint template of the client, nor the recognition result. However, the fingerprint templates stored on the server are not encrypted, thus becoming a potential threat to privacy. Apart from this, another drawback of the scheme lies in the low-accuracy of the recognition result. Higo *et al.* [279] introduced an enhanced scheme for homomorphic fingerprint authentication computations, which takes advantage of the information of fingerprint minutiae (i.e., feature points in the fingerprints). Suppose two fingerprint minutiae are denoted as $((x_1, y_1), t_1)$ and $((x_2, y_2), t_2)$, where the former pair in the tuple is the location of one minutia and the latter is its direction. The two minutiae are judged to match if their locations and orientations are close enough. Scheme [280] focused on secure outsourcing solutions for both the iris and fingerprint recognition. The matching processes are homomorphically conducted with the distance computations between the samples, for iris recognition adopting the Hamming distance and for fingerprint recognition adopting the Euclidean distance. Particularly, the comparison operations are performed using GCs evaluation for better efficiency.

TABLE 6. An overview of the surveyed graph computation literature.

Task		Scheme	Secure techniques	Threat model	Interaction
K-nearest neighbor		[285]	PHE	semi-honest	No
		[286]	PHE	semi-honest	No
		[287]	SWHE+PBM	semi-honest	Comp
Shortest path query	Approximate shortest distance	[74]	SE+SWHE+PRFs	semi-honest	No
	Exact shortest distance	[288]	PHE+GCs+PRFs+OT	semi-honest	No
	Obstructed shortest distance	[289]	PHE+MPC+OT	semi-honest	Comp
	Constrained shortest distance	[290]	ORE+SWHE+PRFs	semi-honest	No
Group location services		[291]	SWHE+PBM	semi-honest	No

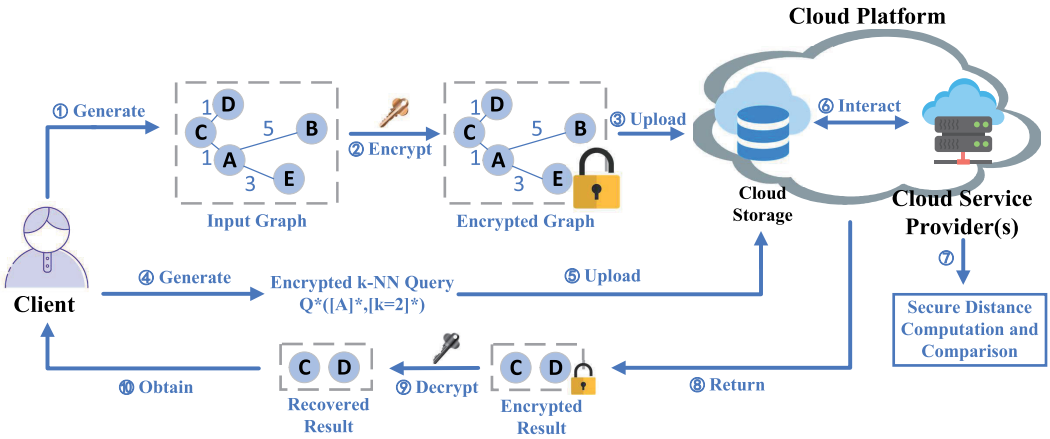


FIGURE 9. A simple example of outsourced k-NN computation system model. The inputs of the system are a specific graph and a specific k-NN query, which are both in the encrypted form. The outputs of the system are the encrypted location points satisfying the defined conditions. The client recovers the system results by a decryption.

D. GRAPH COMPUTATION

In general, a graph structure expresses mutual relationships among a set of objects. Mathematically, the objects are denoted as vertices and the related pairs of vertices are connected by edges. Thus, a graph G can be represented as $G = < V, E >$ where V denotes the vertice set and E denotes the set of the edges between the connected vertices of the graph. The value of a single edge is usually measured by the cost or the strength of the connection. Graphs are used in a wide range of application domains, such as social networks, online knowledge discovery, computer networks, and location-based services. Among them, we focus on the graphic applications of the location-based service (LBS), which is booming up with the rapid growth of the global positioning system (GPS) and mobile devices. We discuss several existing secure outsourcing schemes for LBS computations, considering the factors of security and efficiency performance. In Table 6, we summarize the schemes analyzed in this subsection.

K-nearest neighbor (K-NN) querying has been frequently applied for many practical graphic applications. The systems provide a service where the user obtains top-k nearest points of interest (POIs) by comparing the distances between his current location and the POIs nearby (we illustrate an outsourced k-NN computation model in Fig. 9). A privacy-preserving solution of location-based k-NN query was introduced by Lien *et al.* [285]. In the framework, the POIs of the graph are connected into a circular structure based on the Moore curves, and operated for specific k-NN computations securely due to the homomorphism of Paillier cryptosystem. The proposed scheme is capable of defending against the correlation attack and background knowledge attack. Another practical k-NN query scheme [286] for LBS took into account the type of POIs. In the system, the user specifies the query which asks for k nearest POIs exactly in need without revealing the query type to the server. Considering a $(n \times n)$-cell region with m types of points, the response of k-NN query with one type takes $O(m + n)$ communication complexity. The computation complexities of the user and the cloud LBS provider are $O(m + n)$ and $O(n^2 m)$ respectively. For a higher security level, Kesarwani *et al.* [287] adopted a semi-honest twin-cloud architecture for secure k-NN query computations. In the protocol, due to SWHE homomorphic properties [135], the squared Euclidean distances are computed between encrypted points by one of the cloud servers, and the permuted encrypted distance results are sent to the other server. The second cloud, which holds the secret key decrypts the distances, is responsible for sorting the distances. The confidentiality of specific distance values and query results, as well as access and search patterns, can be protected against the clouds.

Shortest path querying is another popular graph application for LBS services. The query $Q(x, y)$ is intended to calculate the length of the shortest path between the nodes x and y. During this process, the source/destination location

and the distance values should be kept secret. Meng *et al.* [74] described three graph-structured encryption schemes which support approximate shortest distance queries. The first construction, which is computationally efficient, makes use of a symmetric-key encryption scheme. To reduce the communication overhead, the second construction which employs an SWHE scheme is further presented, while increasing the computation burden. The third algorithm, modified from the second one, achieves an optimal balance between the complexities of computation and communication. However, the proposed schemes only provide an estimate of the shortest distance and deal with graphs in a static pattern. To remove these limitations, Wang *et al.* [288] designed a new graph encryption scheme, which supports the exact shortest distance computation and graph dynamic update. The construction employs a hybrid approach combining an AHE scheme and Yao's GCs method for satisfying the security and efficiency requirements. In addition to this, fibonacci heap [292], an advanced data structure for priority queues, is used to assist the shortest distance computation by Dijkstra's algorithm [293]. Another auxiliary data structure, query history (used to store the previous queried results), is also applied to accelerate the computations. The graph updates, like the addition or removal of the edges, can be supported in the design.

To compute the shortest path with obstructions (such as by a traffic jam) in a graph, Zhang *et al.* [289] proposed a secure outsourcing solution using combined cryptographic methods, including MPC, Paillier cryptosystem, and oblivious transfer. In the proposed scheme, the LBS server and cloud server iteratively calculate the shortest path by communicating with each other, based on Floyd-Warshall algorithm. When some obstructions occur in the chosen sub-path, the LBS server uploads related regional information to the cloud server and then collaborately recomputes the shortest path. The shortest path is piecewise determined until the destination is reached. The protocols protect the privacy of both the users and the LBS server. However, the computation overhead of the system [289] should be further reduced.

Constrained shortest distance (CSD) querying is another variation of the shortest distance query. The query pattern aims to find the shortest path in a graph between two given vertices with a constraint that the total cost is no more than a predefined threshold. Since the CSD request over plain graph was proved to be an NP-hard problem, the approximate CSD problem has been guiding the research direction. Shen *et al.* [290] proposed a graph encryption scheme that enables approximate CSD querying based on a tree-based ciphertext comparison protocol. In the system, a user first encrypts the graph and outsources the ciphertexts to the cloud server. Specially, the user constructs a secure searchable index for the graph and encrypts the vertices by particular PRFs, and also encrypts the costs and distances of the graph by order-revealing encryption (ORE) [294] and SWHE, respectively. When receiving CSD query from the user, the cloud matches the entries in the secure index homomorphically with the query token and finally returns the querying result to the user. The construction shows good performance and reaches the security under the chosen-query attack security model [295].

Considering group location services, a secure solution based on distributed architecture was presented by Wang *et al.* [291]. It is applicable for the scenarios of user groups collaboratively completing certain LBS tasks in a privacy-preserving way. Based on BGN cryptosystem, three typical group's services are achieved in the scheme: group nearest neighbor query, optimal group collection point determination, and group friend's distance query, without revealing any group user's location information. Besides, the scheme can well resist distance interaction attacks and collusion attacks.

E. SQL QUERYING

In this subsection, we continue to discuss a practical application for completing data commands on the encrypted database storages, which is structured query language (SQL) querying specifically. As the development of cloud computing and Database-as-a-service (DBaaS) model, increasing individuals and enterprises tend to store their sensitive data in the cloud database and outsource the database operations to the server side. SQL, a kind of programming languages, is often used to access, query, update, and manage the data in the relational database system. Generally, when a client wants to do some operations on the database, he will send a SQL request to the cloud server. Receiving the request, the server accordingly executes on the database and then returns the query result to the client. Since both the database and the querying results should be confidential, how to efficiently and securely outsource the SQL queries becomes a practical issue.

Popa *et al.* [296] proposed a system, called CryptDB, for executing SQL queries over encrypted data on the remote server. The authors employed encryption strategies (like order-preserving encryption (OPE) [297] and HE schemes) to achieve different encryption levels based on the SQL querying types. The system requires no query processing at the client-side. However, in the framework, the same data needs to be re-encrypted to fit for different types of operations. Based on [296], Tu *et al.* [298] designed the first system executing analytical queries on the encrypted data in large databases. The work allocates as much as possible work at the server side and leaves the remaining work to the client. In addition, several optimization techniques are adopted to improve overall performance.

Since both the schemes of [296] and [298] employed different encryption methods aiming at diverse operations, queries with data interoperability (i.e., supporting piping the output of one operation to another as the input) cannot be directly supported. For this reason, a practical system [299] achieved different query operations based on the same encryption scheme, which allows a broader range of queries on the encrypted domains. Unlike the previous systems, the system encrypts the sensitive data using a secret sharing scheme

TABLE 7. An overview of several surveyed schemes of Section VI-E and VI-F.

Task	Scheme	Secure techniques	Verifiability
SQL querying	[296]	PHE+OPE+PRPs	No
	[298]	PHE+OPE	No
	[301]	FHE+OPE+PBM	No
	[302]	PHE	No
Keyword search	[308]	FHE	No
	[311]	FHE+PRPs	No
	[312]	PHE	Yes
	[314]	PHE	Yes

(based on the ideas of [300]) and the row-identifying numbers using AHE. Liu *et al.* [301] also designed a secure outsourcing system for SQL queries with data interoperable, incurring a lower complexity on computation and communication than [299]. In particular, the work employs a homomorphic OPE scheme, which allows the operations of addition, multiplication, order comparison, and equality check directly on the ciphertexts.

Supporting outsourcing numerical range queries (''>'', ''<'', ''='', ''*BETWEEN*'', etc.) to the cloud, some outsourcing schemes like [302] and [303] were proposed. In [302], the system uses a non-colluding twin-cloud architecture, in which the information of the stored database and the query is split into two parts and distributed to the respective cloud server. The numeric-related operations are executed on the encrypted domain jointly by the servers, due to the homomorphic properties of Paillier cryptosystem. When implementing on the real-world datasets, the scheme shows good performance with the advantage of parallel computations. Based on the distributed homomorphic cryptosystem of [158], Cheng *et al.* [303] also employed a two-cloud architecture to support range queries under multiple encrypted domains. In order to improve efficiency, the work adopts a well-designed data packing technique to accelerate the linear processing.

F. KEYWORD SEARCH

Keyword search is also a practical application on encrypted databases frequently used in our daily life. It is a potent tool to find one's desired data from a vast data memory space by specifying the searching keywords. When the searching is implemented on the massive encrypted datasets, constructing efficient keyword searching systems with privacy-preserving has aroused great interest from the researchers. We give an overview of the existing HE-based outsourcing techniques for the function, and summarize several surveyed schemes of SQL querying (mentioned in VI-E) and keyword search in Table 7.

To obtain the desired data, clients often uploads one or more encrypted keywords to the cloud server side. Accordingly, the server searches on the ciphertexts stored and sends back the relevant data to the clients. One solution which supports outsourced keyword searching was introduced by Hou *et al.* [304]. To preserve data privacy, two searching schemes are constructed based on HE and commutative encryption schemes, respectively. However, the system only looks for the data matching a certain keyword instead of simultaneous multiple keywords. An enhanced version of [304] was proposed in [305], which enables the server to match multiple keywords. The work designs both disjunctive and conjunctive multi-keyword search algorithms by the use of HE technique. Scheme [306] achieved conjunctive-keyword searches in a privacy-preserving way, supporting valid search authorization for a limited time period. The system can be resistant to chosen-keyword chosen-time attack and off-line keyword guessing attack. Since most schemes only support exact or fuzzy search, Yang *et al.* [307] described a more practical secure search solution from the keyword semantic perspective. Depending on the semantic information, the system returns relevant results with semantically related keywords to the users.

