Privacy-Preserving Support Vector Machine Training Over Blockchain-Based Encrypted IoT Data in Smart Cities

Meng Shen, *Member, IEEE*, Xiangyun Tang, Liehuang Zhu, *Member, IEEE*, Xiaojiang Du, *Senior Member, IEEE*, and Mohsen Guizani, *Fellow, IEEE*

***Abstract*—Machine learning (ML) techniques have been widely used in many smart city sectors, where a huge amount of data is gathered from various (IoT) devices. As a typical ML model, support vector machine (SVM) enables efficient data classification and thereby finds its applications in real-world scenarios, such as disease diagnosis and anomaly detection. Training an SVM classifier usually requires a collection of labeled IoT data from multiple entities, raising great concerns about data privacy. Most of the existing solutions rely on an implicit assumption that the training data can be reliably collected from multiple data providers, which is often not the case in reality. To bridge the gap between ideal assumptions and realistic constraints, in this paper, we propose secureSVM, which is a privacy-preserving SVM training scheme over blockchain-based encrypted IoT data. We utilize the blockchain techniques to build a secure and reliable data sharing platform among multiple data providers, where IoT data is encrypted and then recorded on a distributed ledger. We design secure building blocks, such as secure polynomial multiplication and secure comparison, by employing a homomorphic cryptosystem, Paillier, and construct a secure SVM training algorithm, which requires only two interactions in a single iteration, with no need for a trusted third-party. Rigorous security analysis prove that the proposed scheme ensures the confidentiality of the sensitive data for each data provider as well as the SVM model parameters for data analysts. Extensive experiments demonstrates the efficiency of the proposed scheme.**

***Index Terms*—Blockchain, encrypted Internet of Things (IoT) data, homomorphic cryptosystem (HC), machine learning (ML), privacy protection.**

I. Introduction

IN RECENT years, smart cities are incorporating more and more advanced Internet-of-Things (IoT) infrastructures, resulting a huge amount of data gathered from various IoT devices deployed in many city sectors, such as transportation, manufactory, energy transmission, and agriculture [1]. In order to deal with the challenges arising from processing requirements of IoT data, an increasing amount of innovations driven by machine learning (ML) technology have been proposed. Among all ML models, support vector machine (SVM) is a kind of prominent supervised learning models that can efficiently perform data classification. Thus, SVM is adopted in many domains to solve real-world classification problems in IoT-enabled smart cities. In the scenario of personal healthcare, fitness records monitored by wearable IoT sensors can be feeded to SVM classifiers for accurate diagnosis. In the domain of network intrusion detection, SVM classifiers can be used to identify anomalies from a series of traffic data derived from communications among IoT devices [2].

The construction of supervised ML classifiers (e.g., SVM) is known as the training phase, which trains a specific classifier from a set of labeled samples. It is evidenced that the performance of ML classifiers increases as the order of magnitude of training data grows [3], [4]. Since the training dataset owned by a single entity (e.g., a hospitals or a network provider) is usually limited in terms of data volume and variety, there has long been a need for an efficient mechanism to train ML classifiers using a combination of datasets gathered from multiple entities.

However, different entities are usually reluctant to share their data for training due to the concerns about data privacy, integrity, and ownership. First, many training tasks handle sensitive data (e.g., clinic data recorded by medical IoT devices), which may result in sensitive and confidential information leakage during the training process. Second, data records might be tampered with or unauthorized modified by potential attackers during the sharing process, making the resulting ML classifiers inaccurate. And finally, data providers can lose control of their data as shared datasets are accessible to the participants and can be freely replicated by others.

The data privacy issues of training ML classifiers have attracted a significant amount of research attention from both academia and industry. Existing solutions [5]–[8] resort to differential privacy (DP) or cryptographic algorithms to protect the data privacy for each individual provider. They generally rely on an implicit assumption that the training data can be collected reliably from multiple data providers for

Manuscript received September 14, 2018; revised January 8, 2019; accepted February 18, 2019. Date of publication February 26, 2019; date of current version October 8, 2019. This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFB0803405, in part by the National Natural Science Foundation of China under Grant 61602039 and Grant 61872041, in part by the Beijing Natural Science Foundation under Grant 4192050, and in part by CCF-Tencent Open Fund WeBank Special Funding. *(Corresponding author: Liehuang Zhu.)*

M. Shen, X. Tang, and L. Zhu are with the School of Computer Science, Beijing Institute of Technology, Beijing 100081, China (e-mail: shenmeng@bit.edu.cn; tangguotxy@163.com; liehuangz@bit.edu.cn).

X. Du is with the Department of Computer and Information Sciences, Temple University, Philadelphia, PA 19122 USA (e-mail: dxj@ieee.org).

M. Guizani is with the Department of Computer Science and Engineering, Qatar University, Doha, Qatar (e-mail: mguizani@ieee.org).

Digital Object Identifier 10.1109/JIOT.2019.2901840

further analysis, and pay little attention to the concerns of data integrity and ownership. This, however, is not always the case in reality due to potential attacks. To bridge the gap between ideal assumptions and realistic constraints, in this paper, we exploit the blockchain techniques to build a secure data sharing platform. Blockchain is essentially a distributed filing system designed to allow the sharing of tamper-proof records among multiple parties [9]. The permanent and immutable records on blockchain enable data auditing, which can be used to confirm the ownership of data records. This also facilitates quantification of the contribution of individual data providers and designing incentive strategies to encourage sharing of training data.

Although promising, it is still a challenging task to incorporate blockchian into ML training process. The first challenge is designing an appropriate training data format that can be easily accommodated on blockchain while protecting the data privacy of each individual provider. The second challenge lies in the construction of a training algorithm that builds accurate SVM classifiers using the data recorded on blockchian without revealing sensitive information.

To address the above challenges, we propose secureSVM, which is a privacy-preserving SVM training scheme using blockchain-based encrypted IoT data. secureSVM employs a public-key cryptosystem, Paillier, to protect the privacy of IoT data, where data providers encrypt their data locally by their own private keys. Paillier is an additive HC and is more efficient than other algorithms (e.g., Goldwasser-Micali, RSA, and Rabin) in term of encryption and decryption efficiency [10].

Conventional SVM optimization algorithms for unencrypted data, such as the sequential minimal optimization (SMO) algorithm, require intensive comparison operations, which result in a huge amount of interactions in the scenario of encrypted data. Therefore, we resort to a relatively simple optimization algorithm named gradient descent (GD) as the optimization algorithm in secureSVM. Since GD still contains basic operations, such as polynomial operations and comparisons, we design secure polynomial multiplication (SPM) and secure comparison by using the homomorphic properties of Paillier. With the above building blocks, an iteration of secureSVM requires only two interactions, which significantly reduces the computation and communication overhead.

The main contributions of this paper are as follows.

1) We employ the blockchain techniques to enable secure and reliable IoT data sharing. Each IoT data provider can encrypt the data instances locally by its own private key, and then record the encrypted data on blockchain via specially formatted transactions.
2) By using the HC, Paillier, we design secure building blocks, such as SPM and secure comparison, and construct a secure SVM training algorithm, which requires only two interactions in a single iteration, with no need for a trusted third-party.
3) Extensive experiments are conducted to show that our scheme is able to securely train SVM classifiers with a relatively high accuracy. Through rigorous security analysis, we show that the proposed scheme ensures the confidentiality of the sensitive data for each data provider as well as the SVM model parameters for data analysts.

