Hindawi
Wireless Communications and Mobile Computing
Volume 2018, Article ID 2385150, 10 pages
https://doi.org/10.1155/2018/2385150

Research Article

Securely Outsourcing ID3 Decision Tree in Cloud Computing

Ye Li,[1] Zoe L. Jiang,[1] Xuan Wang,[1] Junbin Fang,[2] En Zhang,[3] and Xianmin Wang[4]

[1]*Harbin Institute of Technology, Shenzhen, Shenzhen 518055, China*
[2]*Jinan University, Guangzhou, China*
[3]*Henan Normal University, Henan, China*
[4]*School of Computer Science, Guangzhou University, China*

Correspondence should be addressed to Zoe L. Jiang; zoeljiang@hit.edu.cn

Received 2 May 2018; Accepted 2 September 2018; Published 4 October 2018

Academic Editor: Jaime Lloret

With the wide application of Internet of Things (IoT), a huge number of data are collected from IoT networks and are required to be processed, such as data mining. Although it is popular to outsource storage and computation to cloud, it may invade privacy of participants' information. Cryptography-based privacy-preserving data mining has been proposed to protect the privacy of participating parties' data for this process. However, it is still an open problem to handle with multiparticipant's ciphertext computation and analysis. And these algorithms rely on the semihonest security model which requires all parties to follow the protocol rules. In this paper, we address the challenge of outsourcing ID3 decision tree algorithm in the malicious model. Particularly, to securely store and compute private data, the two-participant symmetric homomorphic encryption supporting addition and multiplication is proposed. To keep from malicious behaviors of cloud computing server, the secure garbled circuits are adopted to propose the privacy-preserving weight average protocol. Security and performance are analyzed.

1. Introduction

In the modern Internet of Things (IoT), huge data are collected from sensor-networks and need to be provided for analysis by high-effective techniques, such as data mining. This process requires enormous computation and storage to support; cloud computing technology can provide the corresponding support. However, this process may leak the privacy of participants' information. The privacy-preserving data mining (PPDM) based on encryption method has emerged as a solution to this problem.

Privacy-Preserving Data Mining Framework. Considering different frameworks and theories, PPDM was originated by Lindell et al. [1] and Agrawal et al. [2] in 2002, respectively. Lindell's framework is essentially a secure cryptography-based two-participant computation protocol without outsourcing. In other words, two parties can interactively compute $(x_1 + x_2)\ln(x_1 + x_2)$ on their private input x_1 and x_2. Agrawal's framework is essentially a single-participant disturbance-based data storage and computation outsourcing algorithm. In particular, one party can upload disturbed data to server for private computation. With the development of cloud computation and IoT, a multiparty storage and computation outsourcing framework is preferred.

Cryptography-based privacy-preserving data mining supporting one-party outsourcing has been studied [3, 4], with homomorphic encryption. However, multiple-key homomorphic encryption is an open problem when multiple parties are involved in the outsourcing framework. For example, how to execute ciphertext addition and multiplication on ciphertexts encrypted by different public keys?

Security Models. We usually consider two different security models, including the semihonest and malicious security model. The definition in the semihonest model requires that all the users need to follow the rules of protocol. But we allow the dishonest users to obtain internal states of the other users. In the malicious model, different from the first security model, the corrupted users are allowed to deviate from the specified protocol. The success of the adversary means that the adversary can get the results of these protocols.

Data Distribution. Three types of distributed datasets are defined in related works, including the horizontally distributed datasets, vertically distributed datasets, and arbitrarily distributed datasets. The users in the horizontally distributed dataset can keep divided parts for the same attributes. However, in the vertical datasets, users are allowed to keep different attributes. In the last one, the datasets can be arbitrarily divided and stored by the users.

Due to the existence of malicious participants in the real environment, malicious participants may not follow the protocol. For example, they can intentionally tamper with the data, suspend the protocol anytime during the execution of the protocol, and so on. To solve this problem, this paper combines the noncontact commitment and confusion circuit mechanism, studies the average computing protocol based on confusion circuit, and then proposes the framework of a secure cryptography-based two-participant protocol with data storage and computation outsourcing. The framework consists of two data owners and two cloud servers (cloud storage server (CSS) and cloud computing server (CCS)). Each data owner has a horizontally distributed private database that is encrypted before being outsourced to the cloud for storage and computation.

1.1. Our Contribution. We decompose the key function of distributed ID3 decision tree, $Gain(S, A_i)$, into counting, $(x_1 + x_2)/(a_1 + a_2)$, sum, multiplication, and comparison.

In counting, we propose the Secure Equivalent Testing (SET) protocol to calculate the number of items for each attribute value based on the encrypted data.

To calculate the value of $(x_1 + x_2)/(a_1 + a_2)$ in malicious model, we implement the Outsourcing Secure Circuit (OSC) protocol.

To perform the sum and multiplication operations over ciphertext, we adopt the Paillier encryption system and implement the Secure Multiplication (SM) protocol.

To execute comparison over ciphertext, we adopt the Secure Minimum out of 2 Numbers (SMIN2) protocol.