Yu *et al.* [308] proposed a two-round top-k multi-keyword retrieval scheme, which adopts a vector space model (VSM) to represent the file, and a modified FHE scheme [309] to encrypt the index/trapdoor. When receiving the multi-keyword query, the server calculates the file relevance scores (depending on the rules of term frequency-inverse document frequency (TF-IDF) [310]) and returns the encrypted scores to the client. Then, the client decrypts the scores and executes a top-k sorting algorithm locally. Finally, the client sends the k highest-scoring files' identifiers to the server and accesses their corresponding files. Nevertheless, the system is inefficient for practical applications on mass encrypted data on account of the efficiency limitations of FHE. Similarly, Strizhov and Ray [311] also achieved a multi-keyword search system with the returning results sorting by the scores. The proposed scheme reaches an optimal sublinear search time and is secure against adaptive chosen-keyword attacks (CKAs). Zhang *et al.* [312] designed a secure ranked keyword search scheme with verifiability. Once the server misbehaves, it will be detected with a high possibility. With the Paillier cryptosystem with threshold decryption (PCTD) in [157], Yang *et al.* [11], [313], [314] also proposed secure top-k rank systems for multi-keyword search. In [313], wildcards are allowed in the queried keywords. Besides, the keywords can be joined by the logical operator *AND* or *OR*. By using a standard encoding technique (i.e., Unicode [315]), the system of [11] is capable of searching on the encrypted data in arbitrary languages. Moreover, the clients can set preference scores for queried keywords in order to get more satisfying results. For more expressive queries, [314] supports different querying patterns, such as single/conjunctive keyword query and mixed boolean query. In the schemes of [11], [313], [314], searching on the data from multiple data owners only requires one trapdoor. In addition, flexible search authorization and revocation are also realized in the works.

In recent years, plenty of the proposed keyword search schemes employ VSM method to support file relevance computations, while a considerable computation complexity will be introduced as the data dimension increasing. For this reason, Yao *et al.* [316] proposed a secure index scheme for ranked multiple keywords search based on counting bloom

TABLE 8. Efficiency and security comparison of schemes for outsourcing fundamental functions.

Task	Scheme	Threat model	Secure techniques	Verifiability	Computation complexity	Communication complexity	Interaction
Set intersection	[48]	semi-honest	PHE+PBM	No	$O(k \ln \ln k)$	$O(k)$	No
		malicious	PHE+PBM	Yes	$O(k \ln \ln k)$	$O(k)$	Verf
	[165]	malicious	PHE+SS	Yes	$O(k^2 \log k + k \log^2 k)$	$O(k \log^2 k)$	Comp, Verf
	[167]	semi-honest	PHE+PRFs	No	$O(k)$	$O(k)$	No
	[169]	semi-honest	PHE+SS+OT	No	$O(\lambda + k)$	$O(\lambda)$	Comp
Set union	[170]	semi-honest	PHE	No	$O(k \log \log k)$	$O(k)$	No
	[171]	semi-honest	PHE+PBM	No	$O(k)$	$O(k)$	No
Set intersection or union cardinality	[171]	semi-honest	PHE+PBM	No	$O(k)$	$O(k)$	No
	[172]	semi-honest	FHE	No	$O(k)$	$O(k)$	No
Matrix multiplication	[173]	semi-honest	PHE	Yes	$O(m^3)$	$O(m^2)$	Comp
	[174]	malicious	PHE+SS+PBM	Yes	$O(m^3)$	$O(m^2)$	No
	[175]	malicious	PHE/SWHE	Yes	$O(m^3)$	$O(m^2)$	No
	[179]	semi-honest	SWHE+PBM	No	$O(m_1 m_2 m_3 / l)$	$O(m_1 m_3 / r)$	No
Matrix eigenvalues and eigenvectors	[182]	malicious	PHE+PBM	Yes	$O(pm^2)$	$O(m^2)$	Comp
	[183]	semi-honest	PHE+PBM	No	$O(m^2/q)$	$O(m)$	Comp

filter (CBF) with a lower computation overhead. Moreover, the major ranking computations are executed by the server, thus further reducing the workloads on the client side. Pervez *et al.* [317] proposed an oblivious similarity based search (OS2) scheme for the encrypted data, achieving secure ranked keyword search by utilizing encrypted bloom filter and HE algorithms.

VII. SECURITY AND PERFORMANCE COMPARISON

We compare the security level and efficiency performance of several existing outsourcing schemes in Table 8, which are analyzed below.

A. THREAT MODEL

For secure outsourced schemes, the threat models are generally classified as ''honest-but-curious'' (or say ''semi-honest'') model and ''malicious'' model. The security level of the ''semi-honest'' model is inferior to the ''malicious'' model, where the server in the latter model may return an incorrect result to the client. Most of the existing works are designed based on the semi-honest model, which are unable to defend malicious server adversaries. In Table 8, the schemes [48], [165], [174], [175], and [182] are secure in the malicious model, while the other schemes can only be secure in the semi-honest model. Take the two schemes in [48] as examples, which are respectively secure in the semi-honest model and malicious model. In the semi-honest setting, the client represents the private set as a polynomial and sends the polynomial's coefficients (encrypted by an HE scheme) to the cloud server. The server (which is assumed to hold the cryptosystem's public key) cannot recover the plaintexts of the polynomial's coefficients. Then, the server performs necessary computations directly on the ciphertexts using the homomorphic properties and returns the computation result to the client. In the scheme constructed for the malicious model, additional parameters are set in the protocol to participate in the verifying procedures. The client accepts the returned result only if it passes the verification. As the elements of the private sets are obscured in both the protocols, data privacy of the client and the server is preserved.

B. VERIFIABILITY

As mentioned before, the verifiability of computational results is an important security guarantee for many outsourced schemes (e.g., [173], [312], and [318]), which checks the correctness of remote computations. In scheme [173], a verification mechanism for matrix multiplication is designed by comparing the values between $\mathbf{A}(\mathbf{B}\boldsymbol{v})$ and $\mathbf{C}\boldsymbol{v}$ at the client (the column vector $\boldsymbol{v}$ is randomly generated by the client). The input matrices are $\mathbf{A}$ and $\mathbf{B}$, and the matrix $\mathbf{C}$ is the result of matrix multiplication returned by the server. If the equation $\mathbf{A}(\mathbf{B}\boldsymbol{v}) = \mathbf{C}\boldsymbol{v}$ holds, the computational result can be proved to be correct. The verification involves products of a $(n \times n)$-dimension matrix and a $(n \times 1)$-dimension column vector, leading to $O(n^2)$ workload at the client. In most of outsourcing schemes [165], [173]–[175], [182], [191], [225], [314], the verification algorithms are elaborately designed to lower computation and communication complexities.

C. COMPUTATION AND COMMUNICATION COMPLEXITY

Another aspect for evaluating outsourcing computation schemes is efficiency. A well-constructed scheme is not supposed to disturb users nor add excessive burden to servers. To measure the efficiency of related works [167], [170], [171], [179], [182], [183], we consider the overhead of the involved operations (including the time and resources consumed for data encryption/decryption, data processing, data transmission and result verification), which are generalized as computation and communication complexities. Take the scheme [170] and [171] as examples to analyze the complexities of their private set union protocols. In the protocols, the set size of both client and server is assumed to be n. In the set-union protocol of [171], the client performs n encryptions and $2n$ decryptions, and the server performs $O(n \log \log n)$ homomorphic operations. The amounts of transmitted ciphertexts are n and $2n$ for the client and the server, respectively. Therefore, the protocol requires $O(n \log \log n)$ computation and $O(n)$ communication. The protocol of [171] adopts bloom filters (instead of the polynomials in [170]) to represent private sets. On the one hand, the client computes B encryptions (B is the number of bits in a bloom filter, which is reckoned

as n in complexity analysis), $2n$ decryptions, and n element inverses, with the total computation complexity as $O(n)$. And the server performs $O(n(k+1))$ homomorphic operations, where k (deemed as a constant) is the number of hash functions. On the other hand, in the protocol, the client sends B ciphertexts to the server, and the server sends $2n$ ciphertexts back to the client. Hence, the protocol's complexities of both computation and communication are linear with respect to the set size of n. Apparently, the set-union protocol of [171] is more efficient than the one in [170].

D. INTERACTION

In addition to the computation and communication complexities, interaction also influences the efficiency performance of related schemes, in terms of transmission costs, numbers of interactions, etc. The interactions generally occur between the client and the server (or between the servers). Many HE-based outsourcing schemes [152], [167], [171], [208], [245], [263], [291] involve one single interaction between the client and the server, which means the client only needs to send his encrypted computational request to the server and waits for the server to return the computational result. The computation processes are entirely performed by the server, thereby minimizing the client's burden. However, some works [165], [169], [173], [182], [183], [219] require additional interactions between the parties of the client and server to complete outsourced tasks or execute result verification. In [182], the algorithm iterations are carried out by the interactions between the client and the server. During each interaction, the server returns the current computational result (in an encrypted form) to the client. Receiving the result, the client decrypts the ciphertexts using his secret key and computes on the plaintexts to update the result. After that, the updated result (also in the encrypted form) is sent to the server for the next iterative round. The interactions continue until the result is converged. To verify the correctness of the result from outsourced computation, the client and the server of [165] sacrifice the efficiency to implement the necessary interactions. In the scenarios of multiple cloud servers (typically two) collaboratively completing the outsourced tasks, the servers' interactions (e.g., exchanging necessary encrypted data) are also essential, like the schemes in [201] and [239].

Compared with the other techniques (mentioned in Section III), homomorphic encryption is always used for privacy-preserving computations with a higher security requirement. For a fixed secret key, if a plaintext is always encrypted into the same ciphertext, we say the corresponding encryption scheme is deterministic. If an encryption scheme produces different ciphertexts for the same plaintext and secret key, the scheme is called probabilistic cryptosystem. In general, the probabilistic HE schemes enjoy a stronger security level than deterministic ones. For example, RSA [125] is not secure against chosen plaintext attacks (CPA) as its encryption algorithm is deterministic. In terms of different HE-based algorithms, various security levels are realized. For instance, the security of Goldwasser-Micali (GM) cryptosystem [118] relies on the quadratic residuosity problem, El-Gamal encryption algorithm [114] based on the hardness of the Diffie-Hellman problem (DHP), and Brakerski and Vaikuntanathan (BV) scheme [177] based on the hardness of ring learning with error problem (ring-LWE).

For most HE-based outsourcing schemes, relatively lower efficiency is the main shortcoming. Nevertheless, there are numerous PHE schemes which can be well used in practical applications due to its acceptable overhead (e.g., some performing the encryption/decryption operations only in milliseconds [17]). Besides, some related works based on FHE are also efficient in use. While for most FHE-based outsourcing schemes, the overhead for storage and computation is still costly. For example, large-size keys and ciphertexts are generated during the process of FHE algorithms, thus leading to longer response time to user's requests [18]. Although some HE-based schemes (especially FHE) currently suffer from poor performance, as the evolvement of modern accelerating techniques and the progress of designed algorithms, the efficiency performance will be continuously improved to meet users' needs.

VIII. FUTURE RESEARCH DIRECTIONS

Among the existing secure outsourcing works, there are still some areas for improvement and exploration to fit actual needs. We discuss several open issues and challenges as positive directions for further outsourcing computation designs (especially HE-based ones):

Computation without Central Trusted Authority. For HE-based outsourcing schemes, trusted authority (TA) is necessary for the system to generate and distribute keys for all parties in the system. In the real environment, it may be hard for all the users to trust the central TA. Moreover, if the TA is compromised, all parties' data will fall into danger.

Secure Complex Cure Processing Efficiently. Most of the secure data processing is to handle some linear functions or simple comparison functions. In the real scenario, a lot of applications rely on complex non-linear functions, such as Tanh function used for neural network [319] or Sigmoid function for logistic regression [320]. How to process these complex cures securely without occupying too many computational resources is still an important question.

New Encrypt-Data format Balancing Secure and High-Precision Computation. In order to increase the precision of outsourced computations, some data formats (such as rational number or floating-point number format) have already been used in the schemes. However, the existing schemes either have a relatively low-security level (like the MPC-based scheme of floating-point number format [321]) or occupy too many computation and communication resources (like the HE-based schemes of outsourced rational number [10] and floating-point number format [152]). Thus, achieving a secure and efficient data format has become a significant issue to be resolved.

Solve Secure Computation Overflow. As all the data is encrypted during the processing, it is impossible for any party to judge whether the data is overflow/underflow in the midway. Without any safeguard procedures, the possible overflow/underflow problems will lead to an unreasonable or wrong result, which is a disaster to the system. Since few existing works have taken this factor into consideration, efficient strategies for testing and solving the overflow/underflow problem are to be broadly implemented in the follow-up works.

Support Large Scale of Multiple Users. Although previous techniques [55], [56], and [158] have considered secure computations over two different parties, the system complexity significantly increases when large-scale of users are involved. Therefore, more secure techniques should be designed for answering massive users efficiently.

Efficiency-Enhancement with Software or Hardware Support. Since the data is always stored individually (i.e., each ciphertext only stores one plaintext) and the computations are usually executed individually. Some new secure SIMD techniques can be designed to perform the same secure operation on multiple encrypted data simultaneously. In addition, many existing secure computation implements are focused on programming in the CPU-based testbed, and some secure computations even do not support multiple threading programs. To further save the running time, researchers can also use GPU-based programming and environment to accelerate secure computation.