The rest of this paper is organized as follows. We summarize the related work in Section II and describe the preliminaries in Section III. Then, we present the system overview in Section IV and describe the proposed scheme in Section V. After that, we formally analyze the security issues in Section VI and experimentally evaluate the proposed scheme in Section VII. Finally, we conclude this paper in Section VIII.

II. Related Work

In general, supervised learning consists of two phases: 1) the training phase which learns an ML model from a given set of labeled samples and 2) the classification phase which outputs a label with maximum likelihood for a given sample. Accordingly, existing studies on privacy-preserving ML can be broadly classified into two categories, namely privacy-preserving ML training and privacy-preserving ML classification.

A. Privacy-Preserving ML Training

The training of ML models usually get multiple parties involved, thus the privacy goal is to train a model with a collection of the data from all parties while protecting the data of an individual party from being learned by other parties. This paper falls in this category, where plentiful solutions have been proposed during the last decade [6], [11]–[17].

1) Differential Privacy: is a commonly used technique to protect data privacy in the publishing stage [6]. More specifically, DP ensures the security of the published data by adding carefully calculated perturbations to the original data. Abadi *et al.* [6] proposed a DP-based deep learning scheme, which enables multiple parties to jointly learn a neural network while protecting the sensitive information of their datasets.

The DP-based solutions can achieve high computational efficiency, as all calculations are performed over plaintext data. However, the resulting ML models can be inaccurate as the perturbations inevitably reduce the quality of training data. In addition, perturbations may not protect the data privacy completely, as a bounded amount of sensitive information about each individual training data is exposed. For instance, a parameter of privacy budget is employed in DP to balance data privacy and model accuracy: a larger budget helps protect the data privacy, while reducing the model accuracy.

In order to achieve better privacy guarantees in ML training, HC is introduced to train ML models based on encrypted data. It is inherent natures of HC that allow computations on ciphertext while preserving their correctness. Several secure algorithms based on HC have been proposed for training different ML models, including SVM [11], logistic regression [12], [13], decision trees [15], and Naive Bayes [14]. González-Serrano *et al.* [11] developed protocols for secure addition and substraction based on Paillier and constructed a private SVM training algorithm. Since some computations are not supported by Paillier, they employ an authorization server a trusted third-party for computation outsourcing.

Compared with the DP-based solutions, the HC-based ones achieve higher data privacy at the cost of low efficiency. The reasons lie in two aspects.

1) Although the fully homomorphic encryption (FHE) enables complex computations (e.g., with a arbitrary combinations of additions and multiplications) on ciphertexts, the current implementations of FHE lead to prohibitively high cost in terms of encryption and computation, making them impractical in real-world applications.
2) The partially homomorphic encryption (PHE) is more practical compared with FHE but only supports a single type of operation (i.e., addition or multiplication). Therefore, in order to enable complex computations, existing solutions usually rely on a trusted third-party (e.g., the authorization server [11]) or lead to inaccurate models by approximating complex equations with a single type of computation [16], [17].

B. Privacy-Preserving ML Classification

In a *classification-as-a-service* scenario, a data sample for classification and the ML model are usually belonged to two different parties. The data owner wants to know the classification result but is reluctant to expose the sensitive data sample to an untrusted model owner for obtaining classification service. Meanwhile, the model owner may be also reluctant to reveal the information of the classification model as it is a highly valuable asset to the service provider.

To protect the privacy of both sides, efforts have been dedicated to developing efficient solutions [5], [8], [18]–[20]. Wang *et al.* [19] proposed an scheme for classifying encrypted images based on multilayer learning. They rely on an assumption that the image content should be protected, whereas the classifier is not confidential. Zhu *et al.* [20] proposed an privacy-preserving nonlinear SVM classification scheme for online medical prediagnosis. With their design, both the sensitive information of each individual's health record and the SVM model are protected. Rahulamathavan *et al.* [8] proposed a privacy-preserving SVM data classification scheme that performs secure classification for both two-class and multiclass, where the server is unable to learn any knowledge about clients' input data samples while the server side classifier is also kept secret from the clients during the classification process. Several works leveraged the HC techniques to develop a series of classification protocols for applying typical ML classifiers (e.g., decision trees, hyperplane decision, and Naive Bayes) on encrypted data [5], [18].

These studies explored a set of generic classifiers and building blocks to construct privacy-preserving classification schemes. The computation of ML classifiers training phase is more complex than classification phase, these proposed building blocks that can build classification algorithms can be powerless for complex training algorithms.

C. Novelty of This Paper

In this paper, we combine Paillier cryptosystem and blockchain techniques together to address the concerns about data privacy, integrity, and ownership, when training SVM classifiers using IoT data from different providers. More precisely, IoT data of each provider is first encrypted with Paillier and then recorded on a distributed ledger. Data analysts who want to train SVM classifiers can get access to the encrypted data by communicating with the corresponding data providers. Note that data analysts can never obtain the plaintext of any IoT data on blockchain. In order to perform training tasks based on encrypted data, we construct secure protocols of two crucial operations in SVM training, i.e., SPM and secure comparison. Using these building blocks, we propose a privacy-preserving SVM training algorithm, secureSVM, without the need for a trusted third party. Since the training process is based on Paillier, secureSVM is able to train SVM classifiers without loss of accuracy.

We utilize two commonly used security definitions as our security goals: 1) secure two-party computation [21] and 2) modular sequential composition [22]. We demonstrate that with secureSVM, each data provider is unable to learn any knowledge about the data of other data providers, while the data analyst's model parameters is also kept secret from data providers during the training process.

TABLE I
NOTATIONS

Notations	Explanation	Notations	Explanation
D	Dataset	d	Dataset dimension
x_i	i-th record in dataset	y_i	Class lable
∇_t	Gradient	λ	Learning rate
m	Size of dataset D	w, b	The model parameters
$[[m]]$	the encryption of m under Paillier	$\phi(N)$	The Euler phi-function

III. Preliminaries

A. Notation

A dataset D is an unordered set of m records with the size of $|D|$, where x_i is the ith record in D and y_i is the label corresponding to x_i. Define w and b as two relevant parameters of SVM. ∇_t is the descend gradient in the iterative execution of the SVM training algorithm, λ is the learning rate. In this paper, we use a partial HC named Paillier as the cryptosystem, and let $[[m]]$ represent the encryption of m under Paillier. Table I summarizes the notations used in the following sections.

B. Homomorphic Cryptosystem

Cryptosystems consists of three algorithms: 1) key generation (KeyGen); 2) encryption (Enc); and 3) decryption (Dec). A pair of keys (PK, SK) are adopted in public-key cryptosystems, i.e., the public key PK for encryption and the private key SK for decryption.

Homomorphic is a property of cryptosystem if it is capable of mapping the operations over ciphertext to the corresponding plaintext, with no need for the knowledge of the decryption key. Formally, we define homomorphic property of cryptosystem in Definition 1.

Definition 1 (Homomorphic [10]): A public-key encryption scheme (Gen, Enc, Dec) is homomorphic if for all n and all $(\mathsf{PK}, \mathsf{SK})$ output by Gen (1^n), it is possible to define groups $\mathbb{M}$, $\mathbb{C}$ (depending on PK only) such that:

1) the message space is $\mathbb{M}$, and all ciphertexts output by Enc_{pk} are elements of $\mathbb{C}$;
2) for any $m_1, m_2 \in \mathbb{M}$, any c_1 output by $\text{Enc}_{pk}(m_1)$, and any c_2 output by $\text{Enc}_{pk}(m_2)$, it holds that $\text{Dec}_{\text{sk}}(o(c_1, c_2)) = \sigma(m_1, m_1)$.