1.2. Related Work

Distributed PPDM without Outsourcing. Distributed PPDM without outsourcing is mainly for data stored and calculated locally by the participant, based on distributed data based on various data mining methods, which can be decomposed to different operations, such as average calculation, calculation, and calculation of logarithmic vector inner product. Then the cryptography-based technology is used to design various privacy-preserving computing protocols. In 2002, Lindell and Pinkas [1] proposed a secure ID3 decision tree algorithm over horizontally partitioned data. They decompose the distributed ID3 algorithm to multilogarithmic calculation, polynomial evaluation calculation, and data comparison, and then designed the security log protocol, polynomial evaluation protocol, and secure comparison protocol, so as to achieve privacy-preserving in distributed ID3 algorithm. In 2007, Emekci et al. [5] implemented a secure addition computational protocol based on the secret sharing algorithm and extended the secure logarithmic computing protocol from two parties to multiple parties; thus realizing the multiparty participation of the privacy protection ID3 method. However, the complexity of the algorithm increases exponentially when the participant data are more numerous. In 2012, Lory et al. [6] used Chebyshev polynomial expansion to replace Taylor expansion in [1], thus further improving the computational efficiency of secure logarithmic computing protocols. However, their agreements still have limited efficiency in the implementation of privacy protection protocols.

Different from above, in 2003, Vaidya et al. [7, 8] designed a multiparty privacy-preserving ID3 algorithm of vertically distributed data sets. They vectorized all attribute value information by constructing constraint sets and then computed it by using the method of secure intersection protocol, thus designing privacy-preserving ID3 for vertically distributed data sets.

In 2007, Han and Ng [9] proposed a multiparty distributed privacy-preserving ID3 method based on arbitrary distributed data sets. Firstly, each participant's data set is vectorized, and then the attribute value information is computed by using security intersection protocol and so on. Then, the entropy value of each attribute is computed by using security logarithm computation protocol and so on. Thus, the ID3 decision tree classification method of privacy protection based on arbitrary distributed data set is obtained. However, with the increase of the number of participants, the computing volume of the client increases exponentially.

Li et al. [10] and Gao et al. [11] addressed the Naive Bayes Learning for aggregated arbitrary distributed databases.

PPDM with Computation Outsourcing. Cryptography-based privacy-preserving data mining has a lot of encryption and decryption operations in the computation process. Therefore, it is difficult for large-scale data processing. As a measure for solving resource-restricted problems, the outsourcing technique has been widely used in cloud computing applications, such as data sharing [12, 13], data storing [14, 15], data updating [16, 17], and social network analysis [18, 19]. In this context, we need to rely on security outsourcing technology to outsource computing or storage tasks of all participants to the cloud to process, thus greatly reducing the computing/storage load of each participant. In 2014, Liu et al. [3] adopted a new encryption scheme that supports both addition and multiplication over cipher texts. In this scheme, most of the computations are performed on the cloud, which reduces the computation workload of the data owner. However, the scheme is limited to a single party's data mining operation. Chen et al. [20] designed new algorithms for secure outsourcing of modular exponentiations. In 2015, Bost et al. [21] proposed the privacy-preserving hyperplane decision, Naive Bayesian, and decision tree classification algorithms, and through the semihonest model, secure two-party computation model to prove that the above scheme can satisfy the semantic security (Semantic Security); and the related protocol makes it possible to design an adaptive enhancement algorithm (Adaptive Boosting) combine to further enhance the accuracy of the algorithm; building a classifier can be used to construct the privacy protection of the library, the further development of the classification

algorithms for privacy-preserving technology in the future lays a solid foundation.

PPDM with Multiparticipant Data Storage and Computation Outsourcing. In 2013, Peter et al. [22] proposed a new solution for the outsourcing of multiparty computation. Such a technique can be used in our setting. But as the security analysis in the previous works, they can only achieve security in the semihonest model. In [23], a new protocol was proposed to achieve data mining for two parties. In [24], association rule mining was addressed in the malicious model. In [25], the privacy-preserving *KNN* classification was addressed. In [26], the deep learning task was addressed. Besides the above related work, several fundamental secure algorithms, such as dynamic homomorphic encryption [27, 28], authentication [29, 30], and light-weight multiparty computation [31], which have also been considered in the malicious model, have been proposed. However, to the best of our knowledge, no existing study has considered a method for outsourcing computation in the malicious model.

In this study, the secure outsourcing of ID3 data mining is considered in the malicious model for the cloud environment. We show how to solve the outsourcing problem for ID3 protocol over horizontally partitioned data.

2. Preliminaries

In this section, we present a brief overview of the preliminaries used in this paper, including the ID3 decision tree algorithm, Paillier's homomorphic encryption scheme, and the other related protocols.

2.1. Distribute ID3 Decision Tree Algorithm. The ID3 algorithm description is given as follows. It builds a decision tree in a top-down manner with the information of samples. Starting at the root, the best object classification will be obtained. The best prediction is computed with the *information gain*. The *information gain* of an attribute A_t is defined as

$$Gain\left(S, A_t\right) = Entropy\left(S\right) - \sum_{a_{tj} \in A_t} \frac{\left|S_{a_{tj}}\right|}{|S|} Entropy\left(S_{a_{tj}}\right) \tag{1}$$

Assume that there are 2 parties, $\mathcal{P} = \{P_i \mid i = 1, 2\}$, with 2 databases, $\mathcal{S} = \{S_i \mid i = 1, 2\}$. Each party P_i has one database S_i. All databases share the same general attribute (column) set $\mathcal{A} = \{A_t \mid t = 1, 2, .., p\}$ and each attribute A_t has several general discrete attribute values, denoted by $A_t = \{a_{tj} \mid j = 1, 2, \ldots, m_t\}$, and one class attribute $C = \{c_j \mid j = 1, 2, ..., m\}$.