IX. CONCLUSION

In this survey, we provide a comprehensive overview of existing works for secure outsourcing computations based on homomorphic encryption technique for the backgrounds of fundamental functions (e.g., scalar operations, set operation, and matrix operations) and application-specific tasks (e.g., machine learning, image processing, and biometric computation). Two main factors are considered in the schemes, which are the security issues (including data security, privacy, and correctness) and the efficiency performance. In addition to these, we give a brief introduction to the secure outsourcing computation and four other standard secure techniques (i.e., secure multi-party computation, pseudorandom functions, software guard extensions, and perturbation approaches). Meanwhile, we explain the theories and evolutions of homomorphic encryption technique. To further provide overall understandings of existing HE-based outsourcing solutions, we also analyze and compare the security levels and efficiency performance of these works. Finally, we discuss several open research directions for further research.

REFERENCES

[1] L. Wang, G. V. Laszewski, A. Younge, X. He, M. Kunze, J. Tao, and C. Fu, "Cloud computing: A perspective study," *New Generat. Comput.*, vol. 28, no. 2, pp. 137–146, 2010.

[2] I. Arpaci, "Antecedents and consequences of cloud computing adoption in education to achieve knowledge management," *Comput. Hum. Behav.*, vol. 70, pp. 382–390, May 2017.

[3] Y. Yang, X. Zheng, and C. Tang, "Lightweight distributed secure data management system for health Internet of Things," *J. Netw. Comput. Appl.*, vol. 89, pp. 26–37, Jul. 2017.

[4] Y. Yang, X. Liu, R. H. Deng, and Y. Li, "Lightweight sharable and traceable secure mobile health system," *IEEE Trans. Dependable Secure Comput.*, to be published.

[5] W. Guo, B. Lin, G. Chen, Y. Chen, and F. Liang, "Cost-driven scheduling for deadline-based workflow across multiple clouds," *IEEE Trans. Netw. Service Manag.*, vol. 15, no. 4, pp. 1571–1585, Dec. 2018.

[6] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud computing: State-of-the-art and research challenges," *J. Internet Services Appl.*, vol. 1, no. 1, pp. 7–18, May 2010.

[7] X. Liu, R. Lu, J. Ma, L. Chen, and B. Qin, "Privacy-preserving patient-centric clinical decision support system on Naïve Bayesian classification," *IEEE J. Biomed. Health Inform.*, vol. 20, no. 2, pp. 655–668, Mar. 2016.

[8] O. Goldreich, "Secure multi-party computation," *Manuscript Preliminary Version*, vol. 78, 1998.

[9] S. Patel, G. Shah, and A. Patel, "Techniques of data perturbation for privacy preserving data mining," *Int. J. Advent Res. Comput. Electron.*, vol. 1, pp. 5–10, Mar. 2014.

[10] X. Liu, K. K. R. Choo, R. H. Deng, R. Lu, and J. Weng, "Efficient and privacy-preserving outsourced calculation of rational numbers," *IEEE Trans. Dependable Secure Comput.*, vol. 15, no. 1, pp. 27–39, Jan./Feb. 2018.

[11] Y. Yang, X. Liu, and R. Deng, "Multi-user multi-keyword rank search over encrypted data in arbitrary language," *IEEE Trans. Dependable Secure Comput.*, to be published.

[12] Y. Yang, X. Liu, and R. H. Deng, "Lightweight break-glass access control system for healthcare Internet-of-Things," *IEEE Trans. Ind. Informat.*, vol. 14, no. 8, pp. 3610–3617, Aug. 2018.

[13] Y. Yang, X. Liu, X. Zheng, C. Rong, and W. Guo, "Efficient traceable authorization search system for secure cloud storage," *IEEE Trans. Cloud Comput.*, to be published.

[14] J. Tang, Y. Cui, Q. Li, K. Ren, J. Liu, and R. Buyya, "Ensuring security and privacy preservation for cloud data services," *ACM Comput. Surv.*, vol. 49, no. 1, 2016, Art. no. 13.

[15] E. Aguiar, Y. Zhang, and M. Blanton, "An overview of issues and recent developments in cloud computing and storage security," in *High Performance Cloud Auditing and Applications*. New York, NY, USA: Springer, 2014, pp. 3–33.

[16] S. Basu, A. Bardhan, K. Gupta, P. Saha, M. Pal, M. Bose, K. Basu, S. Chaudhury, and P. Sarkar, "Cloud computing security challenges & solutions—A survey," in *Proc. IEEE 8th Annu. Comput. Commun. Workshop Conf. (CCWC)*, Jan. 2018, pp. 347–356.

[17] T. S. Fun and A. Samsudin, "A survey of homomorphic encryption for outsourced big data computation," *KSII Trans. Internet Inf. Syst.*, vol. 10, no. 8, pp. 3826–3851, 2016.

[18] M. Alkharji and H. Liu, "Homomorphic encryption algorithms and schemes for secure computations in the cloud," in *Proc. Int. Conf. Secure Comput. Technol.*, 2016, pp. 1–19.

[19] A. Acar, H. Aksu, A. S. Uluagac, and M. Conti, "A survey on homomorphic encryption schemes: Theory and implementation," *ACM Comput. Surv.*, vol. 51, no. 4, 2018, Art. no. 79.

[20] P. Martins, L. Sousa, and A. Mariano, "A survey on fully homomorphic encryption: An engineering perspective," *ACM Comput. Surv.*, vol. 50, no. 6, 2018, Art. no. 83.

[21] X. Yu, Z. Yan, and A. V. Vasilakos, "A survey of verifiable computation," *Mobile Netw. Appl.*, vol. 22, no. 3, pp. 438–453, 2017.

[22] D. Demirel, L. Schabhüser, and J. Buchmann, *Privately and Publicly Verifiable Computing Techniques: A Survey*. New York, NY, USA: Springer, 2017.

[23] X. Chen, *Introduction to Secure Outsourcing Computation* (Synthesis Lectures on Information Security, Privacy and Trust), vol. 8, no. 2. San Rafael, CA, USA: Morgan & Claypool, 2016, pp. 1–93.

[24] S. Salinas, X. Chen, J. Ji, and P. Li, "A tutorial on secure outsourcing of large-scale computations for big data," *IEEE Access*, vol. 4, pp. 1406–1416, 2016.

[25] L. Zhang, Z. Yan, and R. Kantoa, "A review of homomorphic encryption and its applications," in *Proc. 9th EAI Int. Conf. Mobile Multimedia Commun.*, 2016, pp. 97–106.

[26] Z. Shan, K. Ren, M. Blanton, and C. Wang, "Practical secure computation outsourcing: A survey," *ACM Comput. Surv.*, vol. 51, no. 2, 2018, Art. no. 31.

[27] A. C. Yao, "Protocols for secure computations," in *Proc. FOCS*, vol. 82, Nov. 1982, pp. 160–164.

[28] A. C.-C. Yao, "How to generate and exchange secrets," in *Proc. IEEE 27th Annu. Symp. Found. Comput. Sci. (SFCS)*, Oct. 1986, pp. 162–167.

[29] D. Malkhi, N. Nisan, B. Pinkas, and Y. Sella, "Fairplay-secure two-party computation system," in *Proc. USENIX Secur. Symp.*, San Diego, CA, USA, vol. 4, 2004, p. 9.

[30] G. Asharov, Y. Lindell, T. Schneider, and M. Zohner, "More efficient oblivious transfer and extensions for faster secure computation," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2013, pp. 535–548.

[31] D. Beaver, S. Micali, and P. Rogaway, "The round complexity of secure protocols," in *Proc. STOC*, vol. 90, 1990, pp. 503–513.

[32] A. Ben-Efraim, Y. Lindell, and E. Omri, "Optimizing semi-honest secure multiparty computation for the Internet," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2016, pp. 578–590.

[33] D. Evans, Y. Huang, J. Katz, and L. Malka, "Efficient privacy-preserving biometric identification," in *Proc. 17th Conf. Netw. Distrib. Syst. Secur. Symp. (NDSS)*, vol. 68, 2011, pp. 1–40.

[34] M. Barni, P. Failla, V. Kolesnikov, R. Lazzeretti, A.-R. Sadeghi, and T. Schneider, "Secure evaluation of private linear branching programs with medical applications," in *Proc. Eur. Symp. Res. Comput. Secur.* Berlin, Germany: Springer, 2009, pp. 424–439.

[35] J. Brickell, D. E. Porter, V. Shmatikov, and E. Witchel, "Privacy-preserving remote diagnostics," in *Proc. 14th ACM Conf. Comput. Commun. Secur.*, 2007, pp. 498–507.

[36] A.-R. Sadeghi, T. Schneider, and I. Wehrenberg, "Efficient privacy-preserving face recognition," in *Proc. Int. Conf. Inf. Secur. Cryptol.* Berlin, Germany: Springer, 2009, pp. 229–244.

[37] M. Bellare, V. T. Hoang, and P. Rogaway, "Foundations of garbled circuits," in *Proc. ACM Conf. Comput. Commun. Secur.*, 2012, pp. 784–796.

[38] A. Shamir, "How to share a secret," *Commun. ACM*, vol. 22, no. 11, pp. 612–613, Nov. 1979.

[39] G. R. Blakley, "Safeguarding cryptographic keys," in *Proc. Nat. Comput. Conf.*, vol. 48, 1979, pp. 1–6.

[40] M. Ben-Or, S. Goldwasser, and A. Wigderson, "Completeness theorems for non-cryptographic fault-tolerant distributed computation," in *Proc. 20th Annu. ACM Symp. Theory Comput.*, 1988, pp. 1–10.

[41] D. Chaum, C. Crépeau, and I. Damgard, "Multiparty unconditionally secure protocols," in *Proc. 20th Annu. ACM Symp. Theory Comput.*, 1988, pp. 11–19.

[42] I. Damgård, M. Fitzi, E. Kiltz, J. B. Nielsen, and T. Toft, "Unconditionally secure constant-rounds multi-party computation for equality, comparison, bits and exponentiation," in *Proc. Theory Cryptogr. Conf.* Berlin, Germany: Springer, 2006, pp. 285–304.

[43] T. Nishide and K. Ohta, "Multiparty computation for interval, equality, and comparison without bit-decomposition protocol," in *Proc. Int. Workshop Public Key Cryptogr.* Berlin, Germany: Springer, 2007, pp. 343–360.

[44] M. J. Atallah and W. Du, "Secure multi-party computational geometry," in *Proc. Workshop Algorithms Data Struct.* Berlin, Germany: Springer, 2001, pp. 165–179.

[45] T. Gong, H. Huang, P. Chen, R. Malekian, and T. Chen, "Secure two-party distance computation protocol based on privacy homomorphism and scalar product in wireless sensor networks," *Tsinghua Sci. Technol.*, vol. 21, no. 4, pp. 385–396, 2016.

[46] Y. Yao and F. Yu, "Privacy-preserving similarity sorting in multi-party model," *IJ Netw. Secur.*, vol. 19, no. 5, pp. 851–857, 2017.

[47] E. Ioannidis, E. Weinsberg, N. A. Taft, M. Joye, and V. Nikolaenko, "A method and system for privacy preserving matrix factorization," U.S. Patent 14 771 534, Jan. 5, 2016.

[48] M. J. Freedman, K. Nissim, and B. Pinkas, "Efficient private matching and set intersection," in *Proc. Int. Conf. Theory Appl. Cryptograph. Techn.* Berlin, Germany: Springer, 2004, pp. 1–19.

[49] R. X. Lu, H. Zhu, J. K. Liu, J. Shao, and X. Liu, "Toward efficient and privacy-preserving computing in big data era," *IEEE Netw.*, vol. 28, no. 4, pp. 46–50, Jul./Aug. 2014.

[50] W. Du and M. J. Atallah, "Privacy-preserving cooperative scientific computations," in *Proc. CSFW*, 2001, pp. 273–282.

[51] S. S. M. Chow, J.-H. Lee, and L. Subramanian, "Two-party computation model for privacy-preserving queries over distributed databases," in *Proc. NDSS*, 2009, pp. 1–16.

[52] R. Wang, X. Wang, Z. Li, H. Tang, M. K. Reiter, and Z. Dong, "Privacy-preserving genomic computation through program specialization," in *Proc. 16th ACM Conf. Comput. Commun. Secur.*, 2009, pp. 338–347.

[53] C. Gentry, "Fully homomorphic encryption using ideal lattices," in *Proc. STOC*, vol. 9. 2009, pp. 169–178.

[54] C. Gentry, A. Sahai, and B. Waters, "Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based," in *Proc. Annu. Cryptol. Conf.* Berlin, Germany: Springer, 2013, pp. 75–92.

[55] A. López-Alt, E. Tromer, and V. Vaikuntanathan, "On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption," in *Proc. 44th Annu. ACM Symp. Theory Comput.*, 2012, pp. 1219–1234.

[56] P. Mukherjee and D. Wichs, "Two round MPC from LWE via multi-key FHE," *IACR Cryptol. ePrint Arch.*, vol. 2015, p. 345, Apr. 2015.