We apply Paillier a partial HC in our scheme. Paillier is a public key cryptography scheme with homomorphic property that allows two important operations: 1) secure addition and 2) secure subtraction. Paillier based on an assumption related to the hardness of factoring. Let p and q are n-bit primes, $N = pq$. The public key is N, and the private key is $(N, \phi(N))$[1] in Paillier. Encryption function in Paillier is $c := [[(1+N)^m r^N \text{mod} N^2]]$, where $m \in \mathbb{Z}_N$. Decryption function in Paillier is $m := [([c^{\phi(N)} \text{mod} N^2 - 1]/N) \times \phi(N)^{-1} \text{mod} N]$. For more details about Paillier, we refer the reader to [10].

C. Support Vector Machine

SVM is a supervised learning model which gives the maximum-margin hyperplane that might classify the test data [23]. The form of the hyperplane is expressed as $y = w^T x + b$, $(x_i, y_i) \in D$. $w^T x_i + b \geq 1$, $y_i = +1$; $w^T x_i + b \leq 1$, $y_i = -1$. The optimization problem of the primary of SVM as follows:

$$\min_{w,b} \ \frac{1}{2}||w||^2$$
$$\text{s.t.}, y_i\big(w^T x_i + b\big) \ \geq 1, \ \ i = 1, 2, \ldots, m. \quad (1)$$

D. Blockchain Systems

Blockchain is an open and distributed ledger taking the form of a list of *blocks*, which are originally designed for recording transactions in cryptocurrency systems, e.g., Bitcoin. It enables reliable transactions among a group of untrusted participants. In recent year, many different kinds of blockchain platforms, such as the HyperLedger, Ethereum, and EOS, have been proposed and applied to a variety of application scenarios. According to the access restriction on blockchain users, blockchain platforms can be roughly classified into three categories, namely public blockchains, private blockchains, and consortium blockchains.

Blockchain has several desirable features, making it inherently suitable for reliable data sharing.

1) *Decentralized:* As a distributed ledger, blockchian is built on a peer-to-peer network, with no need for a trusted third-party or a central administrator. Multiple replicas of the data recorded in the ledger exist in the system, avoiding data loss in case of single point-of-failure.
2) *Tamper-Proof:* Blockchain employs consensus protocols, such as proof-of-work (PoW), to manage the right of creating new blocks. Thus, data manipulating is usually impractical in terms of computational overhead, making the data recorded in blocks unalterable.
3) *Traceability:* The transactions between two parties in a blockchain system can be easily verified by the rest participants. Any transaction can be tracked and the data owner can benefit in real-time, e.g., getting paid for every bit of the data that is used by a third party.

Though Blockchain has several advantages over other systems, it is far from being perfect when serving as a data sharing platform due to its vulnerability of data privacy to potential attacks. Originally, all transactions are recorded in blocks in the form of plaintexts, making sensitive information in transactions exposing to all participants including adversaries [24]. Therefore, security and privacy concerns should be carefully addressed when employing blockchain as a data sharing platform.

IV. Problem Description

In this section, we describe the problem of secure SVM training over encrypted data gathered from multiple parties, including the system model, threat model, and design goals.

A. System Model

We envision a data-driven IoT ecosystem, shown in Fig. 1, including IoT devices, IoT data providers, blockchain-based IoT platform, and IoT data analysts.

1) IoT devices are capable of sensing and transmitting valuable data through wireless or wired networks, such as ZigBee, 3G/4G, and WiFi. The data can cover a wide range of real-world applications in smart cities, from environmental conditions to physiological information. Note that IoT devices will not participate in the data sharing and analysis processes due to their limited computational capabilities.
2) IoT data providers collect all pieces of data from the IoT devices within their own domains. As a valuable asset of data providers, IoT data usually contains sensitive information. Thus, each data provider encrypts its IoT data using PHE and then records the data on blockchain.
3) Blockchain-based IoT platform serves as a distributed database, where the encrypted IoT data gathered from all data providers are recorded in a shared ledger. Through the built-in consensus mechanism, we can ensure that IoT data is shared in a secure and temper-proof way.
4) IoT data analysts aims at getting a deep insight into the IoT data recorded in the blockchain-based platform, by taking full advantages of emerging analyzing techniques. Data analysts should communicate with corresponding data providers to obtain parameters of training SVM classifiers.

B. Threat Model

According to the system model described in Fig. 1, there exist multiple potential threats to each kind of entities as well as their interactions. Since our efforts are dedicated to designing a privacy-preserving scheme for training SVM models over

[1] Let $N > 1$ be an integer. Then, Z_N^* is an abelian group under multiplication modulo N. Define $\phi(N)$def$|Z_N^*|$, the order of the group Z_N^*.

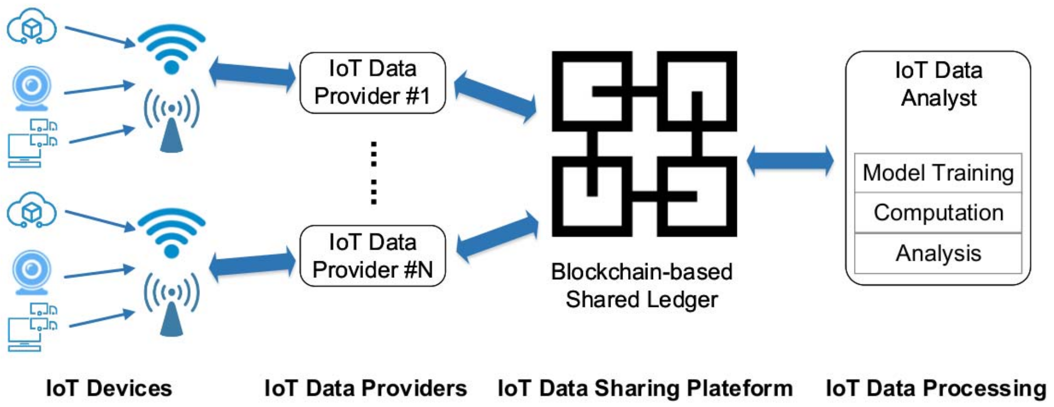

Fig. 1. System model of a data-driven IoT ecosystem.

multiple IoT providers, we only consider the threats to data privacy during the interaction between data providers and data analysts.

We consider the data analyst as an *honest-but-curious* adversary. That is, the data analyst would honestly follow the predesigned ML training protocols, but it may be curious about the contents of the data and attempt to learn additional information by analyzing the encrypted data as well as the intermediate data of computation.

We consider the following two threat models with different attack capabilities that are commonly used in [25] and [26], depending on the sensitive information that can be obtained by the data analyst.

1) *Known Ciphertext Model:* The IoT data analyst can only access the encrypted IoT data recorded in the blockchain-based platform. The IoT data analyst is also capable of recording the intermediate results enerated during the execution of the secure training algorithm, such as the iteration step and descent gradient.
2) *Known Background Model:* In this stronger model, the IoT data analyst is assumed to be aware of more facts than what can be known in the *known ciphertext model*. In particular, the IoT data analyst can collude with one or more IoT data providers to infer the sensitive data of other IoT data providers.