Without considering privacy, each party P_i shares his own $|S_{a_{tj}c_j}|_i, |S_{a_{tj}}|_i$ and $|S_{c_j}|_i$ to all other parties. As a result, any party can calculate $Entropy(S)$ and $Entropy(S_{a_{tj}})$.

$$\begin{aligned} Entropy\left(S\right) &= \sum_{j=1}^{m} -\frac{\left|S_{c_j}\right|}{|S|} \log_2 \frac{\left|S_{c_j}\right|}{|S|} \\ &= \sum_{j=1}^{m} -\frac{\sum_{i=1}^{n} \left|S_{c_j}\right|_i}{|S|} \log_2 \frac{\sum_{i=1}^{n} \left|S_{c_j}\right|_i}{|S|} \end{aligned} \tag{2}$$

where S_{c_j} is the subset of S with tuples that have value c_j for class attribute C. $|S_{c_j}|_i$ equals the set of transactions with class attribute C set to c_j in database S_i.

Then the value of $Entropy(S_{a_j})$ can be calculated as

$$Entropy\left(S_{a_{tj}}\right) = \sum_{j=1}^{m_t} -\frac{\sum_{i=1}^{n} \left|S_{a_{tj}c_j}\right|_i}{\sum_{i=1}^{n} \left|S_{a_{tj}}\right|_i} \cdot \log_2 \frac{\sum_{i=1}^{n} \left|S_{a_{tj}c_j}\right|_i}{\sum_{i=1}^{n} \left|S_{a_{tj}}\right|_i} \tag{3}$$

where $S_{a_{tj}c_j}$ is the subset of S with tuples that have values a_j for attribute A_t and c_j for class attribute C.

Therefore, (3) can be easily computed by party P_i and parties P_j $(j \neq i)$ all of the values $|S_{a_{tj}}|_i$ and $|S_{a_{tj}c_j}|_i$ from its database. Each party P_j then sums these together with the values $|S_{a_{tj}}|_j$ and $|S_{a_{tj}c_j}|_j$ from its database and completes the computation.

Then each party can calculate $Gain(S, A_t)$ value at its own side.

2.2. Paillier's Homomorphic Encryption Scheme. Homomorphic encryption is a special type of encryption in which the result of applying a special algebraic operation to plain texts can be obtained by applying another algebraic operation (which may be different or the same) to the corresponding ciphertexts. Thus, even when the user does not know the plain texts, he/she can still obtain the results of applying that algebraic operation to the plain texts.

Let m_1 and m_2 be two plain texts with encryptions $E(m_1)$ and $E(m_2)$, respectively.

The Paillier encryption scheme [32] is described as follows:

$$E\left(m_1\right) \oplus E\left(m_2\right) = E\left(m_1 + m_2\right) \tag{4}$$

2.3. Li's Symmetric Homomorphic Encryption Scheme. The description of symmetric homomorphic encryption scheme proposed by Li et al. [33] is as follows.

(i) **KeyGen**():

$$\left(s, q, p\right) \longleftarrow KeyGen\left(\lambda\right) \tag{5}$$

KeyGen() is used to generate key for users as $SK = (s, q)$. p and q are primes with the condition that $p \gg q$. s is chosen from $\mathbb{Z}_N^*$.

(ii) $\mathbf{Enc_{sk}}$():

$$Enc_{sk}\left(m, d\right) = s^d \left(rq + m\right) \mod p. \tag{6}$$

d is a small positive integer, which is denoted as **ciphertext degree** in this paper.

1. The computation at Data owner:
Compute $\tau = Enc_{sk}(1)$ for the cloud.
2. The computation at cloud:
u, v are chosen such that

$$u \gg v$$
$$v \gg \max(\alpha, \beta)$$
$$\frac{(q-1)}{2} \gg \beta \times u + v$$
$$-\alpha \times u + v \gg -\frac{(q-1)}{2}$$

3. The cloud compute the following value for the data owner:
$\Phi = cu + \tau v \bmod p$, and sends Φ.
4. Data Owner computes the following values:
$\varphi = Dec_{sk}(SK, \Phi, d) = (m \times u + v) \bmod q$,
and compares φ with $(q-1)/2$.
If $\varphi < (q-1)/2, m \geq 0$. Otherwise, $m < 0$.

Algorithm 1: Secure outsourcing comparison (OSCP).