[57] S. Garg, C. Gentry, S. Halevi, M. Raykova, A. Sahai, and B. Waters, "Candidate indistinguishability obfuscation and functional encryption for all circuits," *SIAM J. Comput.*, vol. 45, no. 3, pp. 882–929, 2016.

[58] A. Sahai and B. Waters, "How to use indistinguishability obfuscation: Deniable encryption, and more," in *Proc. 46th Annu. ACM Symp. Theory Comput.*, 2014, pp. 475–484.

[59] S. Garg, C. Gentry, S. Halevi, and M. Raykova, "Two-round secure MPC from indistinguishability obfuscation," in *Proc. Theory Cryptogr. Conf.* Berlin, Germany: Springer, 2014, pp. 74–94.

[60] S. Garg and A. Polychroniadou, "Two-round adaptively secure MPC from indistinguishability obfuscation," in *Proc. Theory Cryptogr. Conf.* Berlin, Germany: Springer, 2015, pp. 614–637.

[61] O. Goldreich, S. Goldwasser, and S. Micali, "How to construct random functions," *J. ACM*, vol. 33, no. 4, pp. 792–807, 1986.

[62] M. J. Freedman, Y. Ishai, B. Pinkas, and O. Reingold, "Keyword search and oblivious pseudorandom functions," in *Proc. Theory Cryptogr. Conf.* Berlin, Germany: Springer, 2005, pp. 303–324.

[63] S. Benabbas, R. Gennaro, and Y. Vahlis, "Verifiable delegation of computation over large datasets," in *Proc. Annu. Cryptol. Conf.* Berlin, Germany: Springer, 2011, pp. 111–131.

[64] D. Fiore and R. Gennaro, "Publicly verifiable delegation of large polynomials and matrix computations, with applications," in *Proc. ACM Conf. Comput. Commun. Secur.*, 2012, pp. 501–512.

[65] C. Wang, B. Zhang, K. Ren, J. M. Roveda, C. W. Chen, and Z. Xu, "A privacy-aware cloud-assisted healthcare monitoring system via compressive sensing," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, Apr./May 2014, pp. 2130–2138.

[66] S. Kamara, P. Mohassel, M. Raykova, and S. Sadeghian, "Scaling private set intersection to billion-element sets," in *Proc. Int. Conf. Financial Cryptogr. Data Secur.* Berlin, Germany: Springer, 2014, pp. 195–215.

[67] H. Li, S. Zhang, T. H. Luan, H. Ren, Y. Dai, and L. Zhou, "Enabling efficient publicly verifiable outsourcing computation for matrix multiplication," in *Proc. IEEE Int. Telecommun. Netw. Appl. Conf. (ITNAC)*, Nov. 2015, pp. 44–50.

[68] N. H. Tran, H. H. Pang, and R. H. Deng, "Efficient verifiable computation of linear and quadratic functions over encrypted data," in *Proc. 11th ACM Asia Conf. Comput. Commun. Secur.*, 2016, pp. 605–616.

[69] X. Yuan, H. Duan, and C. Wang, "Bringing execution assurances of pattern matching in outsourced middleboxes," in *Proc. IEEE 24th Int. Conf. Netw. Protocols (ICNP)*, Nov. 2016, pp. 1–10.

[70] J. Wang, X. Chen, J. Li, J. Zhao, and J. Shen, "Towards achieving flexible and verifiable search for outsourced database in cloud computing," *Future Gener. Comput. Syst.*, vol. 67, pp. 266–275, Feb. 2017.

[71] C. Hazay and Y. Lindell, "Efficient protocols for set intersection and pattern matching with security against malicious and covert adversaries," in *Proc. Theory Cryptogr. Conf.* Berlin, Germany: Springer, 2008, pp. 155–175.

[72] S. Evdokimov and O. Günther, "Encryption techniques for secure database outsourcing," in *Proc. Eur. Symp. Res. Comput. Secur.* Berlin, Germany: Springer, 2007, pp. 327–342.

[73] X. Ma, J. Li, and F. Zhang, "Outsourcing computation of modular exponentiations in cloud computing," *Cluster Comput.*, vol. 16, no. 4, pp. 787–796, 2013.

[74] X. Meng, S. Kamara, K. Nissim, and G. Kollios, "GRECS: Graph encryption for approximate shortest distance queries," in *Proc. 22nd ACM SIGSAC Conf. Comput. Commun. Secur.*, 2015, pp. 504–517.

[75] I. Anati, S. Gueron, S. P. Johnson, and V. R. Scarlata, "Innovative technology for CPU based attestation and sealing," in *Proc. ACM 2nd Int. Workshop Hardw. Archit. Support Secur. Privacy*, New York, NY, USA, vol. 13, 2013, pp. 1–7.

[76] V. Costan and S. Devadas, "Intel SGX explained," IACR Cryptol. ePrint Arch., Tech. Rep. 2016/086, 2016, pp. 1–118.

[77] F. Chen, C. Wang, W. Dai, X. Jiang, N. Mohammed, M. M. Al Aziz, M. N. Sadat, C. Sahinalp, K. Lauter, and S. Wang, "PRESAGE: Privacy-preserving genetic testing via software guard extension," *BMC Med. Genomics*, vol. 10, no. 2, 2017, Art. no. 48.
[78] O. Ohrimenko, F. Schuster, C. Fournet, A. Mehta, S. Nowozin, K. Vaswani, and M. Costa, "Oblivious multi-party machine learning on trusted processors," in *Proc. 25th USENIX Secur. Symp. (USENIX Secur.)*, 2016, pp. 619–636.
[79] F. Schuster, M. Costa, C. Fournet, C. Gkantsidis, M. Peinado, G. Mainar-Ruiz, and M. Russinovich, "VC3: Trustworthy data analytics in the cloud using SGX," in *Proc. IEEE Symp. Secur. Privacy*, May 2015, pp. 38–54.
[80] J. Dean and S. Ghemawat, "MapReduce: Simplified data processing on large clusters," *Commun. ACM*, vol. 51, no. 1, pp. 107–113, 2008.
[81] M. N. Sadat, M. M. Al Aziz, N. Mohammed, F. Chen, S. Wang, and X. Jiang, "SAFETY: Secure GWAS in federated environment through a hybrid solution with Intel SGX and homomorphic encryption," 2017, *arXiv:1703.02577*. [Online]. Available: https://arxiv.org/abs/1703.02577
[82] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Proc. Int. Conf. Theory Appl. Cryptograph. Techn.* Berlin, Germany: Springer, 1999, pp. 223–238.
[83] Y. Chen, W. Sun, N. Zhang, Q. Zheng, W. Lou, and Y. T. Hou, "A secure remote monitoring framework supporting efficient fine-grained access control and data processing in IoT," in *Proc. Int. Conf. Secur. Privacy Commun. Syst.* Cham, Switzerland: Springer, 2018, pp. 3–21.
[84] M. Reuter, "Privacy preserving deep neural network prediction using trusted hardware," M.S. thesis, Aalto Univ., Finland, Espoo, Finland, 2018. [Online]. Available: https://aaltodoc.aalto.fi/handle/123456789/34699
[85] Y. Jiang, J. Hamer, C. Wang, X. Jiang, M. Kim, Y. Song, Y. Xia, N. Mohammed, M. N. Sadat, and S. Wang, "SecureLR: Secure logistic regression model via a hybrid cryptographic protocol," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 16, no. 1, pp. 113–123, Jan./Feb. 2019.
[86] J. Allen, B. Ding, J. Kulkarni, H. Nori, O. Ohrimenko, and S. Yekhanin, "An algorithmic framework for differentially private data analysis on trusted processors," 2018, *arXiv:1807.00736*. [Online]. Available: https://arxiv.org/abs/1807.00736
[87] V. Kulkarni, B. Chapuis, and B. Garbinato, "Privacy-preserving location-based services by using Intel SGX," in *Proc. ACM 1st Int. Workshop Hum.-Centered Sens., Netw., Syst.*, 2017, pp. 13–18.
[88] W. Wang, G. Chen, X. Pan, Y. Zhang, X. Wang, V. Bindschaedler, H. Tang, and C. A. Gunter, "Leaky cauldron on the dark land: Understanding memory side-channel hazards in SGX," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2017, pp. 2421–2434.
[89] J. Götzfried, M. Eckert, S. Schinzel, and T. Müller, "Cache attacks on Intel SGX," in *Proc. ACM 10th Eur. Workshop Syst. Secur.*, 2017, Art. no. 2.
[90] F. Brasser, U. Müller, A. Dmitrienko, K. Kostiainen, S. Capkun, and A.-R. Sadeghi, "Software grand exposure: SGX cache attacks are practical," in *Proc. 11th USENIX Workshop Offensive Technol. (WOOT)*, 2017, pp. 1–17.
[91] S. Lee, M.-W. Shih, P. Gera, T. Kim, H. Kim, and M. Peinado, "Inferring fine-grained control flow inside SGX enclaves with branch shadowing," in *Proc. 26th USENIX Secur. Symp. (USENIX Secur.)*, 2017, pp. 557–574.
[92] Y. Xu, W. Cui, and M. Peinado, "Controlled-channel attacks: Deterministic side channels for untrusted operating systems," in *Proc. IEEE Symp. Secur. Privacy*, May 2015, pp. 640–656.
[93] S. Matetic, M. Ahmed, K. Kostiainen, A. Dhar, D. Sommer, A. Gervais, A. Juels, and S. Capkun, "ROTE: Rollback protection for trusted execution," in *Proc. 26th USENIX Secur. Symp. (USENIX Secur.)*, 2017, pp. 1289–1306.
[94] D. E. R. Denning, *Cryptography and Data Security*. Reading, MA, USA: Addison-Wesley, 1982.
[95] S. P. Reiss, "Practical data-swapping: The first steps," *ACM Trans. Database Syst.*, vol. 9, no. 1, pp. 20–37, 1984.
[96] R. Agrawal and R. Srikant, "Privacy-preserving data mining," *ACM SIGMOD Rec.*, vol. 29, no. 2, pp. 439–450, 2000.
[97] J. Liu and Y. Xu, "Privacy preserving clustering by random response method of geometric transformation," in *Proc. IEEE 4th Int. Conf. Internet Comput. Sci. Eng.*, Dec. 2009, pp. 181–188.
[98] S. Oliveira and O. Zaiane, "Data perturbation by rotation for privacy preserving clustering," Tech. Rep. TR04-17, 2004, doi: 10.7939/R3344P.
[99] K. Chen and L. Liu, "A random rotation perturbation approach to privacy preserving data classification," Georgia Inst. Technol., Atlanta, GA, USA, Tech. Rep. GIT-CC-05-12, 2005.
[100] E. Lefons, A. Silvestri, and F. Tangorra, "An analytic approach to statistical databases," in *Proc. VLDB*, 1983, pp. 260–274.
[101] C. K. Liew, U. J. Choi, and C. J. Liew, "A data distortion by probability distribution," *ACM Trans. Database Syst.*, vol. 10, no. 3, pp. 395–411, 1985.
[102] J. Duan, J. Zhou, and Y. Li, "Secure and verifiable outsourcing of nonnegative matrix factorization (NMF)," in *Proc. 4th ACM Workshop Inf. Hiding Multimedia Secur.*, 2016, pp. 63–68.
[103] D. E. Knuth, *The Art of Computer Programming: Seminumerical Algorithms*, vol. 2. Boston, MA, USA: Addison-Wesley, 1981.
[104] P. Yang, X. Gui, J. An, J. Yao, J. Lin, and F. Tian, "A retrievable data perturbation method used in privacy-preserving in cloud computing," *China Commun.*, vol. 11, no. 8, pp. 73–84, Aug. 2014.
[105] K.-P. Lin, "Privacy-preserving kernel k-means clustering outsourcing with random transformation," *Knowl. Inf. Syst.*, vol. 49, no. 3, pp. 885–908, 2016.
[106] S. Balasubramaniam and V. Kavitha, "Geometric data perturbation-based personal health record transactions in cloud computing," *Sci. World J.*, vol. 2015, Jan. 2015, Art. no. 927867.
[107] V. S. Reddy and B. T. Rao, "A combined clustering and geometric data perturbation approach for enriching privacy preservation of healthcare data in hybrid clouds," *Int. J. Intell. Eng. Syst.*, vol. 11, no. 1, pp. 201–210, 2018.
[108] F. Chen, T. Xiang, and Y. Yang, "Privacy-preserving and verifiable protocols for scientific computation outsourcing to the cloud," *J. Parallel Distrib. Comput.*, vol. 74, no. 3, pp. 2141–2151, 2014.
[109] Y. Yu, Y. Luo, D. Wang, S. Fu, and M. Xu, "Efficient, secure and non-iterative outsourcing of large-scale systems of linear equations," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2016, pp. 1–6.
[110] Z. Xia, X. Ma, Z. Shen, X. Sun, N. N. Xiong, and B. Jeon, "Secure image LBP feature extraction in cloud-based smart campus," *IEEE Access*, vol. 6, pp. 30392–30401, 2018.
[111] W. Wu, U. Parampalli, J. Liu, and M. Xian, "Privacy preserving k-nearest neighbor classification over encrypted database in outsourced cloud environments," *World Wide Web*, vol. 22, no. 1, pp. 101–123, 2019.
[112] K. Chen and L. Liu, "A survey of multiplicative perturbation for privacy-preserving data mining," in *Privacy-Preserving Data Mining*. Boston, MA, USA: Springer, 2008, pp. 157–181.
[113] J. Daemen and V. Rijmen, *The Design of RijndaeL: AES—The Advanced Encryption Standard*. New York, NY, USA: Springer, 2013.
[114] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," *IEEE Trans. Inf. Theory*, vol. IT-31, no. 4, pp. 469–472, Jul. 1985.
[115] A. C.-F. Chan, "Symmetric-key homomorphic encryption for encrypted data processing," in *Proc. IEEE Int. Conf. Commun.*, Jun. 2009, pp. 1–5.
[116] I. Sharma, "Fully homomorphic encryption scheme with symmetric keys," 2013, *arXiv:1310.2452*. [Online]. Available: https://arxiv.org/abs/1310.2452
[117] J.-S. Coron, D. Naccache, and M. Tibouchi, "Public key compression and modulus switching for fully homomorphic encryption over the integers," in *Proc. Annu. Int. Conf. Theory Appl. Cryptograph. Techn.* Berlin, Germany: Springer, 2012, pp. 446–464.
[118] S. Goldwasser and S. Micali, "Probabilistic encryption & how to play mental poker keeping secret all partial information," in *Proc. 14th Annu. ACM Symp. Theory Comput.*, 1982, pp. 365–377.
[119] J. Benaloh, "Dense probabilistic encryption," in *Proc. Workshop Sel. Areas Cryptogr.*, 1994, pp. 120–128.
[120] Y.-C. Chang and M. Mitzenmacher, "Privacy preserving keyword searches on remote encrypted data," in *Proc. Int. Conf. Appl. Cryptogr. Netw. Secur.* Berlin, Germany: Springer, 2005, pp. 442–455.
[121] R. Lu, X. Liang, X. Li, X. Lin, and X. Shen, "EPPA: An efficient and privacy-preserving aggregation scheme for secure smart grid communications," *IEEE Trans. Parallel Distrib. Syst.*, vol. 23, no. 9, pp. 1621–1631, Sep. 2012.
[122] B. K. Samanthula, Y. Elmehdwi, and W. Jiang, "k-nearest neighbor classification over semantically secure encrypted relational data," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 5, pp. 1261–1273, May 2015.
[123] C. A. Melchor, P. Gaborit, and J. Herranz, "Additively homomorphic encryption with d-operand multiplications," in *Proc. Annu. Cryptol. Conf.* Berlin, Germany: Springer, 2010, pp. 138–154.