C. Design Goals

We allow any two or more IoT data providers conspire with IoT data analyst to steal the privacy of other participants. We make the following assumptions. Each participate as a honest-but-curious adversary performs protocol honestly but may have interest in the private information of other domains. Any two or more participates may collude with each other. As passive adversaries, they do follow the protocol but try to infer other's privacy as much as possible from the values they learn.

Our scheme aims at protecting privacy of each participant and training an SVM model securely. To be specific, the privacy of each IoT data provider is their IoT records, and IoT data analyst is the SVM model parameters. We specify our security goals as follows.

1) When facing honest-but-curious adversaries, the IoT data analyst and each IoT data provider's privacy are confidential.
2) When facing any two or more parties collude with each other, IoT data analyst and each IoT data provider's privacy are confidential.

V. CONSTRUCTION OF SECURESVM

This section presents the construction details of the proposed privacy-preserving SVM training scheme over blockchain-based encrypted IoT data.

A. System Overview

For simplicity, we assume that a single data analyst aims at training SVM models using the data collected from multiple IoT data providers. The system overview of secureSVM is illustrated in Fig. 2. Each IoT data provider preprocesses IoT data instances, encrypts them locally using their own private keys, and records them in a blockchain-based shared ledger by generating transactions. Existing key management mechanisms [27]–[29] can be employed to manage the encryption capabilities of data providers.

The data analyst who want to train an SVM model can get access to the encrypted data recorded in the global ledger, and assemble a secure training algorithm with several building blocks, such as secure comparison and secure polynomial addition. During the training process, interactions between the data analyst and each data provider are necessary for exchanging intermediate results.

B. Encrypted Data Sharing via Blockchain

Now, we describe the data sharing process. To facilitate model training, without loss of generality, we assume that the data instances for a same training task has been locally preprocessed and represented with a same feature vectors [7].

In order to store the encrypted IoT data in the blockchain, we define a special transaction structure. The transaction format mainly consists of two fields: 1) input and 2) output. The input field includes the data provider's address, the encrypted data, and the IoT device type. The corresponding output field includes the data analyst address, the encrypted data, and the

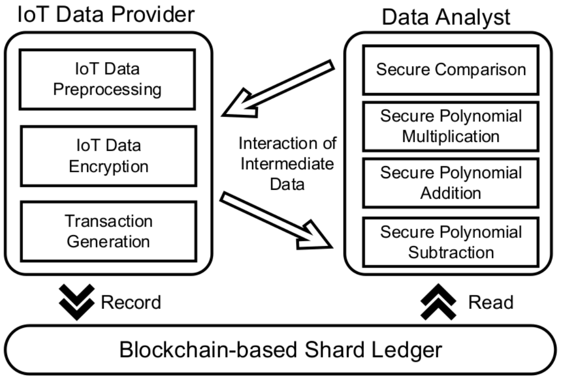
IoT Data Provider
IoT Data Preprocessing
IoT Data Encryption
Transaction Generation
Data Analyst
Secure Comparison
Secure Polynomial Multiplication
Secure Polynomial Addition
Secure Polynomial Subtraction
Interaction of Intermediate Data
Record
Read
Blockchain-based Shard Ledger

Fig. 2. System overview of secureSVM.

IoT device type. The addresses in both fields are a 32-byte hash value. The encrypted data is derived from the homomorphic encryption, Paillier. The length of each encrypted data instance stored in the blockchain is 128 bytes, based on the assumption that the private key length is 128 bytes. The IoT device type segment has a length of 4 bytes.

After constructing a new transaction, a node representing the data provider in the blockchain network broadcasts it in the P2P network, where the miner nodes can verify the correctness of the transaction. Using existing consensus algorithms, such as the PoW mechanism, a specific miner node is qualified for packaging the transaction in a new block and adding the block to the existing chain. Note that each block may consist of multiple transactions.

C. Building Blocks

As described in Section IV, we should ensure privacy of multiple IoT providers, and we aim to designing a privacy-preserving scheme for training SVM models over several private datasets provided by several IoT providers. We now introduce the basic building blocks to achieve these goals.

1) Gradient Descent for SVM: Several optimization methods is able to solve the model parameters of the SVM in (1). These applicative optimization algorithms include the SMO and the GD algorithms. SMO is an optimization approach for the SVM dual quadratic program [30] and performs well for linear SVM and data sparseness. However, the calculation steps of SMO are complex due to a large number of comparisons, dot products, and division operations [11]. Applying SMO in encryption domain may cause horrible computation cost and communication cost. SVM optimization algorithm based on GD is simple and efficient, which involves a small amount of comparison and vector multiplication. Therefor, we choose GD as the optimization algorithm in secure SVM training algorithm to optimize the SVM model parameters in (1).

GD converts the SVM primary into an empirical loss minimization problem with a penalty factor, as

$$\min_{w}, b\frac{1}{2}||w||^2 + C\sum_{i=1}^{m} L(w, b, (x_i, y_i)) \tag{2}$$

where the right part of the equation is the hinge-loss function, $C\sum_{i=1}^{m} L(w, b, (x_i, y_i)) = C\sum_{i=1}^{m} \max\{0, 1 - y_i(wx_i - b)\}$, and C is the misclassification penalty that usually takes the value $(1/m)$.

Algorithm 1 GD Optimization Algorithm

```
Input: Training set D = {$(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)$}, learning rate $\lambda$, maxIters $T$.
Output: $w^*$,$b^*$.
while $cost$ < precision or $t < T$ do
    Compute $\nabla_{t+1}$ by Eq. (3).
    Update $w_{t+1}$,$b_{t+1}$ by $\nabla_{t+1}$.
    Compute $cost$ by Eq. (2).
end while
return $w^*$,$b^*$.
```

TABLE II
CONDITIONS OF SECURE COMPARISON

Input A	Input B	Output A	Output B
$[[a]]$, 1	SK, PK	$(a < 1)$	-

The basic form of GD is: $x_{n+1} = x_n - \lambda\nabla\text{Grad}(x_n)$. The gradient calculation formula of the SVM is exhibited in (3), where $I|(wx + b) < 1|$ is the indication function, and if $(wx + b) < 1$ is true, the value is 1, otherwise it is 0. The steps of the SVM model training algorithm using GD are shown in Algorithm 1

$$\nabla_t = \lambda w_t - \sum_{i=1}^{m} I|(wx_i + b) < 1| \times x_i y_i. \tag{3}$$

2) Secure Polynomial Multiplication: In order to securely train the SVM model, we describe the SPM used in our secure SVM training algorithm.

Using Paillier's homomorphic property, we can obtain secure additions and secure subtractions straightforward. The additively homomorphic properties in Paillier can be described as $[[m_1 + m_2]] = [[m_1]] \times [[m_2]](\text{mod}N^2)$, and the subtraction homomorphic properties can be described as $[[m_1 - m_2]] = [[m_1]] \times [[(m_2)]]^{-1}(\text{mod}N^2)$. $[[m]]^{-1}$ is the modular multiplicative inverse, which preforms that $[[m]] \times [[m]]^{-1}\text{mod}N^2 = 1$ in Paillier. $[[m]]^{-1}$ can be computed by function $\phi(N)$, $[[m]]^{-1} = [[m]]^{\phi(N)-1}$. Thereby, the SPM can be obtained through ciphertext manipulation, as

$$[[am_1 + bm_2]] = [[{m_1}^a]] \times [[{m_2}^b]]\left(\text{mod}N^2\right). \tag{4}$$

The security of SPM constructed by Paillier depends on Paillier's statistically indistinguishable. Thus, SPM is statistically indistinguishable, as Paillier is statistically indistinguishable [10].