(iii) $\mathbf{Dec_{sk}}()$:

$$Dec_{sk}(c, d) = \left(c \times s^{-d} \bmod p\right) \quad \bmod q. \tag{7}$$

2.3.1. Properties of the Proposed Homomorphic Encryption

Homomorphic Multiplication. Let c_1, c_2 be the ciphertexts of two plaintexts m_1, m_2. Then we have $c_1 = s^{d_1}(r_1 q + m_1) \bmod p$ and $c_2 = s^{d_2}(r_2 q + m_2) \bmod p$ for some random ingredients r_1 and r_2. And we can obtain that

$$\begin{aligned}
&(c_1 \times c_2) \bmod p \\
&= s^{d_1}(r_1 q + m_1) \bmod p \times s^{d_2}(r_2 q + m_2) \bmod p \\
&= s^{d_1 + d_2}\left(\left(r_1 r 2 q + r_1 m_2 + r_2 m_1\right) q + m_1 \times m_2\right) \bmod p \\
&= Enc_{sk}(m_1 \times m_2, d_1 + d_2).
\end{aligned} \tag{8}$$

Homomorphic Addition

$$\begin{aligned}
(c_1 + c_2) \bmod p &= s^{d}(r_1 q + m_1) \bmod p \\
&\quad + s^{d}(r_2 q + m_2) \bmod p \\
&= s^{d}\left(\left(r_1 + r 2\right) q + m_1 + m_2\right) \bmod p \\
&= Enc_{sk}(m_1 + m_2, d)
\end{aligned} \tag{9}$$

Readers may refer to [18] for details on the scheme.

2.4. Garbling Scheme. A garbling scheme [34] consists of four algorithms, which is denoted by $G = (Gb, En, De, Ev)$. f can be transformed by Gb into (F, e, d). Note that F is the garbled circuit. The encoding and decoding information algorithms are denoted by e, d. The output of garbled Y can be encrypted and get the result $y = De(d, Y)$.

2.5. Noninteractive Commitment. A noninteractive commitment scheme [35] is also required in our paper, denoted by $(\mathrm{Com}_{crs}, \mathrm{Chk}_{crs})$. The distribution of $\mathrm{Com}_{crs}(x; r)$ is determined by the value of r as $\mathrm{Com}_{crs}(x)$.

2.6. Basic Cryptographic Subprotocols. In this section, we present a set of cryptographic subprotocols that will be used as subroutines when constructing the proposed protocol.

2.6.1. Outsourcing Secure Comparison Protocol (OSCP). The value of m is kept secure from the cloud and users. The value of $c = E(m \bmod q)$ is computed. SK is kept by the data owner (Algorithm 1).

2.6.2. Secure Equivalent Testing Protocol (SET). With two ciphertexts $\mathrm{c}_1 = \mathrm{Enc}_{\mathrm{sk}}(\mathrm{m}_1)$ and $\mathrm{c}_2 = \mathrm{Enc}_{\mathrm{sk}}(\mathrm{m}_2)$, **SET** is to compute f and decide if the plaintexts are identical ($\mathrm{m}_1 = \mathrm{m}_2$) (Algorithm 2).

2.6.3. Secure Multiplication Protocol (SMP). The algorithm is described as in Algorithm 3.

2.6.4. Secure Minimum out of 2 Numbers Protocol (SMIN2). The algorithm is described as in Algorithm 4.

2.7. Secure Circuit Protocol (SCP). We denote the three parties of the protocol by CSS_1, CSS_2, and CCS and their respective inputs by x_1, x_2, or x_3^*. Their goal is to securely compute the function $\mathrm{y} = \mathrm{f}(\mathrm{x}_1, \mathrm{x}_2, \mathrm{x}_3^*) = \mathrm{x}_1 \cdot \mathrm{x}_2 / \mathrm{x}_3^*$ [34] (Algorithm 5).

For simplicity, we assume that $|\mathrm{x}_\mathrm{i}| = |\mathrm{y}| = \mathrm{m}$. All communication between parties is via private point-to-point channels. Next, we assume that CSS_1 and CSS_2 can learn the same output y, while CCS can get the garbled values for the portion of the output wires corresponding to its own outputs only. CCS cannot get the output y with these garbled values. This protocol uses a garbling scheme, a four-tuple algorithm $\delta = (\mathrm{Gb}, \mathrm{En}, \mathrm{De}, \mathrm{Ev})$, as the underlying algorithm. Gb is a randomized garbling algorithm that transforms a function of a triple. En and De are encoding and decoding algorithms, respectively. Ev is an algorithm that produces a garbled output

Two ciphertext are computed by the cloud as $c_1 = Enc_{sk}(m_1)$ and $c_2 = Enc_{sk}(m_2)$.
1. The cloud computes $c_{00} = Enc_{sk}(m_{00}) = c_1 - c_2$ and $c_{01} = Enc_{sk}(m_{01}) = c_2 - c_1$.
2. Check if $m_{00} \geq 0$ or $m_{01} \geq 0$ and computes
if $m_{00} \geq 0 \wedge m_{01} < 0$, $f_0 = 1$, $f_1 = 0$. else if $m_{00} < 0 \wedge m_{01} \geq 0$, $f_0 = 0$, $f_1 = 1$.
3. The value of f is computed as follows:
$$f = f_0 \oplus f_1.$$
If $f = 0$, set $m_1 = m_2$.

Algorithm 2: Secure equivalent testing protocol (SET).