[124] I. Damgård and M. Jurik, "A generalisation, a simpli.cation and some applications of Paillier's probabilistic public-key system," in *Proc. Int. Workshop Public Key Cryptogr.* Berlin, Germany: Springer, 2001, pp. 119–136.

[125] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Commun. ACM*, vol. 21, no. 2, pp. 120–126, Feb. 1978.

[126] T. Okamoto and S. Uchiyama, "A new public-key cryptosystem as secure as factoring," in *Proc. Int. Conf. Theory Appl. Cryptograph. Techn.* Berlin, Germany: Springer, 1998, pp. 308–318.

[127] K. Peng, C. Boyd, and E. Dawson, "A multiplicative homomorphic sealed-bid auction based on Goldwasser–Micali encryption," in *Proc. Int. Conf. Inf. Secur.* Berlin, Germany: Springer, 2005, pp. 374–388.

[128] D. Boneh, E.-J. Goh, and K. Nissim, "Evaluating 2-DNF formulas on ciphertexts," in *Proc. Theory Cryptogr. Conf.* Berlin, Germany: Springer, 2005, pp. 325–341.

[129] F. Armknecht and A.-R. Sadeghi, "A new approach for algebraically homomorphic encryption," IACR Cryptol. ePrint Arch., Tech. Rep. 422, 2008.

[130] C. Gentry and D. Boneh, "A fully homomorphic encryption scheme," Ph.D. dissertation, Stanford Univ. Stanford, Stanford, CA, USA, 2009, vol. 20.

[131] C. Gentry and S. Halevi, "Implementing gentry's fully-homomorphic encryption scheme," in *Proc. Annu. Int. Conf. Theory Appl. Cryptograph. Techn.* Berlin, Germany: Springer, 2011, pp. 129–148.

[132] D. Stehlé and R. Steinfeld, "Faster fully homomorphic encryption," in *Proc. Int. Conf. Theory Appl. Cryptol. Inf. Secur.* Berlin, Germany: Springer, 2010, pp. 377–394.

[133] N. P. Smart and F. Vercauteren, "Fully homomorphic encryption with relatively small key and ciphertext sizes," in *Proc. Int. Workshop Public Key Cryptogr.* Berlin, Germany: Springer, 2010, pp. 420–443.

[134] Z. Brakerski and V. Vaikuntanathan, "Efficient fully homomorphic encryption from (standard) LWE," in *Proc. IEEE 52nd Annu. Symp. Found. Comput. Sci.*, Oct. 2011, pp. 97–106.

[135] Z. Brakerski, C. Gentry, and V. Vaikuntanathan, "(Leveled) fully homomorphic encryption without bootstrapping," in *Proc. ACM 3rd Innov. Theor. Comput. Sci. Conf.*, 2012, pp. 309–325.

[136] N. P. Smart and F. Vercauteren, "Fully homomorphic SIMD operations," *Designs, Codes Cryptogr.*, vol. 71, no. 1, pp. 57–81, 2014.

[137] S. Halevi and V. Shoup, "HElib—An implementation of homomorphic encryption," Cryptol. ePrint Arch., Tech. Rep. 2014/039, 2014.

[138] S. Halevi and V. Shoup, "Bootstrapping for HElib," in *Proc. Annu. Int. Conf. Theory Appl. Cryptograph. Techn.* Berlin, Germany: Springer, 2015, pp. 641–670.

[139] C. Gentry, S. Halevi, and N. P. Smart, "Homomorphic evaluation of the AES circuit," in *Proc. Annu. Cryptol. Conf.* Berlin, Germany: Springer, 2012, pp. 850–867.

[140] L. Ducas and D. Micciancio, "FHEW: Bootstrapping homomorphic encryption in less than a second," in *Proc. Annu. Int. Conf. Theory Appl. Cryptograph. Techn.* Berlin, Germany: Springer, 2015, pp. 617–640.

[141] I. Chillotti, N. Gama, M. Georgieva, and M. Izabachène, "Faster fully homomorphic encryption: Bootstrapping in less than 0.1 seconds," in *Proc. Int. Conf. Theory Appl. Cryptol. Inf. Secur.* Berlin, Germany: Springer, 2016, pp. 3–33.

[142] I. Chillotti, N. Gama, M. Georgieva, and M. Izabachène, "Faster packed homomorphic operations and efficient circuit bootstrapping for TFHE," in *Proc. Int. Conf. Theory Appl. Cryptol. Inf. Secur.* Berlin, Germany: Springer, 2017, pp. 377–408.

[143] P. Méaux, A. Journault, F.-X. Standaert, and C. Carlet, "Towards stream ciphers for efficient FHE with low-noise ciphertexts," in *Proc. Annu. Int. Conf. Theory Appl. Cryptograph. Techn.* Berlin, Germany: Springer, 2016, pp. 311–343.

[144] Z. Brakerski, "Quantum FHE (almost) as secure as classical," in *Advances in Cryptology—CRYPTO* (Lecture Notes in Computer Science), vol. 10993. Cham, Switzerland: Springer, 2018, pp. 67–95.

[145] D. Boneh, R. Gennaro, S. Goldfeder, A. Jain, S. Kim, P. M. R. Rasmussen, and A. Sahai, "Threshold cryptosystems from threshold fully homomorphic encryption," in *Proc. Annu. Int. Cryptol. Conf.* Cham, Switzerland: Springer, 2018, pp. 565–596.

[146] M. G. Kaosar, R. Paulet, and X. Yi, "Fully homomorphic encryption based two-party association rule mining," *Data Knowl. Eng.*, vols. 76–78, pp. 1–15, Jun./Aug. 2012.

[147] X. Yi, M. Kaosar, M. Golam, R. Paulet, and E. Bertino, "Single-database private information retrieval from fully homomorphic encryption," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 5, pp. 1125–1134, May 2013.

[148] X. Liu, R. Deng, K.-K. R. Choo, and Y. Yang, "Privacy-preserving outsourced clinical decision support system in the cloud," *IEEE Trans. Services Comput.*, to be published.

[149] C. Gong, M. Li, L. Zhao, Z. Guo, and G. Han, "Homomorphic evaluation of the integer arithmetic operations for mobile edge computing," *Wireless Commun. Mobile Comput.*, vol. 2018, Nov. 2018, Art. no. 8142102.

[150] C. Ugwuoke, Z. Erkin, and R. L. Lagendijk, "Secure fixed-point division for homomorphically encrypted operands," in *Proc. ACM 13th Int. Conf. Availability, Rel. Secur.*, 2018, Art. no. 33.

[151] X. Liu, R. Deng, K.-K. R. Choo, Y. Yang, and H. Pang, "Privacy-preserving outsourced calculation toolkit in the cloud," *IEEE Trans. Dependable Secure Comput.*, to be published.

[152] X. Liu, R. H. Deng, W. Ding, R. Lu, and B. Qin, "Privacy-preserving outsourced calculation on floating point numbers," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 11, pp. 2513–2527, Nov. 2016.

[153] S. Arita and S. Nakasato, "Fully homomorphic encryption for point numbers," in *Proc. Int. Conf. Inf. Secur. Cryptol.* Cham, Switzerland: Springer, 2016, pp. 253–270.

[154] S. Bai, G. Yang, J. Shi, G. Liu, and Z. Min, "Privacy-preserving oriented floating-point number fully homomorphic encryption scheme," *Secur. Commun. Netw.*, vol. 2018, Jul. 2018, Art. no. 2363928.

[155] Z. Min, G. Yang, A. K. Sangaiah, S. Bai, and G. Liu, "A privacy protection-oriented parallel fully homomorphic encryption algorithm in cyber physical systems," *EURASIP J. Wireless Commun. Netw.*, vol. 2019, no. 1, 2019, Art. no. 15.

[156] J. Basilakis and B. Javadi, "Efficient parallel binary operations on homomorphic encrypted real numbers," *IEEE Trans. Emerg. Topics Comput.*, to be published.

[157] X. Liu, R. H. Deng, K.-K. R. Choo, and J. Weng, "An efficient privacy-preserving outsourced calculation toolkit with multiple keys," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 11, pp. 2401–2414, Nov. 2016.

[158] X. Liu, B. Qin, R. H. Deng, R. Lu, and J. Ma, "A privacy-preserving outsourced functional computation framework across large-scale multiple encrypted domains," *IEEE Trans. Comput.*, vol. 65, no. 12, pp. 3567–3579, Dec. 2016.

[159] X. Liu, R. Deng, K.-K. R. Choo, and Y. Yang, "Privacy-preserving reinforcement learning design for patient-centric dynamic treatment regimes," *IEEE Trans. Emerg. Topics Comput.*, to be published.

[160] K. Gai and M. Qiu, "Blend arithmetic operations on tensor-based fully homomorphic encryption over real numbers," *IEEE Trans. Ind. Informat.*, vol. 14, no. 8, pp. 3590–3598, Aug. 2018.

[161] X. Liu, B. Qin, R. H. Deng, and Y. Li, "An efficient privacy-preserving outsourced computation over public data," *IEEE Trans. Services Comput.*, vol. 10, no. 5, pp. 756–770, Sep./Oct. 2017.

[162] H. W. Lim, S. Tople, P. Saxena, and E.-C. Chang, "Faster secure arithmetic computation using switchable homomorphic encryption," IACR Cryptol. ePrint Arch., Tech. Rep. 539, 2014.

[163] X. Yu, Z. Yan, and R. Zhang, "Verifiable outsourced computation over encrypted data," *Inf. Sci.*, vol. 479, pp. 372–385, Apr. 2019.

[164] Y. Azar, A. Z. Broder, A. R. Karlin, and E. Upfal, "Balanced allocations," *SIAM J. Comput.*, vol. 29, no. 1, pp. 180–200, 1999.

[165] D. Dachman-Soled, T. Malkin, M. Raykova, and M. Yung, "Efficient robust private set intersection," in *Proc. Int. Conf. Appl. Cryptogr. Netw. Secur.* Berlin, Germany: Springer, 2009, pp. 125–142.

[166] H. Chen, K. Laine, and P. Rindal, "Fast private set intersection from homomorphic encryption," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2017, pp. 1243–1255.

[167] X. Yang, X. Luo, X. A. Wang, and S. Zhang, "Improved outsourced private set intersection protocol based on polynomial interpolation," *Concurrency Comput., Pract. Exper.*, vol. 30, no. 1, p. e4329, 2018.

[168] O. Ruan, Z. Wang, J. Mi, and M. Zhang, "New approach to set representation and practical private set-intersection protocols," *IEEE Access*, vol. 7, pp. 64897–64906, 2019.

[169] H. Zhu, M. Chen, M. Sun, X. Liao, and L. Hu, "Outsourcing set intersection computation based on Bloom filter for privacy preservation in multimedia processing," *Secur. Commun. Netw.*, vol. 2018, Apr. 2018, Art. no. 5841967.

[170] K. Frikken, "Privacy-preserving set union," in *Proc. Int. Conf. Appl. Cryptogr. Netw. Secur.* Berlin, Germany: Springer, 2007, pp. 237–252.