3) Secure Comparison: The secure comparison in our secure SVM is defined as comparing an encrypted (and unsigned) number $[[m]]$ with a constant 1, as shown in Table II. For parties A and B participating in the secure comparison algorithm, neither party can obtain any information other than the information implied by the input. Our secure comparison protocol is exhibited in Algorithm 2. We analyze its correctness here, and the security proof is detailed in Section VI.

Correctness: $|r_3 - r_2| < r_1 \leftrightarrow [(|r_3 - r_2|)/r_1] < 1$. $(ar_1 + r_2) = (r_1 + r_3) \leftrightarrow (a - 1) = [(r_3 - r_2)/r_1]$. If $(ar_1 + r_2) >$

```
Algorithm 2 Secure Comparison
Input: A: $[[a]]$,1.
Input: B: A pair of keys (PK,SK)
Output: B: $(a < 1)$.
A uniformly picks three positive integers $r_1$, $r_2$ and $r_3$, where $|r_3 - r_2| < r_1$.
A sends $[[ar_1 + r_2]]$ and $[[r_1 + r_3]]$ to B.
B decrypts and compares $(ar_1 + r_2)$ with $(r_1 + r_3)$, and tell the results to A.
$a > 1$, if and only if $(ar_1 + r_2) > (r_1 + r_3)$; otherwise $a \leq 1$.
return $(a < 1)$
```

```
Algorithm 3 Privacy-Preserving SVM Training Algorithm
P's Input: $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}$.
C's Input: learning rate $\lambda$, maxIters $T$, a pair of keys $(\mathsf{PK_C}, \mathsf{SK_C})$.
C's Output: $w^*$,$b^*$.
C initializes $w^1$, $b^1$.
while $cost$ < precision or $t < T$ do
    for $i = 1$ to $n$ do
        C sends $[[w^t]]$ to $P_i$.
        for $j = 1$ to m do
            $P_i$ computes $[[y_j(wx_j - b)]]$ by secure polynomial multiplication.
            $P_i$ compares $[[y_j(wx_j - b)]]$ with 1 by secure comparison.
            $P_i$ updates $\nabla^i_{t+1}$ and $cost^i_{t+1}$ and send to C.
        end for
        C decrypts $\nabla^i_{t+1}$ and $cost^i_{t+1}$, and updates $w_{t+1}$,$b_{t+1}$.
    end for
end while
return $w^*$, $b^*$.
```

$(r_1 + r_3)$. Because a is an integer, we can infer that $(a - 1) > 1 \rightarrow a > 1$, otherwise $a \leq 1$.

Proposition 1 (Security of Secure Comparison): Algorithm 2 is secure in the honest-but-curious model.

D. Training Algorithm of SecureSVM

For secure optimization model parameters, we design a privacy-preserving SVM training algorithm. Suppose there are n data providers $\mathcal{P}$ and a single IoT data analyst $\mathcal{C}$. Algorithm 3 specifies the secure SVM training algorithm. In Algorithm 3, the sensitive data of IoT data providers and the SVM model parameters are confidential. Except for legal input, each participant cannot infer any sensitive information of other participants from the intermediate results of the algorithm's running process when facing honest-but-curious adversaries or any collusion. The security proofs for each Algorithm 3 are given in Section VI.

Proposition 2 (Security of Privacy-Preserving SVM Training Algorithm): Algorithm 3 is secure in the honest-but-curious model.

VI. Security Analysis

This section presents the security analysis under the *known ciphertext model* and the *known background model*. We adopt two security definitions: 1) secure two-party computation [21] and 2) modular sequential composition [22]. A protocol that satisfies secure two-party computation is secure in the face of honest-but-curious adversaries, and modular sequential composition provides a way to build private protocols in a modular way. We present our security proof according to the ideas of these two definitions. For more details, we refer the reader to [21] for secure two-party computation and [22] for modular sequential composition.

We follow the notation in [5]: Let $F = (F_A, F_B)$ be a polynomial function and π a protocol computing F; a is A's input and b is B's input and A and B desire to compute $F(a, b)$, using π; the view of A is the tuple $\mathrm{view}^{\pi}_{A}(\lambda, a, b) = (\lambda; a; m_1, m_2, \dots, m_n)$ where $m_1, m_2, \dots, m_n$ are the messages received by A when the execution. We define the view of B similarly. A's and B's outputs are $\mathrm{output}^{\pi}_{A}(a, b)$ and $\mathrm{output}^{\pi}_{B}(a, b)$, respectively. The global output of π is $\mathrm{output}^{\pi}(a, b) = (\mathrm{output}^{\pi}_{A}(a, b), \mathrm{output}^{\pi}_{B}(a, b))$.

Definition 2 (Secure Two-Party Computation [21]): A protocol π privately computes f with statistical security if for all possible inputs (a, b) and simulators S_A and S_B hold the following properties:[2]

$$\{S_A, f_2(a, b)\} \approx \left\{\mathrm{view}^{\pi}_{A}(a, b), \mathrm{output}^{\pi}(a, b)\right\}$$
$$\{f_1(a, b), S_B\} \approx \left\{\mathrm{output}^{\pi}(a, b), \mathrm{view}^{\pi}_{B}(a, b)\right\}.$$

The basic idea of sequential modular composition is that: n participants run a protocol π calling to an ideal functionality F, e.g., A and B compute F privately by sending their inputs to a trusted third-party and receiving the result. If we can prove that the protocol π satisfies secure two-party computation, and a protocol ρ can achieve the same function as F privately, then we can replace the ideal protocol for F by the protocol of ρ in π; the new protocol π^{ρ} is then secure under the honest-but-curious model [5], [22].

Theorem 1 (Modular Sequential Composition [21]): Let $F_1, F_2, \dots, F_n$ be two-party probabilistic polynomial time functionalities and $\rho_1, \rho_2, \dots, \rho_n$ protocols that compute, respectively, $F_1, F_2, \dots, F_n$ in the presence of semi-honest adversaries. Let G be a two-party probabilistic polynomial time functionality and π a protocol that securely computes G in the $(F_1, F_2, \dots, F_n)$-hybrid model in the presence of semi-honest adversaries. Then, $\pi^{\rho_1, \rho_2, \dots, \rho_n}$ securely computes G in the presence of semi-honest adversaries.

A. Security Proof for Secure Comparison

Two roles are involved in Algorithm 3: A and B. The function is $F : F([[a]]_B, 1, \mathsf{PK_B}, \mathsf{SK_B}) = (\phi, (\mathsf{a} < 1))$.

Proof of Proposition 1: The view of A is $\mathrm{view}^{\pi}_{A} = ([[a]]_B, \mathsf{PK_B})$. As A does not receive any message form B, her view only consists of her input and three random numbers she produces. Hence, the simulator $S^{\pi}_{A}((a, 1); F(a, 1)) = \mathrm{view}^{\pi}_{A}([[a]]_B, 1, \mathsf{PK_B})$. $[[a]]_B$ is encrypted by $\mathsf{PK_B}$, and the confidentiality of $[[a]]_B$ is equivalent to the used cryptosystem Paillier. Therefore, A cannot infer the value directly.

[2] $\approx$ denotes computational indistinguishability against probabilistic polynomial time adversaries with negligible advantage in the security parameter λ.