The values are computed $E_{pk}(x)$ and $E_{pk}(y)$; P keeps (pk, sk).
1. C:
(a) Choose $r_x, r_y \in Z_N$
(b) $x' \longleftarrow E_{pk}(x)E_{pk}(r_x)$, $y' \longleftarrow E_{pk}(y)E_{pk}(r_y)$
(c) computes and sends x', y' to P
2. P computes:
(a) $h_x \longleftarrow D_{sk}(x')$, $h_y \longleftarrow D_{sk}(y')$, $h \longleftarrow h_x h_y \bmod N$, $h' \longleftarrow E_{pk}(h)$
(b) The value of h' is sent to C
3. C computes:
(a) $s \longleftarrow h' E_{pk}(x)^{N-r_y}$, $s' \longleftarrow sE_{pk}(y)^{N-r_x}$
(b) $E_{pk}(xy) \longleftarrow s' E_{pk}(r_x r_y)^{N-1}$

Algorithm 3: $SM(E_{pk}(x), E_{pk}(y)) \longrightarrow E_{pk}(xy)$.

1. C:
(a) The function is chosen F
(b)
for $i = 1$ *to* λ **do**
$E_{pk}(u_i v_i) \longleftarrow SM(E_{pk}(u_i), E_{pk}(v_i))$.
if F: $u > v$ **then**
$W_i \longleftarrow E_{pk}(u_i)E_{pk}(u_i v_i)^{N-1}$,
end
else
$W_i \longleftarrow E_{pk}(v_i)E_{pk}(u_i v_i)^{N-1}$
end
$G_i \longleftarrow E_{pk}(u_i \oplus v_i)$, $H_i \longleftarrow H_{i-1}^{r_i} G_i$; $r_i \in_R$
Z_N and $H_0 = E(0)$
$\Phi_i \longleftarrow E_{pk}(-1)H_i$,
$L_i \longleftarrow W_i \Phi_i^{r_i'}$; $r_i' \in Z_N$
end
(c) Sends L to P
2. P:
(a) $M_i \longleftarrow D(L_i)$, for $1 \leq i \leq \lambda$
(b)
if $\exists j$ *such that* $M_j = 1$ **then**
$\alpha \longleftarrow 1$ (which means $u > v$)
end
else
$\alpha \longleftarrow 0$ (which means $u < v$)
end

Algorithm 4: Secure Minimum out of 2 Numbers Protocol (SMIN2).

Input: CSS_1 has x_1, CSS_2 has x_2
Output: $(x_1 + x_2)/(a_1 + a_2)$
1. CCS:
(a) Randomly selects crs for commitment, and sends crs to CSS_1 and CSS_2.
(b) Generates the random number λ and share it secretly as $\lambda = \lambda_1 \oplus \lambda_2$, sends $lambda_1$ to CSS_1, and sends $lambda_2$ to CSS_2.
2. CSS_1: Select seed $r \longleftarrow 0, 1^k$ for pseudo random function PRF and send r to CSS_2.
3. CSS_1 and CSS_2: (a) Generate corresponding circuit $Gb(1^\kappa, f) \longrightarrow (F, e, d)$ based on function $f = (x_1 + x_2) * \lambda$. ($b$) Random selection of $b_1, b_2 \longleftarrow \{0,1\}^{4m}$ and generate the following commitments for all $j \in [4m]$ and $t \in \{0,1\}$:
$(C^t_{1,j}, \sigma^t_{1,j}) \longleftarrow \mathrm{Com}_{crs}(e[j, b_1[j] \oplus t])$.
$(C^t_{2,j}, \sigma^t_{2,j}) \longleftarrow \mathrm{Com}_{crs}(e[j, b_2[j] \oplus t])$.
(c) CSS_1 and CSS_2 send the following information to CCS:
$(b_1[2m + 1 \cdots 4m], F, \{C^t_{1,j}\}_{j,t})$.
$(b_2[2m + 1 \cdots 4m], F, \{C^t_{2,j}\}_{j,t})$.
4. CCS: Abort if CSS_1 and CSS_2 report different values for these items.
5. CSS_1 and CSS_2:
(a) CSS_1 sends decommitment
$\sigma^{x_1[j] \oplus b_1[j]}_{1,j}$, $\sigma^{\lambda_1[j] \oplus b_1[2m+j]}_{1,2m+j}$, $\sigma^{x_1[j] \oplus b_2[j]}_{2,j}$ and $\sigma^{\lambda_1[j] \oplus b_2[2m+j]}_{2,2m+j}$ to CCS
(b) CSS_2 sends decommitment $\sigma^{x_2[j] \oplus b_1[m+j]}_{1,m+j}$, $\sigma^{\lambda_2[j] \oplus b_1[3m+j]}_{1,3m+j}$, $\sigma^{x_2[j] \oplus b_2[m+j]}_{2,m+j}$ and $\sigma^{\lambda_2[j] \oplus b_2[3m+j]}_{2,3m+j}$ to CCS.
6. CCS: (a) For $j \in [4m]$, compute $X[j] = \mathrm{Chk}_{crs}(C^{o[j]}_{1,j}, \sigma^{o[j]}_{1,j})$, $X'[j] = \mathrm{Chk}_{crs}(C^{o[j]}_{2,j}, \sigma^{o[j]}_{2,j})$, for the appropriate $o[j]$. If any call to Chk returns $\perp$, then abort. Similarly, CCS knows the values $b_1[2m + 1 \cdots 4m]$ and $b_2[2m + 1 \cdots 4m]$, and aborts if CSS_1 or CSS_2 can not open the corresponding commitments of λ_1 and λ_2: $C^{\lambda_1[j] \oplus b_1[2m+j]}_{1,2m+j}$, $C^{\lambda_2[j] \oplus b_1[3m+j]}_{1,3m+j}$, $C^{\lambda_1[j] \oplus b_2[2m+j]}_{2,2m+j}$ and $C^{\lambda_2[j] \oplus b_2[3m+j]}_{2,3m+j}$.
(b) Run $Y \longleftarrow \mathsf{Ev}(F, X)$ and $Y' \longleftarrow \mathsf{Ev}(F, X')$, then broadcasts Y and Y' to CSS_1 and CSS_2.
7. CSS_1 and CSS_2: Compute $(x_1 + x_2)/(a_1 + a_2) = De(d, Y)/De(d, Y')$.