[171] A. Davidson and C. Cid, "An efficient toolkit for computing private set operations," in *Proc. Australas. Conf. Inf. Secur. Privacy*. Cham, Switzerland: Springer, 2017, pp. 261–278.
[172] A. Tajima, H. Sato, and H. Yamana, "Outsourced private set intersection cardinality with fully homomorphic encryption," in *Proc. IEEE 6th Int. Conf. Multimedia Comput. Syst. (ICMCS)*, May 2018, pp. 1–8.
[173] D. Benjamin and M. J. Atallah, "Private and cheating-free outsourcing of algebraic computations," in *Proc. IEEE 6th Annu. Conf. Privacy, Secur. Trust*, Oct. 2008, pp. 240–245.
[174] M. J. Atallah and K. B. Frikken, "Securely outsourcing linear algebra computations," in *Proc. 5th ACM Symp. Inf., Comput. Commun. Secur.*, 2010, pp. 48–59.
[175] P. Mohassel, "Efficient and secure delegation of linear algebra," IACR Cryptol. ePrint Arch., Tech. Rep. 605, 2011.
[176] D. H. Duong, P. K. Mishra, and M. Yasuda, "Efficient secure matrix multiplication over LWE-based homomorphic encryption," *Tatra Mountains Math. Publications*, vol. 67, no. 1, pp. 69–83, 2016.
[177] Z. Brakerski and V. Vaikuntanathan, "Fully homomorphic encryption from ring-LWE and security for key dependent messages," in *Proc. Annu. Cryptol. Conf.* Berlin, Germany: Springer, 2011, pp. 505–524.
[178] P. K. Mishra, D. Rathee, D. H. Duong, and M. Yasuda, "Fast secure matrix multiplications over ring-based homomorphic encryption," IACR Cryptol. ePrint Arch., Tech. Rep. 663, 2018.
[179] W.-J. Lu and J. Sakuma, "More practical privacy-preserving machine learning as a service via efficient secure matrix multiplication," in *Proc. ACM 6th Workshop Encrypted Comput. Appl. Homomorphic Cryptogr.*, 2018, pp. 25–36.
[180] E. Kiltz, P. Mohassel, E. Weinreb, and M. Franklin, "Secure linear algebra using linearly recurrent sequences," in *Proc. Theory Cryptogr. Conf.* Berlin, Germany: Springer, 2007, pp. 291–310.
[181] Y.-R. Chen, S.-T. Shen, and W.-G. Tzeng, "Weave ElGamal encryption for secure outsourcing algebraic computations over Zp," IACR Cryptol. ePrint Arch., Tech. Rep. 947, 2015.
[182] J. F. Moon, S. Aktar, and M. M. A. Hashem, "Securely outsourcing large scale Eigen value problem to public cloud," in *Proc. IEEE 18th Int. Conf. Comput. Inf. Technol. (ICCIT)*, Dec. 2015, pp. 490–494.
[183] J. Powers and K. Chen, "Secure computation of top-K eigenvectors for shared matrices in the cloud," in *Proc. IEEE 6th Int. Conf. Cloud Comput.*, Jun./Jul. 2013, pp. 155–162.
[184] V. Nikolaenko, S. Ioannidis, U. Weinsberg, M. Joye, N. Taft, and D. Boneh, "Privacy-preserving matrix factorization," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2013, pp. 801–812.
[185] S. Kim, J. Kim, D. Koo, Y. Kim, H. Yoon, and J. Shin, "Efficient privacy-preserving matrix factorization via fully homomorphic encryption: Extended abstract," in *Proc. 11th ACM Asia Conf. Comput. Commun. Secur.*, 2016, pp. 617–628.
[186] E. Kaltofen and B. D. Saunders, "On Wiedemann's method of solving sparse linear systems," in *Proc. Int. Symp. Appl. Algebra, Algebraic Algorithms, Error-Correcting Codes*. Berlin, Germany: Springer, 1991, pp. 29–38.
[187] C. Wang, K. Ren, J. Wang, and K. M. R. Urs, "Harnessing the cloud for securely solving large-scale systems of linear equations," in *Proc. IEEE 31st Int. Conf. Distrib. Comput. Syst.*, Jun. 2011, pp. 549–558.
[188] Y. Aono, T. Hayashi, L. T. Phong, and L. Wang, "Input and output privacy-preserving linear regression," *IEICE Trans. Inf. Syst.*, vol. E100.D, no. 10, pp. 2339–2347, 2017.
[189] T. Morshed, D. Alhadidi, and N. Mohammed, "Parallel linear regression on encrypted data," in *Proc. IEEE 16th Annu. Conf. Privacy, Secur. Trust (PST)*, Aug. 2018, pp. 1–5.
[190] H. Yang, W. He, Q. Zhou, and H. Li, "Efficient and secure outsourced linear regression," in *Proc. Int. Conf. Algorithms Archit. Parallel Process.* Cham, Switzerland: Springer, 2018, pp. 89–102.
[191] V. Nikolaenko, U. Weinsberg, S. Ioannidis, M. Joye, D. Boneh, and N. Taft, "Privacy-preserving ridge regression on hundreds of millions of records," in *Proc. IEEE Symp. Secur. Privacy*, May 2013, pp. 334–348.
[192] I. Giacomelli, S. Jha, M. Joye, C. D. Page, and K. Yoon, "Privacy-preserving ridge regression with only linearly-homomorphic encryption," in *Proc. Int. Conf. Appl. Cryptogr. Netw. Secur.* Cham, Switzerland: Springer, 2018, pp. 243–261.
[193] S. Hu, Q. Wang, J. Wang, S. S. M. Chow, and Q. Zou, "Securing fast learning! Ridge regression over encrypted big data," in *Proc. IEEE Trustcom/BigDataSE/ISPA*, Aug. 2016, pp. 19–26.
[194] Y. Aono, T. Hayashi, L. T. Phong, and L. Wang, "Privacy-preserving logistic regression with distributed data sources via homomorphic encryption," *IEICE Trans. Inf. Syst.*, vol. 99.D, no. 8, pp. 2079–2089, 2016.
[195] C. Bonte and F. Vercauteren, "Privacy-preserving logistic regression training," *BMC Med. Genomics*, vol. 11, no. 4, 2018, Art. no. 86.
[196] A. Kim, Y. Song, M. Kim, K. Lee, and J. H. Cheon, "Logistic regression model training based on the approximate homomorphic encryption," *BMC Med. Genomics*, vol. 11, no. 4, p. 83, 2018.
[197] J. H. Cheon, D. Kim, Y. Kim, and Y. Song, "Ensemble method for privacy-preserving logistic regression based on homomorphic encryption," *IEEE Access*, vol. 6, pp. 46938–46948, 2018.
[198] M. N. Sadat, X. Jiang, M. M. Al Aziz, S. Wang, and N. Mohammed, "Secure and efficient regression analysis using a hybrid cryptographic framework: Development and evaluation," *JMIR Med. Inform.*, vol. 6, no. 1, p. e14, 2018.
[199] S. Laur, H. Lipmaa, and T. Mielikäinen, "Cryptographically private support vector machines," in *Proc. 12th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2006, pp. 618–624.
[200] F.-J. González-Serrano, A. Amor-Martín, and J. Casamayón-Antón, "Supervised machine learning using encrypted training data," *Int. J. Inf. Secur.*, vol. 17, no. 4, pp. 365–377, 2018.
[201] W. Sun, Z. L. Jiang, J. Zhang, S.-M. Yiu, Y. Wu, H. Zhao, X. Wang, and P. Zhang, "Outsourced privacy preserving SVM with multiple keys," in *Proc. Int. Conf. Algorithms Archit. Parallel Process.* Cham, Switzerland: Springer, 2018, pp. 415–430.
[202] Y. Rahulamathavan, R. C.-W. Phan, S. Veluru, K. Cumanan, and M. Rajarajan, "Privacy-preserving multi-class support vector machine for outsourcing the data classification in cloud," *IEEE Trans. Dependable Secure Comput.*, vol. 11, no. 5, pp. 467–479, Sep. 2014.
[203] X. Liu, R. Deng, K.-K. R. Choo, and Y. Yang, "Privacy-preserving outsourced support vector machine design for secure drug discovery," *IEEE Trans. Cloud Comput.*, to be published.
[204] T. Graepel, K. Lauter, and M. Naehrig, "ML confidential: Machine learning on encrypted data," in *Proc. Int. Conf. Inf. Secur. Cryptol.* Berlin, Germany: Springer, 2012, pp. 1–21.
[205] R. Bost, R. A. Popa, S. Tu, and S. Goldwasser, "Machine learning classification over encrypted data," in *Proc. NDSS*, vol. 4324, 2015, p. 4325.
[206] C.-Z. Gao, Q. Cheng, P. He, W. Susilo, and J. Li, "Privacy-preserving Naïve Bayes classifiers secure against the substitution-then-comparison attack," *Inf. Sci.*, vol. 444, pp. 72–88, May 2018.
[207] A. Khedr, G. Gulak, and V. Vaikuntanathan, "SHIELD: Scalable homomorphic implementation of encrypted data-classifiers," *IEEE Trans. Comput.*, vol. 65, no. 9, pp. 2848–2858, Sep. 2016.
[208] S. Bian, M. Hiromoto, and T. Sato, "Towards practical homomorphic email filtering: A hardware-accelerated secure Naïve Bayesian filter," in *Proc. ACM 24th Asia South Pacific Design Autom. Conf.*, 2019, pp. 621–626.
[209] A. Alabdulkarim, M. Al-Rodhaan, T. Ma, and Y. Tian, "PPSDT: A novel privacy-preserving single decision tree algorithm for clinical decision-support systems using IoT devices," *Sensors*, vol. 19, no. 1, p. 142, 2019.
[210] P. Xie, M. Bilenko, T. Finley, R. Gilad-Bachrach, K. Lauter, and M. Naehrig, "Crypto-nets: Neural networks over encrypted data," 2014, *arXiv:1412.6181*. [Online]. Available: https://arxiv.org/abs/1412.6181
[211] R. Gilad-Bachrach, N. Dowlin, K. Laine, K. Lauter, M. Naehrig, and J. Wernsing, "CryptoNets: Applying neural networks to encrypted data with high throughput and accuracy," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 201–210.
[212] X. Ma, X. Chen, and X. Zhang, "Non-interactive privacy-preserving neural network prediction," *Inf. Sci.*, vol. 481, pp. 507–519, May 2019.
[213] E. Hesamifard, H. Takabi, and M. Ghasemi, "CryptoDL: Deep neural networks over encrypted data," 2017, *arXiv:1711.05189*. [Online]. Available: https://arxiv.org/abs/1711.05189
[214] F. Tang, W. Wu, J. Liu, H. Wang, and M. Xian, "Privacy-preserving distributed deep learning via homomorphic re-encryption," *Electronics*, vol. 8, no. 4, p. 411, 2019.
[215] B. K. Samanthula, F.-Y. Rao, E. Bertino, X. Yi, and D. Liu, "Privacy-preserving and outsourced multi-user k-means clustering," 2014, *arXiv:1412.4378*. [Online]. Available: https://arxiv.org/abs/1412.4378
[216] D. Liu, E. Bertino, and X. Yi, "Privacy of outsourced k-means clustering," in *Proc. 9th ACM Symp. Inf., Comput. Commun. Secur.*, 2014, pp. 123–134.

[217] X. Liu, Z. L. Jiang, S.-M. Yiu, X. Wang, C. Tan, Y. Li, Z. Liu, Y. Jin, and J. Fang, "Outsourcing two-party privacy preserving k-means clustering protocol in wireless sensor networks," in *Proc. IEEE 11th Int. Conf. Mobile Ad-Hoc Sensor Netw. (MSN)*, Dec. 2015, pp. 124–133.

[218] A. Theodouli, K. A. Draziotis, and A. Gounaris, "Implementing private k-means clustering using a LWE-based cryptosystem," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Jul. 2017, pp. 88–93.

[219] N. Almutairi, F. Coenen, and K. Dures, "K-means clustering using homomorphic encryption and an updatable distance matrix: Secure third party data clustering with limited data owner interaction," in *Proc. Int. Conf. Big Data Analytics Knowl. Discovery*. Cham, Switzerland: Springer, 2017, pp. 274–285.

[220] G. Sakellariou and A. Gounaris, "Homomorphically encrypted k-means on cloud-hosted servers with low client-side load," *Computing*, vol. 101, no. 12, pp. 1813–1836, 2019.

[221] Q. Zhang, L. T. Yang, A. Castiglione, Z. Chen, and P. Li, "Secure weighted possibilistic c-means algorithm on cloud for clustering big data," *Inf. Sci.*, vol. 479, pp. 515–525, Apr. 2019.

[222] J. H. Cheon, D. Kim, and J. H. Park, "Towards a practical clustering analysis over encrypted data," *Networks*, vol. 6, p. 25, 2019.

[223] F. Liu, W. K. Ng, and W. Zhang, "Encrypted association rule mining for outsourced data mining," in *Proc. IEEE 29th Int. Conf. Adv. Inf. Netw. Appl.*, Mar. 2015, pp. 550–557.

[224] X. Yi, F. Hao, E. Bertino, and A. Bouguettaya, "Privacy-preserving association rule mining in cloud computing," in *Proc. 10th ACM Symp. Inf., Comput. Commun. Secur.*, 2015, pp. 439–450.