The view of B is $\text{view}_B^{\pi} = ((ar_1+r_2), (r_1+r_3), \mathsf{PK}_{\mathsf{B}}, \mathsf{SK}_{\mathsf{B}})$. S_B^{π} runs as follows.

1) Generates l random coins and obtains $[[(m_1, m_2, \ldots, m_l)]]_B$ by PK_{B}, where l is the length of a.
2) B uniformly picks three positive integers c_1, c_2, and c_3, where $|r_3 - r_2| < r_1$.
3) Outputs $((mc_1 + c_2), (c_1 + r_3), \mathsf{PK}_{\mathsf{B}}, \mathsf{SK}_{\mathsf{B}})$.

The distribution of (a, r_1, r_2, r_3) and (m, c_1, c_2, c_3) are identical, so the real distribution $((ar_1+r_2), (r_1+r_3), \mathsf{PK}_{\mathsf{B}}, \mathsf{SK}_{\mathsf{B}})$ and the ideal distribution $((mc_1 + c_2), (c_1 + c_3), \mathsf{PK}_{\mathsf{B}}, \mathsf{SK}_{\mathsf{B}})$ are statistically indistinguishable. ■

B. Security Proof for Privacy-Preserving SVM Training Algorithm

The roles involved in Algorithm 3 are n IoT data providers P and an IoT data analyst C. Each IoT data providers behaves in the same way. If we can prove that one of them is meets the security requirements, then every data provider meets the security requirements. The function is $F : F(D_{P_i}, \mathsf{PK}_{\mathsf{C}}, \mathsf{SK}_{\mathsf{C}}) = (\phi, (\mathsf{w}_*, \mathsf{b}_*))$.

Proof of Proposition 2: Each IoT data provider's view is $\text{view}_A^{\pi} = (D_{P_i}, [[w^t]]_C, \mathsf{PK}_{\mathsf{C}})$. $[[w^t]]_C$ is encrypted by PK_{C}, and the confidentiality of $[[w^t]]_C$ is equivalent to the used cryptosystem Paillier. So each IoT data provider cannot infer the value directly. Hence, the simulator $S_P^{\pi}(D_{P_i}; F(a, 1)) = \text{view}_P^{\pi}(D_{P_i}, [[w^t]]_C, \mathsf{PK}_{\mathsf{C}})$.

IoT data analyst's view is $\text{view}_C^{\pi} = (\sum_{i=1}^{m} I|(wx_i + b) < 1| \times x_i y_i, \sum_{i=1}^{m} \max\{0, 1 - y_i(wx_i - b)\}, w_t, \mathsf{PK}_{\mathsf{B}}, \mathsf{SK}_{\mathsf{B}})$. Now, we need to discuss the confidentiality of $\sum_{i=1}^{m} I|(wx_i + b) < 1| \times x_i y_i$ and $\sum_{i=1}^{m} \max\{0, 1 - y_i(wx_i - b)\}$; that is, whether IoT data analyst can guess the sensitive information of each IoT data provider from the two equations.

Obviously, both of the equations are no-solution equations for the unknown x_i and y_i. Except for brute force cracking, there is no better way to get the real value of dataset D. We assume that each IoT data provider with a small dataset has 2-D 100 instances, and each dimension is 32 bits.[3] Under this circumstance, the probability that IoT data analyst guesses success is $(1/2^{n*6400})$. It is a negligible probability of success [10].

As Algorithm 2 and SPM used in Algorithm 3 are secure in the honest-but-curious model, we obtain the security of Algorithm 3 using modular sequential composition. ■

VII. Performance Evaluation

In this section, we evaluate the performance of secureSVM in terms of accuracy and efficiency through extensive experiments using real-world datasets. We first describe the experiment settings and then exhibit the experimental results to demonstrate its effectiveness and efficiency.

TABLE III
Statistics of Datasets

Datasets	Instances number	Attributes number	Discrete attributes	Numerical attributes
BCWD	699	9	0	9
HDD	294	13	13	0

TABLE IV
Summary of Accuracy Performance

Datasets	Precision		Recall	
	secureSVM	*SVM*	secureSVM	*SVM*
BCWD	90.35%	90.47%	96.19%	97.24%
HDD	93.89%	93.35%	89.78	90.87%

A. Experiment Setup

1) Testbed: In our design, each IoT data provider collects all pieces of data from the IoT devices in its own domain and then performs the following operations (e.g., data encryption) on the IoT data. Since IoT providers and the data analyst usually have adequate computing resources, the experiments are run on a PC equipped with a 4-core Intel i7 (i7-3770 64bit) processor at 3.40 GHz and 8 GB RAM, serving as IoT data providers and an IoT data analysts simultaneously. We have implemented the SPM, secure comparison, and the secureSVM in Java Development Kit 1.8.

2) Dataset: To implement the method for this paper, we use two real-world datasets, namely Breast Cancer Wisconsin Data Set (BCWD) [31] and Heart Disease Data Set (HDD) [32], which are publicly available from UCI ML repository. The features of BCWD are computed from a digitized image of a fine needle aspirate of a breast mass and describe characteristics of the cell nuclei present in the image. Each instance is labeled as benign or malignant. The HDD consists of 13 numeric attributes, and all the instance are classified by the types of heart diseases. The statistics are shown in Table III. We show the average results of cross-validation of ten runs to avoid overfitting or contingent results. In each cross-validation, we randomly take 80% to train the model, and the remainder for testing.

3) Float Format Conversion: The general SVM training algorithms performs on floating point numbers. However, the operations of cryptosystem are carried out on integers. In order to encrypted data taking real values, it is necessary to previously perform a format conversion into an integer representation. Binary floating point number D is expressed as $D = (-1)^s \times M \times 2^E$ in the international standard IEEE 754, where s is the sign bit, M is a significant number, and E is the exponent bit. In our implementation of secureSVM, we employ it to perform the format conversion.

4) Key Length Setting: In public-key cryptosystems, the length of keys is closely related to the security of the cryptosystem, and short key may lead to an unsecure encryption. In particular, homomorphic operations (i.e., the SPM) runs on the ciphertext, a long key reduce the efficiency of the homomorphic operation, and a too short key may cause the plaintext space to overflow. Therefore,

[3] Typically, single-precision floating-point occupies 4 bytes (32-bit) memory space.

TABLE V
PERFORMANCE OF THE BUILDING BLOCKS IN SECURESVM

Datasets	Total Time	$\mathcal{P}$ Time	$\mathcal{C}$ Time	comparison	SPM
BCWD	3195s	2233s	953s	1769s	3072s
HDD	1935s	1324s	601s	1050s	1825s

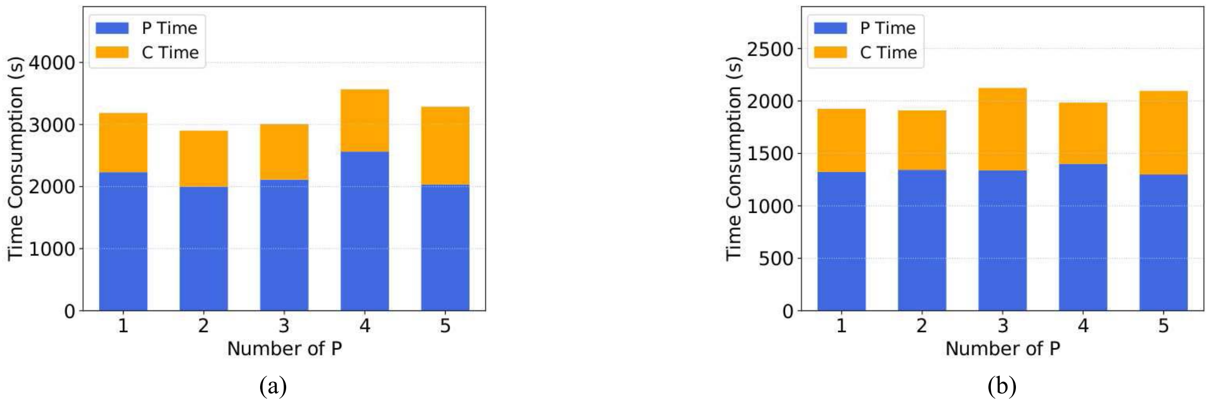

Fig. 3. Time consumption of secureSVM with different numbers of data providers $\mathcal{P}$. (a) BCWD. (b) HDD.

we must consider the key length to avoid the possibility of overflow. In secureSVM, Paillier's key N is set to 1024-bit.