Algorithm 5: Secure circuit protocol (SCP).

for a garbled input and garbled circuit. Further, Chk is an algorithm that can verify commitments.

3. Outsourcing Privacy-Preserving ID3 Decision Tree Algorithm in Malicious Model

In this section, we present our secure outsourcing ID3 decision tree in cloud computing using the homomorphic encryption scheme and subprotocols proposed in Section 2 as building blocks.

3.1. Main Concept. The aim is to privately compute ID3 over encrypted databases, and the key is to find privately the attribute A for which Gain is maximum. From the above description, the key value which needs to be calculated with other parties is Entropy(S_{a_i}).

Since all the data was encrypted and sent to the cloud, the cloud server can count the number of $|S(c_k)|_t$, $|S|_{it}$ using the **SET** protocol described in Section 2. Now, (3) can be executed as $(x_1 + x_2)/(a_1 + a_2)\log_2(x_1 + x_2)/(a_1 + a_2)$, and the calculation of the logarithmic operation can be performed in CSS. The value to be calculated is the value of $c_1 = (x_1 + x_2)/(a_1 + a_2)$, which can be easily determined using our **SCP** protocol as explained in Section 2. Then, all the parties can calculate the value of Entropy(S) independently.

3.2. System Model. The system model is shown in Figure 1, which includes two data owners and cloud servers (cloud storage server {CSS_1, CSS_2}, and cloud computing server CCS). Each data owner owns a private data set that is encrypted and outsourced to cloud server storage. Data owners can request cloud server to process ID3 data on encrypted data. At the same time, CSS and CCS servers participate in supporting the outsourcing privacy protection ID3 data mining algorithm steps; after the implementation of the algorithm, the final results are sent to the data owner. Assuming that the data owner and the CSS server are semihonest participants, CCS is a malicious participant.

3.3. Details of the Proposed Algorithm. Our securely outsourcing ID3 decision tree (SOID3) algorithm is detailed as follows:

(1) P_1 and P_2 run KeyGen(λ) to generate the secret key SK_i, i = 1, 2 and a public parameter p of Li's homomorphic encryption scheme. Further, each party shares p with the other party and the cloud but shares SK_i only with itself.

(2) Each party uses its key SK_i to encrypt every attribute value of its database, and then outsources the encrypted database to the CSS (CSS_1 and CSS_2).

(3) The CSS_1 and CSS_2 use the **SET** protocol to calculate the value of $|S_{a_j}|_i$ and $|S_{a_j}(c_k)|_i$ for each attribute with each party P_i.

(4) Each party generates its Paillier public and private keys (pk_i, sk_i), i = 1, 2, and sends the public keys to the CSS_1 and CSS_2.

(5) CCS, CSS_1, and CSS_2 jointly use the **SCP** protocol to compute $(x_1 + x_2)/(a_1 + a_2)$. Here, CSS_1 has (x_1, a_1), and CSS_2 has (x_2, a_2).

(6) Each party decrypts the received information, calculates it with the logarithmic operation of $(x_1 + x_2/a_1 +$

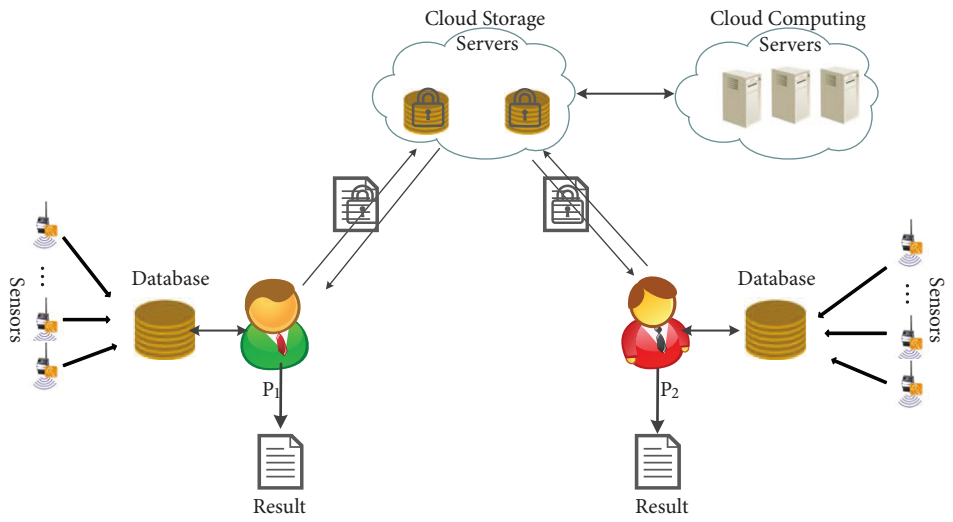

FIGURE 1: System model under consideration.