[225] L. Li, R. Lu, K.-K. R. Choo, A. Datta, and J. Shao, "Privacy-preserving-outsourced association rule mining on vertically partitioned databases," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 8, pp. 1847–1861, Aug. 2016.

[226] L. Liu, J. Su, R. Chen, X. Liu, X. Wang, S. Chen, and H. Leung, "Privacy-preserving mining of association rule on outsourced cloud data from multiple parties," in *Proc. Australas. Conf. Inf. Secur. Privacy*. Cham, Switzerland: Springer, 2018, pp. 431–451.

[227] O. Regev, "On lattices, learning with errors, random linear codes, and cryptography," *J. ACM*, vol. 56, no. 6, 2009, Art. no. 34.

[228] J. H. Cheon, A. Kim, M. Kim, and Y. Song, "Homomorphic encryption for arithmetic of approximate numbers," in *Proc. Int. Conf. Theory Appl. Cryptol. Inf. Secur.* Cham, Switzerland: Springer, 2017, pp. 409–437.

[229] Y. E. Nesterov, "A method for solving the convex programming problem with convergence rate $O(1/k^2)$," *Sov. Math. Doklady*, vol. 269, pp. 543–547, 1983.

[230] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *J. Comput. Syst. Sci.*, vol. 55, no. 1, pp. 119–139, Aug. 1997.

[231] M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz, "A Bayesian approach to filtering junk e-mail," in *Proc. Learn. Text Categorization: Papers Workshop*, Madison, Wisconsin, vol. 62, 1998, pp. 98–105.

[232] E. Bresson, D. Catalano, and D. Pointcheval, "A simple public-key cryptosystem with a double trapdoor decryption mechanism and its applications," in *Proc. Int. Conf. Theory Appl. Cryptol. Inf. Secur.* Berlin, Germany: Springer, 2003, pp. 37–54.

[233] C.-Y. Hsu, C.-S. Lu, and S.-C. Pei, "Homomorphic encryption-based secure SIFT for privacy-preserving feature extraction," *Proc. SPIE*, vol. 7880, Feb. 2011, Art. no. 788005.

[234] C. Y. Hsu, C. S. Lu, and S. C. Pei, "Image feature extraction in encrypted domain with privacy-preserving SIFT," *IEEE Trans. Image Process.*, vol. 21, no. 11, pp. 4593–4607, Nov. 2012.

[235] S. Hu, Q. Wang, J. Wang, Z. Qin, and K. Ren, "Securing SIFT: Privacy-preserving outsourcing computation of feature extractions over encrypted image data," *IEEE Trans. Image Process.*, vol. 25, no. 7, pp. 3411–3425, Jul. 2016.

[236] D. Li, X. Dong, Z. Cao, and H. Wang, "Privacy-preserving outsourced image feature extraction," *J. Inf. Secur. Appl.*, vol. 47, pp. 59–64, Aug. 2019.

[237] Y. Bai, L. Zhuo, B. Cheng, and Y. F. Peng, "Surf feature extraction in encrypted domain," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2014, pp. 1–6.

[238] Q. Wang, S. Hu, J. Wang, and K. Ren, "Secure surfing: Privacy-preserving speeded-up robust feature extractor," in *Proc. IEEE 36th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jun. 2016, pp. 700–710.

[239] Q. Wang, J. Wang, S. Hu, Q. Zou, and K. Ren, "SecHOG: Privacy-preserving outsourcing computation of histogram of oriented gradients in the cloud," in *Proc. 11th ACM Asia Conf. Comput. Commun. Secur.*, 2016, pp. 257–268.

[240] H. Yang, Y. Huang, Y. Yu, M. Yao, and X. Zhang, "Privacy-preserving extraction of HOG features based on integer vector homomorphic encryption," in *Proc. Int. Conf. Inf. Secur. Pract. Exper.* Cham, Switzerland: Springer, 2017, pp. 102–117.

[241] T. Shortell and A. Shokoufandeh, "Secure feature extraction in computational vision using fully homomorphic encryption," in *Proc. Future Technol. Conf.* Cham, Switzerland: Springer, 2018, pp. 189–213.

[242] Y. Zhang, L. Zhuo, Y. Peng, and J. Zhang, "A secure image retrieval method based on homomorphic encryption for cloud computing," in *Proc. IEEE 19th Int. Conf. Digit. Signal Process.*, Aug. 2014, pp. 269–274.

[243] R. Bellafqira, G. Coatrieux, D. Bouslimi, and G. Quellec, "Content-based image retrieval in homomorphic encryption domain," in *Proc. 37th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Aug. 2015, pp. 2944–2947.

[244] L. Zhang, T. Jung, K. Liu, X.-Y. Li, X. Ding, J. Gu, and Y. Liu, "PIC: Enable large-scale privacy preserving content-based image search on cloud," *IEEE Trans. Parallel Distrib. Syst.*, vol. 28, no. 11, pp. 3258–3271, Nov. 2017.

[245] P. Zheng and J. Huang, "Walsh–Hadamard transform in the homomorphic encrypted domain and its application in image watermarking," in *Proc. Int. Workshop Inf. Hiding*. Berlin, Germany: Springer, 2012, pp. 240–254.

[246] A. V. Subramanyam, S. Emmanuel, and M. S. Kankanhalli, "Robust watermarking of compressed and encrypted JPEG2000 images," *IEEE Trans. Multimedia*, vol. 14, no. 3, pp. 703–716, Jun. 2012.

[247] J. Guo, P. Zheng, and J. Huang, "Secure watermarking scheme against watermark attacks in the encrypted domain," *J. Vis. Commun. Image Represent.*, vol. 30, pp. 125–135, Jul. 2015.

[248] S. Priya, R. Varatharajan, G. Manogaran, R. Sundarasekar, and P. M. Kumar, "Paillier homomorphic cryptosystem with poker shuffling transformation based water marking method for the secured transmission of digital medical images," *Pers. Ubiquitous Comput.*, vol. 22, nos. 5–6, pp. 1141–1151, 2018.

[249] A. Mishra, R. Gupta, and S. Jain, "Secure and robust color image watermarking scheme using partial homomorphic cryptosystem in ASWDR compressed domain," *Multimedia Tools Appl.*, vol. 78, pp. 22127–22154, Aug. 2019.

[250] J. S. Walker and T. Q. Nguyen, "Adaptive scanning methods for wavelet difference reduction in lossy image compression," in *Proc. IEEE Int. Conf. Image Process.*, vol. 3, Sep. 2000, pp. 182–185.

[251] J. Zhou, Z. Cao, X. Dong, and X. Lin, "PPDM: A privacy-preserving protocol for cloud-assisted e-healthcare systems," *IEEE J. Sel. Topics Signal Process.*, vol. 9, no. 7, pp. 1332–1344, Oct. 2015.

[252] R. S. Choras, "Image feature extraction techniques and their applications for CBIR and biometrics systems," *Int. J. Biol. Biomed. Eng.*, vol. 1, no. 1, pp. 6–16, 2007.

[253] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proc. ICCV*, vol. 99, 1999, pp. 1150–1157.

[254] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Comput. Vis. Image Understand.*, vol. 110, no. 3, pp. 346–359, 2008.

[255] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 1, Jun. 2005, pp. 886–893.

[256] M. Schneider and T. Schneider, "Notes on non-interactive secure comparison in 'image feature extraction in the encrypted domain with privacy-preserving SIFT,'" in *Proc. 2nd ACM workshop Inf. Hiding Multimedia Secur.*, 2014, pp. 135–140.

[257] H. Zhou and G. Wornell, "Efficient homomorphic encryption on integer vectors and its applications," in *Proc. IEEE Inf. Theory Appl. Workshop (ITA)*, Feb. 2014, pp. 1–9.

[258] G. Quellec, M. Lamard, G. Cazuguel, B. Cochener, and C. Roux, "Wavelet optimization for content-based image retrieval in medical databases," *Med. Image Anal.*, vol. 14, no. 2, pp. 227–241, 2010.

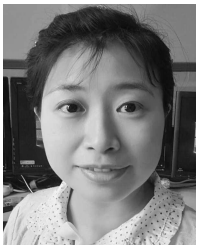
[259] L. Xiao, O. Bastani, and I.-L. Yen, "An efficient homomorphic encryption protocol for multi-user systems," IACR Cryptol. ePrint Arch., Tech. Rep. 193, 2012.

[260] J. B. Fino and V. R. Algazi, "Unified matrix treatment of the fast Walsh–Hadamard transform," *IEEE Trans. Comput.*, vol. C-25, no. 11, pp. 1142–1146, Nov. 1976.

[261] Y. Zhang, W. Dai, X. Jiang, H. Xiong, and S. Wang, "Foresee: Fully outsourced secure genome study based on homomorphic encryption," in *BMC Med. Informat. Decis. Making*, vol. 15, no. 5, p. S5, 2015.