B. Accuracy

We adopt two commonly used criterions for evaluating ML classifiers. Given a dataset for validation, the precision $\mathcal{P}$ is calculated as $\mathcal{P} = t_p/(f_p+t_p)$, and the recall $\mathcal{R}$ is calculated as $\mathcal{R} = t_p/(f_n+t_p)$, where t_p is the numbers of relevant (i.e., the positive class) that are classified correctly, f_p is the numbers of irrelevant (i.e., the negative class) that are classified correctly and f_n is the numbers of relevant that are classified incorrectly in the test results.

To illustrate that secureSVM does not reduce the accuracy upon protecting the privacy of each IoT data provider and securely training classifiers, we implemented the general SVM with Java named SVM. Because we focuses on how to securely training a classifier in this paper, we do not adjust the training parameters and use the default parameters. Table IV summarizes the results of precision and recall.

Compared with the SVM, secureSVM has almost the same accuracy as SVM, which does not reduce the accuracy of classifiers. BCWD is a dataset with all numerical attributes, and HDD is a dataset with all discrete attributes. Our scheme shows good robustness on both types of datasets.

C. Efficiency

1) Building Blocks Evaluation: Table V gives the running time of the secure comparison and SPM with encrypted datasets on Algorithm 3. Table V also gives the time consumption of IoT data providers $\mathcal{P}$ and data analysis $\mathcal{C}$, the total time consumption.

As the performance results in Table V, secureSVM trains SVM classifiers spending less than an hour with encrypted dataset BCWD and HDD, which has the acceptable time consumption. In this experiments, several $\mathcal{P}$ is simulated linearly. Thus, the $\mathcal{P}$ time exhibits in Table V is the accumulation of time spent by several $\mathcal{P}$. In actual application, we could set that several $\mathcal{P}$ general running algorithms in parallel, so that the time consumption of $\mathcal{P}$ and the total time consumption can be decreased sharply. In order to better demonstrate the performance of secureSVM in terms of time consumption, we just show the raw running time. We believe Algorithm 3 to be practical for sensitive applications.

Facing with the different types of databases, a all numerical attributes dataset BCWD or a all discrete attributes dataset HDD, secureSVM shows good robustness in terms of time consumption.

2) Scalability Evaluation: secureSVM assumes that there are several IoT data providers participating and providing date. To evaluate the scalability of our scheme, we divide the dataset into several equal parts to simulate the several IoT data providers scenarios, and observe the changes in time consumption when different numbers of IoT data providers participate in the computation. We simulate the cases, where the number of IoT data providers increases from 1 to 5. The results are displayed in Fig. 3, the abscissa presents the number of IoT data providers involved in the computation, and the ordinate is the time consumption.

Theoretically, it is intuitive that the time consumption of secureSVM is only related to the amount of data and the number of iterations in the GD algorithm. When the total amount of data and data quality are unchanged, the increase in the number of $\mathcal{P}$ does not affect the time consumption. We observe from the results that the time consumption of $\mathcal{P}$ or $\mathcal{C}$ does not change with the change in the amount of the number of $\mathcal{P}$. When the number of $\mathcal{P}$ increases from 1 to 5, the total time consumption has a slight fluctuation, because the running time of the program is disturbed by other processes in the host used for the simulation.

VIII. Conclusion

In this paper, we presented a novel privacy-preserving SVM training scheme named secureSVM, which tackled the challenges of data privacy and data integrity by employing the blockchain techniques to build a secure SVM training algorithm in multipart scenarios, where IoT data is collected from multiple data providers. HC Paillier is used to construct an efficient and accurate privacy-preserving SVM training algorithm. We demonstrated the efficiency and security of secureSVM. In the future work, we plan to develop a generalized framework which enables constructing a wide range of privacy-preserving ML training algorithms on multipart encrypted datasets.

References

[1] Y. Zhang *et al.*, "Cognitive machine-to-machine communications: Visions and potentials for the smart grid," *IEEE Netw.*, vol. 26, no. 3, pp. 6–13, May/Jun. 2012.

[2] M. Shen, M. Wei, L. Zhu, and M. Wang, "Classification of encrypted traffic with second-order Markov chains and application attribute bigrams," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 8, pp. 1830–1843, Aug. 2017.

[3] C. Sun, A. Shrivastava, S. Singh, and A. Gupta, "Revisiting unreasonable effectiveness of data in deep learning era," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 843–852.

[4] M. Shen *et al.*, "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.

[5] R. Bost, R. A. Popa, S. Tu, and S. Goldwasser, "Machine learning classification over encrypted data," in *Proc. Netw. Distrib. Syst. Security Symp.*, 2014, pp. 1–14.

[6] M. Abadi *et al.*, "Deep learning with differential privacy," in *Proc. ACM SIGSAC Conf. Comput. Commun. Security (CCS)*, 2016, pp. 308–318.

[7] Q. Wang, S. Hu, M. Du, J. Wang, and K. Ren, "Learning privately: Privacy-preserving canonical correlation analysis for cross-media retrieval," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, May 2017, pp. 1–9.

[8] 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. Depend. Secure Comput.*, vol. 11, no. 5, pp. 467–479, Sep./Oct. 2014.

[9] H. Li *et al.*, "Blockchain-based data preservation system for medical data," *J. Med. Syst.*, vol. 42, no. 8, p. 141, Jun. 2018.

[10] J. Katz and Y. Lindell, *Introduction to Modern Cryptography*. Boca Raton, FL, USA: CRC Press, 2014.

[11] F.-J. González-Serrano, Á. Navia-Vázquez, and A. Amor-Martín, "Training support vector machines with privacy-protected data," *Pattern Recognit.*, vol. 72, pp. 93–107, Dec. 2017.

[12] M. de Cock, R. Dowsley, A. C. A. Nascimento, and S. C. Newman, "Fast, privacy preserving linear regression over distributed datasets based on pre-distributed data," in *Proc. 8th ACM Workshop Artif. Intell. Security (AISec)*, 2015, pp. 3–14.

[13] T. Graepel, K. Lauter, and M. Naehrig, "ML confidential: Machine learning on encrypted data," in *Information Security and Cryptology—ICISC*. Heidelberg, Germany: Springer, 2012.