$a_2)\log_2(x_1 + x_2/a_1 + a_2)$, and then encrypts it with its public key. Then, it sends it back to the cloud.

(7) After getting the result, CSS_1 and CSS_2 use the **SMIN2** protocol to select the ciphertext data with the minimum value and then further select the attribute label with the maximum information gain and return it to each participant.

(8) The participants divide the data sets and build tree nodes. Then, go to Step (3) until termination.

4. Security Analysis

In this section, we prove that the secure outsourcing ID3 decision tree (SOID3) algorithm can offer protection against the malicious cloud server.

Theorem 1. *The SOID3 algorithm can achieve privacy for each party and the semihonest cloud storage server.*

Proof. We mainly consider the security model under the noncollusive semihonest model and the semihonest cloud server. Suppose there are two parties, P_1 and P_2, and cloud storage server CSS.

Let $P = (P_1, P_2, CSS)$ be the participants of all protocols. Consider three types of attackers (A_{P_1}, A_{P_2}, and A_{CSS}) that can invade P_1, P_2, and CSS. In the real model, P_1 and P_2 have data sets D_x and D_y, respectively, and CSS has encrypted data sets $\mathrm{Enc}(D_x)$ and $\mathrm{Enc}(D_y)$. Make $H \subset P$ a collection of honest participants. For all $P_i \epsilon H$, out_{P_i} indicates the output of P_i. If P_i is invaded, out_{P_i} represents all views of participant P_i in running protocol Π.

For each $P^* \in P$, the attacker $A = (A_{P_1}, A_{P_2}, A_{CSS})$ view in the runtime protocol Π can be defined as

$$\mathrm{REAL}^{P^*}_{\Pi,A,H}\left(D_x, D_y\right) = \mathrm{out}_{P_i} : P_i \epsilon H \cup \mathrm{out}_{P^*} \tag{10}$$

In the ideal model, there exists an ideal model F for function f, and all participants can interact with the model F. That is, Challenger DP_a and participant P_i can send data x and y to F. If D_x or D_y is $\perp$, F returns $\perp$. Finally, F can return $f(D_x, D_y)$ to challenger DP_a. As mentioned earlier, $H \subset P$ is a collection of honest participants. For each participant $P_i \epsilon H$ in the collection, return the out_{P_i} as F output to P_i. If P_i is intruded on by a semihonest attacker, out_{P_i} is still consistent with the output of P_i in previous realistic models.

For all $P^* \in P$, in the ideal model, in the presence of independent simulators $\mathrm{Sim} = (\mathrm{Sim}_{P_1}, \mathrm{Sim}_{P_2}, \mathrm{Sim}_{CSS})$, the P^* view is

$$\mathrm{IDEAL}^{P^*}_{F,\mathrm{Sim},H}\left(D_x, D_y\right) = \mathrm{out}_{P_i} : P_i \epsilon H \cup \mathrm{out}_{P^*} \tag{11}$$

Therefore, it is considered that the protocol Π is secure in the presence of noncolluded semitruthful attackers.

Definition 2. Let f be a deterministic functionality among parties in P. Let $H \subset P$ be the subset of honest parties in P. We say that Π securely realizes f if there exists a set $\mathrm{Sim} = \{\mathrm{Sim}_{P_1}, \mathrm{Sim}_{P_2}, \mathrm{Sim}_S\}$ of PPT transformations (where $\mathrm{Sim}_{D_a} = \mathrm{Sim}_{P_1}(A_{P_1})$ and so on) such that for all semihonest PPT adversaries $A = \{A_{P_1}, A_{P_2}, A_S\}$, for all inputs D_x, D_y and auxiliary inputs z, and for all parties $P \in P$ the following holds:

$$\begin{aligned}&\mathrm{REAL}^{P^*}_{\Pi,A,H,z}\left(\lambda, x, y\right)_{\lambda \in N} \cong \\ &\mathrm{IDEAL}^{P^*}_{F,\mathrm{Sim},H,z}\left(\lambda, x, y\right)_{\lambda \in N},\end{aligned} \tag{12}$$

where $\cong$ denotes computational indistinguishability. □

Theorem 2. *The SOID3 algorithm is secure with the semihonest cloud storage server and the malicious cloud computing server.*

Proof. First consider the case where CSS_1 or CSS_2 is corrupted. It is necessary to prove that, in the **SCP** protocol, the ideal model and the realistic model are not distinguishable. That is, in the following interactions, it is impossible to distinguish between the various types of interaction information and outputs of the participants in the ideal model and the real model.

(1) In the real model, assume that there is an emulator that can simulate various behaviors of a semihonest participant CSS_1 (or CSS_2), and receive inputs (x_1, a_2) and (x_2, a_2) from the execution environment of the protocol. At the same time, the simulator can simulate the function F_f, which sends all inputs (x_1, a_1) and (x_2, a_2) to the simulated F_f. Since the simulator does not do anything computed by F_f, there is no difference between the real F_f and the simulated F_f from the execution environment point of view.

(2) Because in Step 2, CSS_1 and CSS_2 uniformly select the seed r of Pseudo-Random Function (PRF), the PRF security shows that the real model in Step 2 is indistinguishable from the ideal model.