[262] W. Lu, Y. Yamada, and J. Sakuma, "Efficient secure outsourcing of genome-wide association studies," in *Proc. IEEE Secur. Privacy Workshops*, May 2015, pp. 3–6.
[263] W.-J. Lu, Y. Yamada, and J. Sakuma, "Privacy-preserving genome-wide association studies on cloud environment using fully homomorphic encryption," *BMC Med. Inform. Decis. Making*, vol. 15, no. 5, 2015, Art. no. S1.
[264] A. Poon, S. Jankly, and T. Chen, "Privacy preserving Fisher's exact test on genomic data," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2018, pp. 2546–2553.
[265] M. N. Sadat, "Secure and efficient computation on biomedical data in a distributed environment," Ph.D. dissertation, Univ. Manitoba, Winnipeg, MB, Canada, 2018.
[266] J. H. Ziegeldorf, J. Pennekamp, D. Hellmanns, F. Schwinger, I. Kunze, M. Henze, J. Hiller, R. Matzutt, and K. Wehrle, "BLOOM: Bloom filter based oblivious outsourced matchings," *BMC Med. Genomics*, vol. 10, no. 2, 2017, Art. no. 44.
[267] J. H. Cheon, M. Kim, and K. Lauter, "Homomorphic computation of edit distance," in *Proc. Int. Conf. Financial Cryptogr. Data Secur.* Berlin, Germany: Springer, 2015, pp. 194–212.
[268] M. Kim and K. Lauter, "Private genome analysis through homomorphic encryption," *BMC Med. Inform. Decis. Making*, vol. 15, Dec. 2015, Art. no. S3.
[269] M. Yasuda, T. Shimoyama, J. Kogure, K. Yokoyama, and T. Koshiba, "Privacy-preserving wildcards pattern matching using symmetric somewhat homomorphic encryption," in *Proc. Australas. Conf. Inf. Secur. Privacy*. Cham, Switzerland: Springer, 2014, pp. 338–353.
[270] H. Chun, Y. Elmehdwi, F. Li, P. Bhattacharya, and W. Jiang, "Outsourceable two-party privacy-preserving biometric authentication," in *Proc. 9th ACM Symp. Inf., Comput. Commun. Secur.*, 2014, pp. 401–412.
[271] J. Šeděnka, S. Govindarajan, P. Gasti, and K. S. Balagani, "Secure outsourced biometric authentication with performance evaluation on smartphones," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 2, pp. 384–396, Feb. 2015.
[272] S. Hu, M. Li, Q. Wang, S. S. M. Chow, and M. Du, "Outsourced biometric identification with privacy," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 10, pp. 2448–2463, Oct. 2018.
[273] M. Salem, S. Taheri, and J.-S. Yuan, "Utilizing transfer learning and homomorphic encryption in a privacy preserving and secure biometric recognition system," *Computers*, vol. 8, no. 1, p. 3, 2019.
[274] Z. Erkin, M. Franz, J. Guajardo, S. Katzenbeisser, I. Lagendijk, and T. Toft, "Privacy-preserving face recognition," in *Proc. Int. Symp. Privacy Enhancing Technol. Symp.* Berlin, Germany: Springer, 2009, pp. 235–253.
[275] X. Yang, H. Zhu, R. Lu, X. Liu, and H. Li, "Efficient and privacy-preserving online face recognition over encrypted outsourced data," in *Proc. IEEE Int. Conf. Internet Things (iThings), IEEE Green Comput. Commun. (GreenCom), IEEE Cybern, Phys. Social Comput. (CPSCom), IEEE Smart Data (SmartData)*, Jul./Aug. 2018, pp. 366–373.
[276] M. Osadchy, B. Pinkas, A. Jarrous, and B. Moskovich, "SCiFI—A system for secure face identification," in *Proc. IEEE Symp. Secur. Privacy*, May 2010, pp. 239–254.
[277] V. N. Boddeti, "Secure face matching using fully homomorphic encryption," in *Proc. IEEE 9th Int. Conf. Biometrics Theory, Appl. Syst. (BTAS)*, Oct. 2018, pp. 1–10.
[278] M. Barni, T. Bianchi, D. Catalano, M. Di Raimondo, R. D. Labati, P. Failla, D. Fiore, R. Lazzeretti, V. Piuri, and A. Piva, "A privacy-compliant fingerprint recognition system based on homomorphic encryption and fingercode templates," in *Proc. 4th IEEE Int. Conf. Biometrics: Theory, Appl. Syst. (BTAS)*, Sep. 2010, pp. 1–7.
[279] H. Higo, T. Isshiki, K. Mori, and S. Obana, "Privacy-preserving fingerprint authentication resistant to hill-climbing attacks," in *Proc. Int. Conf. Sel. Areas Cryptogr.* Cham, Switzerland: Springer, 2015, pp. 44–64.
[280] M. Blanton and P. Gasti, "Secure and efficient protocols for iris and fingerprint identification," in *Proc. Eur. Symp. Res. Comput. Secur.* Berlin, Germany: Springer, 2011, pp. 190–209.
[281] J. W. Bos, K. Lauter, J. Loftus, and M. Naehrig, "Improved security for a ring-based fully homomorphic encryption scheme," in *Proc. IMA Int. Conf. Cryptogr. Coding*. Berlin, Germany: Springer, 2013, pp. 45–64.
[282] M. Turk and A. Pentland, "Eigenfaces for recognition," *J. Cognit. Neurosci.*, vol. 3, no. 1, pp. 71–86, 1991.
[283] J. Fan and F. Vercauteren, "Somewhat practical fully homomorphic encryption," IACR Cryptol. ePrint Arch., Tech. Rep. 144, 2012.
[284] A. K. Jain, S. Prabhakar, L. Hong, and S. Pankanti, "Filterbank-based fingerprint matching," *IEEE Trans. Image Process.*, vol. 9, no. 5, pp. 846–859, May 2000.
[285] I.-T. Lien, Y.-H. Lin, J.-R. Shieh, and J.-L. Wu, "A novel privacy preserving location-based service protocol with secret circular shift for K-NN search," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 6, pp. 863–873, Jun. 2013.
[286] X. Yi, R. Paulet, E. Bertino, and V. Varadharajan, "Practical k nearest neighbor queries with location privacy," in *Proc. IEEE 30th Int. Conf. Data Eng.*, Mar./Apr. 2014, pp. 640–651.
[287] M. Kesarwani, A. Kaul, P. Naldurg, S. Patranabis, G. Singh, S. Mehta, and D. Mukhopadhyay, "Efficient secure k-Nearest neighbours over encrypted data," in *Proc. EDBT*, 2018, pp. 564–575.
[288] Q. Wang, K. Ren, M. Du, Q. Li, and A. Mohaisen, "SecGDB: Graph encryption for exact shortest distance queries with efficient updates," in *Proc. Int. Conf. Financial Cryptogr. Data Secur.* Cham, Switzerland: Springer, 2017, pp. 79–97.
[289] L. Zhang, J. Li, S. Yang, and B. Wang, "Privacy preserving in cloud environment for obstructed shortest path query," *Wireless Pers. Commun.*, vol. 96, no. 2, pp. 2305–2322, 2017.
[290] M. Shen, B. Ma, L. Zhu, R. Mijumbi, X. Du, and J. Hu, "Cloud-based approximate constrained shortest distance queries over encrypted graphs with privacy protection," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 4, pp. 940–953, Apr. 2018.
[291] J. Wang, Y. Han, X. Yang, T. Zhou, and J. Chen, "A new group location privacy-preserving method based on distributed architecture in LBS," *Secur. Commun. Netw.*, vol. 2019, Feb. 2019, Art. no. 2414687.
[292] M. L. Fredman and R. E. Tarjan, "Fibonacci heaps and their uses in improved network optimization algorithms," *J. ACM*, vol. 34, no. 3, pp. 596–615, 1987.
[293] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numer. Math.*, vol. 1, no. 1, pp. 269–271, Dec. 1959.
[294] K. Lewi and D. J. Wu, "Order-revealing encryption: New constructions, applications, and lower bounds," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2016, pp. 1167–1178.
[295] M. Chase and S. Kamara, "Structured encryption and controlled disclosure," in *Proc. Int. Conf. Theory Appl. Cryptol. Inf. Secur.* Berlin, Germany: Springer, 2010, pp. 577–594.
[296] R. A. Popa, N. Zeldovich, and H. Balakrishnan, "CryptDB: A practical encrypted relational DBMS," Comput. Sci. Artif. Intell. Lab., Cambridge, MA, USA, Tech. Rep. MIT-CSAIL-TR-2011-005, 2011.
[297] A. Boldyreva, N. Chenette, Y. Lee, and A. O'Neill, "Order-preserving symmetric encryption," in *Proc. Annu. Int. Conf. Theory Appl. Cryptographic Techn.* Berlin, Germany: Springer, 2009, pp. 224–241.
[298] S. Tu, M. F. Kaashoek, S. Madden, and N. Zeldovich, "Processing analytical queries over encrypted data," *Proc. VLDB Endowment*, vol. 6, no. 5, pp. 289–300, 2013.
[299] W. K. Wong, B. Kao, D. W. L. Cheung, R. Li, and S. M. Yiu, "Secure query processing with data interoperability in a cloud database environment," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2014, pp. 1395–1406.
[300] Y. Lindell, "Secure multiparty computation for privacy preserving data mining," in *Encyclopedia Data Warehousing Mining*. Philadelphia, PA, USA: IGI Global, 2005, pp. 1005–1009.
[301] G. Liu, G. Yang, H. Wang, Y. Xiang, and H. Dai, "A novel secure scheme for supporting complex SQL queries over encrypted databases in cloud computing," *Secur. Commun. Netw.*, vol. 2018, Jul. 2018, Art. no. 7383514.
[302] K. Xue, S. Li, J. Hong, Y. Xue, N. Yu, and P. Hong, "Two-cloud secure database for numeric-related SQL range queries with privacy preserving," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 7, pp. 1596–1608, Jul. 2017.
[303] K. Cheng, Y. Shen, Y. Wang, L. Wang, J. Ma, X. Jiang, and C. Su, "Strongly secure and efficient range queries in cloud databases under multiple keys," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, Apr./May 2019, pp. 2494–2502.
[304] S. Hou, T. Uehara, S. M. Yiu, L. C. K. Hui, and K.-P. Chow, "Privacy preserving confidential forensic investigation for shared or remote servers," in *Proc. 7th Int. Conf. Intell. Inf. Hiding Multimedia Signal Process.*, 2011, pp. 378–383.
[305] S. Hou, T. Uehara, S. M. Yiu, L. C. K. Hui, and K.-P. Chow, "Privacy preserving multiple keyword search for confidential investigation of remote forensics," in *Proc. IEEE 3rd Int. Conf. Multimedia Inf. Netw. Secur.*, Nov. 2011, pp. 595–599.

[306] Y. Yang and M. Ma, "Conjunctive keyword search with designated tester and timing enabled proxy re-encryption function for e-health clouds," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 4, pp. 746–759, Apr. 2015.
[307] Y. Yang, X. Zheng, V. Chang, S. Ye, and C. Tang, "Lattice assumption based fuzzy information retrieval scheme support multi-user for secure multimedia cloud," *Multimedia Tools Appl.*, vol. 77, no. 8, pp. 9927–9941, 2018.
[308] J. Yu, P. Lu, Y. Zhu, G. Xue, and M. Li, "Toward secure multikeyword top-k retrieval over encrypted cloud data," *IEEE Trans. Dependable Secure Comput.*, vol. 10, no. 4, pp. 239–250, Jul./Aug. 2013.
[309] M. van Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan, "Fully homomorphic encryption over the integers," in *Proc. Annu. Int. Conf. Theory Appl. Cryptograph. Techn.* Berlin, Germany: Springer, 2010, pp. 24–43.
[310] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Inf. Process. Manage.*, vol. 24, no. 5, pp. 513–523, 1988.
[311] M. Strizhov and I. Ray, "Multi-keyword similarity search over encrypted cloud data," in *Proc. IFIP Int. Inf. Secur. Conf.* Berlin, Germany: Springer, 2014, pp. 52–65.
[312] W. Zhang, Y. Lin, and G. Qi, "Catch you if you misbehave: Ranked keyword search results verification in cloud computing," *IEEE Trans. Cloud Comput.*, vol. 6, no. 1, pp. 74–86, Mar. 2015.
[313] Y. Yang, X. Liu, R. H. Deng, and J. Weng, "Flexible wildcard searchable encryption system," *IEEE Trans. Services Comput.*, to be published.
[314] Y. Yang, X. Liu, and R. Deng, "Expressive query over outsourced encrypted data," *Inf. Sci.*, vols. 442–443, pp. 33–53, May 2018.
[315] Unicode Consortium, *The Unicode Standard, Version 2.0*. Reading, MA, USA: Addison-Wesley, 1996.
[316] H. Yao, N. Xing, J. Zhou, and Z. Xia, "Secure index for resource-constraint mobile devices in cloud computing," *IEEE Access*, vol. 4, pp. 9119–9128, 2016.
[317] Z. Pervez, M. Ahmad, A. M. Khattak, N. Ramzan, and W. A. Khan, "OS_2: Oblivious similarity based searching for encrypted data outsourced to an untrusted domain," *PLoS ONE*, vol. 12, no. 7, 2017, Art. no. e0179720.
[318] A. El-Yahyaoui and E. C. H. Dafir, "A verifiable fully homomorphic encryption scheme for cloud computing security," *Technologies*, vol. 7, no. 1, p. 21, 2019.
[319] X. Liu, R. H. Deng, Y. Yang, H. N. Tran, and S. Zhong, "Hybrid privacy-preserving clinical decision support system in fog–cloud computing," *Future Gener. Comput. Syst.*, vol. 78, pp. 825–837, Jan. 2018.
[320] A. B. Slavkovic, Y. Nardi, and M. M. Tibbits, "Secure logistic regression of horizontally and vertically partitioned distributed databases," in *Proc. 7th IEEE Int. Conf. Data Mining Workshops (ICDMW)*, Oct. 2007, pp. 723–728.
[321] M. Aliasgari, M. Blanton, Y. Zhang, and A. Steele, "Secure computation on floating point numbers," in *Proc. NDSS*, 2013, pp. 1–40.



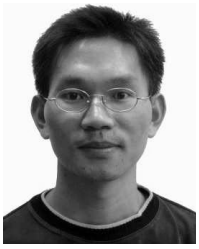
YANG YANG (M'16) received the B.Sc. and Ph.D. degrees from Xidian University, Xi'an, China, in 2006 and 2012, respectively. She is currently an Associate Professor with the College of Mathematics and Computer Science, Fuzhou University.

She has published more than 100 articles in the topics of cloud security and privacy protection-including articles in the IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, the IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, the IEEE TRANSACTIONS ON SERVICE COMPUTING, the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, the IEEE TRANSACTIONS ON CLOUD COMPUTING, and so on. Her research interests include information security and privacy protection. She is a member of CCF.

XINDI HUANG received the B.Sc. degree from the College of Mathematics and Computer Science, Fuzhou University, Fuzhou, China, in 2018, where she is currently pursuing the master's degree. Her current research interest includes privacy preserving computation.

XIMENG LIU (S'13–M'16) received the B.Sc. degree in electronic engineering and the Ph.D. degree in cryptography from Xidian University, Xi'an, China, in 2010 and 2015, respectively. He is currently a Full Professor with the College of Mathematics and Computer Science, Fuzhou University. He has published more than 100 articles on the topics of cloud security and big data security-including articles in the IEEE TRANSACTIONS ON COMPUTERS, the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, the IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, the IEEE TRANSACTIONS ON SERVICE COMPUTING, the IEEE INTERNET OF THINGS JOURNAL, and so on. His current research interests include cloud security, applied cryptography, and big data security.

HONGJU CHENG (M'11) received the B.E. and M.E. degrees in EE from the Wuhan University of Hydraulic and Electric Engineering, in 1997 and 2000, respectively, and the Ph.D. degree in computer science from Wuhan University, in 2007. Since 2007, he has been with the College of Mathematics and Computer Science, Fuzhou University, Fuzhou, China. He is serving as an Editors/Guest Editors of several international journals. He has published almost 60 articles in international journals and conferences. His current research interests include mobile ad hoc networks, wireless sensor networks, and wireless mesh networks.



JIAN WENG received the Ph.D. degree from Shanghai Jiao Tong University, in 2008. From April 2008 to March 2010, he held a postdoctoral position with the School of Information Systems, Singapore Management University. He is currently a Professor and the Executive Dean of the College of Information Science and Technology, Jinan University. He has published more than 60 articles in cryptography conferences and journals such as Eurocryp, Asiacrypt, PKC, and IEEE TIFS.

XIANGYANG LUO received the B.S., M.S., and Ph.D. degrees from the State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou, China, in 2001, 2004, and 2010, respectively. From 2006 to 2007, he was a Visiting Scholar with the Department of Computer Science and Technology of Tsinghua University. He is currently a Full Professor with the State Key Laboratory of Mathematical Engineering and Advanced Computing. He has published more than 100 international journal and conference papers. His current research interest includes networks and information security.

VICTOR CHANG was a Senior Associate Professor and the Director of Ph.D. from June 2016 to May 2018, the Director of MRes from September 2017 to February 2019, and the Interim Director of B.Sc. IMIS Programs from August 2018 to February 2019 with the International Business School Suzhou (IBSS), Xi'an Jiaotong-Liverpool University (XJTLU), Suzhou, China. He has been a Full Professor of data science and information systems with the School of Computing and Digital Technologies, Teesside University, Middlesbrough, U.K., since September 2019. His research interests include big data, cloud computing, and security and applications.