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

[15] J. Vaidya, B. Shafiq, W. Fan, D. Mehmood, and D. Lorenzi, "A random decision tree framework for privacy-preserving data mining," *IEEE Trans. Depend. Secure Comput.*, vol. 11, no. 5, pp. 399–411, Sep./Oct. 2014.

[16] 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, no. 8, pp. 2079–2089, 2016.

[17] Y. Aono, T. Hayashi, L. T. Phong, and L. Wang, "Scalable and secure logistic regression via homomorphic encryption," in *Proc. 6th ACM Conf. Data Appl. Security Privacy (CODASPY)*, 2016, pp. 142–144.

[18] M. De Cock *et al.*, "Efficient and private scoring of decision trees, support vector machines and logistic regression models based on precomputation," *IEEE Trans. Depend. Secure Comput.*, to be published. doi: 10.1109/TDSC.2017.2679189.

[19] W. Wang, C.-M. Vong, Y. Yang, and P.-K. Wong, "Encrypted image classification based on multilayer extreme learning machine," *Multidimensional Syst. Signal Process.*, vol. 28, no. 3, pp. 851–865, Jul. 2017.

[20] H. Zhu, X. Liu, R. Lu, and H. Li, "Efficient and privacy-preserving online medical prediagnosis framework using nonlinear SVM," *IEEE J. Biomed. Health Informat.*, vol. 21, no. 3, pp. 838–850, May 2017.

[21] O. Goldreich, *Foundations of Cryptography: Volume 2, Basic Applications*. Cambridge, U.K.: Cambridge Univ. Press, 2009.

[22] R. Canetti, "Security and composition of multiparty cryptographic protocols," *J. Cryptol.*, vol. 13, no. 1, pp. 143–202, Jan. 2000.

[23] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, Sep. 1995.

[24] F. Gao *et al.*, "A blockchain-based privacy-preserving payment mechanism for vehicle-to-grid networks," *IEEE Netw.*, vol. 32, no. 6, pp. 184–192, Nov./Dec. 2018.

[25] M. Shen, B. Ma, L. Zhu, X. Du, and K. Xu, "Secure phrase search for intelligent processing of encrypted data in cloud-based IoT," *IEEE Internet Things J.*, to be published. doi: 10.1109/JIOT.2018.2871607.

[26] L. Zhu, X. Tang, M. Shen, X. Du, and M. Guizani, "Privacy-preserving DDoS attack detection using cross-domain traffic in software defined networks," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 3, pp. 628–643, Mar. 2018.

[27] X. Du, M. Guizani, Y. Xiao, and H.-H. Chen, "A routing-driven elliptic curve cryptography based key management scheme for heterogeneous sensor networks," *IEEE Trans. Wireless Commun.*, vol. 8, no. 3, pp. 1223–1229, Mar. 2009.

[28] Y. Xiao *et al.*, "A survey of key management schemes in wireless sensor networks," *Comput. Commun.*, vol. 30, nos. 11–12, pp. 2314–2341, 2007.

[29] X. Du, Y. Xiao, M. Guizani, and H.-H. Chen, "An effective key management scheme for heterogeneous sensor networks," *Ad Hoc Netw.*, vol. 5, no. 1, pp. 24–34, 2007.

[30] J. Platt, "Sequential minimal optimization: A fast algorithm for training support vector machines," Microsoft Res., Bengaluru, India, Rep. MSR-TR-98-14, pp. 1–21, Apr. 1998.

[31] D. Dua and C. Graff, "UCI machine learning repository," School Inf. Comput. Sci., Univ. California at Irvine, Irvine, CA, USA, 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[32] R. Detrano *et al.*, "International application of a new probability algorithm for the diagnosis of coronary artery disease," *Amer. J. Cardiol.*, vol. 64, no. 5, pp. 304–310, 1989.

Meng Shen (M'14) received the B.Eng. degree in computer science from Shandong University, Jinan, China, in 2009, and the Ph.D. degree in computer science from Tsinghua University, Beijing, China, in 2014.

He currently serves with the Beijing Institute of Technology, Beijing, China, as an Associate Professor. His current research interests include privacy protection for cloud and IoT, blockchain applications, and encrypted traffic classification.

Dr. Shen was a recipient of the Best Paper Runner-Up Award of IEEE IPCCC in 2014.

Xiangyun Tang received the B.Eng. degree in computer science from the Minzu University of China, Beijing, China, in 2016. She is currently pursuing the Ph.D. degree at the Department of Computer Science, Beijing Institute of Technology, Beijing.

Her current research interests include differential privacy and secure multiparty computation.

Liehuang Zhu (M'16) is a Professor with the Department of Computer Science, Beijing Institute of Technology, Beijing, China. He was selected into the Program for New Century Excellent Talents in University from the Ministry of Education, Beijing. His current research interests include Internet of Things, cloud computing security, Internet and mobile security.

Xiaojiang Du (S'99–M'03–SM'09) received the B.S. and M.S. degrees in electrical engineering from Tsinghua University, Beijing, China, in 1996 and 1998, respectively, and the M.S. and Ph.D. degrees in electrical engineering from the University of Maryland at College Park, College Park, MD, USA, in 2002 and 2003, respectively.

He is a Tenured Professor with the Department of Computer and Information Sciences, Temple University, Philadelphia, PA, USA. His current research interests include wireless communications, wireless networks, security, and systems. He has authored over 300 journal and conference papers in the above areas, as well as a book published by Springer.

Dr. Du has been awarded over $5 million in research grants from the U.S. National Science Foundation, Army Research Office, Air Force, NASA, the State of Pennsylvania, and Amazon, the Best Paper Award of IEEE GLOBECOM 2014 and the Best Poster Runner-Up Award of ACM MobiHoc 2014. He serves on the Editorial Boards of three international journals. He is a Life Member of the ACM.

Mohsen Guizani (S'85–M'89–SM'99–F'09) received the B.S. (with Distinction) and M.S. degrees in electrical engineering and the M.S. and Ph.D. degrees in computer engineering from Syracuse University, Syracuse, NY, USA, in 1984, 1986, 1987, and 1990, respectively.

He served in different academic and administrative positions with the University of Idaho, Moscow, ID, USA, Western Michigan University, Kalamazoo, MI, USA, University of West Florida, Pensacola, FL, USA, University of Missouri–Kansas City, Kansas City, MO, USA, University of Colorado–Boulder, Boulder, CO, USA, and Syracuse University, Syracuse, NY, USA. He is currently a Professor with the Department of Computer Science and Engineering, Qatar University, Doha, Qatar. He has authored 9 books and over 500 publications in refereed journals and conferences. His current research interests include wireless communications and mobile computing, computer networks, mobile cloud computing, security, and smart grid.

Dr. Guizani was a recipient of three teaching awards and four research awards, the 2017 IEEE Communications Society WTC Recognition Award, as well as the 2018 AdHoc Technical Committee Recognition Award for his contribution to outstanding research in wireless communications and ad-hoc sensor networks. He is currently the Editor-in-Chief of *IEEE Network*, serves on the Editorial Boards of several international technical journals and is the founder and the Editor-in-Chief of the *Wireless Communications and Mobile Computing Journal* (Wiley). He has guest edited a number of special issues in IEEE publications. He also served as a member, the Chair, and the General Chair of a number of international conferences. He was the Chair of the IEEE Communications Society Wireless Technical Committee and the Chair of the TAOS Technical Committee. He served as the IEEE Computer Society Distinguished Speaker and is currently the IEEE ComSoc Distinguished Lecturer. He is a Senior Member of the ACM.