(3) In Step 3, we modify the simulator, which knows in advance what promises will be opened when the simulator generates commitment C. First, the simulator selects the random numbers o_1, o_2 that can be marked which promises to be opened and calculates the values of $b_1 = o_1 \oplus x_1 \parallel x_2$, $b_2 = o_2 \oplus a_1 \parallel a_2$. At this point, the simulator has obtained the values of x_1, x_2, and a_1, a_2. Then, the simulator can submit the markings that promise not to be opened. In this process, due to the concealment of commitment, the realistic model and the ideal model are equally indistinguishable.

(4) In Step 6a, the simulated CSS_1 and CSS_2 stop executing when $De(D, \widetilde{Y}) = \perp$. Change the emulator to make $\widetilde{Y} \neq Ev(F, X)$. By obfuscating the authenticity of the circuit, CCS has only negligible probability to obtain $\widetilde{Y} \neq Ev(F, X)$ in $De(d, \widetilde{Y}) = \perp$. Therefore, in this step, the realistic model and the ideal model are equally indistinguishable.

(5) In Step 6b, the correctness of the obfuscation circuit guarantees that both CSS_1 and CSS_2 of the analog can be output. Therefore, if there is no pause in 6a, we can modify the simulator to an analog obfuscation circuit that generates (F, X, d). We can simulate the output of CSS_1 and CSS_2 by simulating the instructions of F_f. According to the security of the confusing circuit, the real model is also indistinguishable from the ideal model in this step.

Therefore, in this protocol, the execution environment can not distinguish between the realistic model and the ideal model. And the protocol is secure when CCS is a malicious participant. □

5. Performance Analysis

In this paper, we consider that CSS has a strong calculation ability and we ignore its computation time. Each data owner does not need to store the ciphertext but can just use the public key to encrypt the message and the private key to decrypt the ciphertext.

In each iteration, first, each data owner will execute the **SBD** protocol and **SMIN2** protocol with the cloud. There are two interactions in the **SBD** protocol and 2k interactions in the **SMIN2** protocol. Then, CSS_1, CSS_2, and CCS will execute 6 interactions in the **SCP** protocol. Finally, each data owner will execute 1 interaction when it goes to the new iteration. We assume that t is the iteration time, so the communication traffic is at most $O(k * t)$.

In this paper, a secure average computing protocol based on **SCP** is implemented. The server selected in the experimental environment is CPU: Intel (R) Xeon (R) CPU E5-2620 v3@2.40GHz∗2, memory: 32G, operating system Ubuntu 16.04.4 LTS version. In the experiment, AES-128 is chosen as the basic encryption method of the confusion circuit, and the open source code of JustGarble is changed, and the commitment protocol is implemented based on SHA-256. Finally, the average values obtained from experiments are as follows.

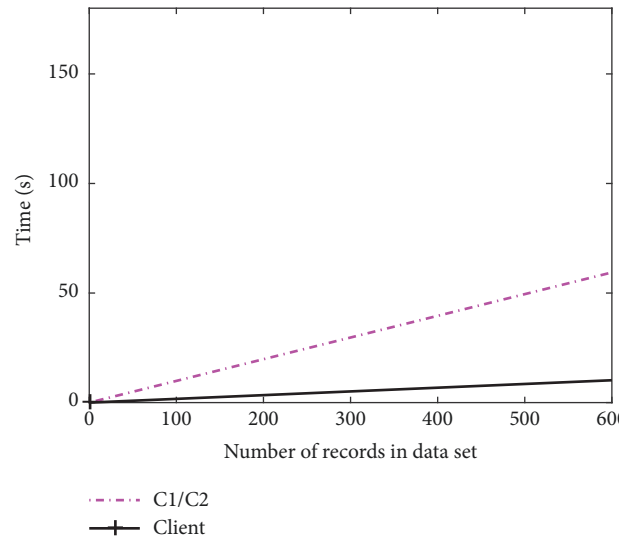

Figure 2: Performance measurements for our SOID3 with 2 participants.

In our secure outsourcing ID3 decision tree (SOID3) algorithm experiment, two participants were tested with different numbers of records. The experimental results are shown in Figure 2.

From Figure 2, since the client is only responsible for encrypting uploaded data, the time consumption is very low. In the cloud, CCS and CSS servers need to run **SCP** protocol, resulting in a lot of time consumption (Table 1). The main reason is that a large number of bit commitment processing is needed in the obfuscation circuit, and the performance improvement will be focused on this issue in the follow-up work.

6. Conclusion

In this paper, we proposed a secure outsourcing ID3 decision tree algorithm for two parties of the malicious model. Our algorithm can preserve the privacy of the users' data as well as that of the data mining scheme for the cloud servers. The parties can get only the result trees and have no knowledge about the data mining scheme. Moreover, the cloud servers cannot get any private information about the parties. In summary, our protocol offers protection against malicious cloud servers.

In the future, we intend to extend our algorithm to vertical and arbitrary partitioning in the malicious model. In addition, we plan to extend our algorithm to a general multiparty privacy-preserving framework suitable for other useful schemes, such as random decision tree, Bayes, SVM,

Table 1: Time cost of the SCP protocol.

AND	XOR	OR	Input size (bits)	Output size (bits)	CSS_1/CSS_2 (ms)	*CCS* (ms)
5090	4034	2016	129	65	26	47

and other data mining methods, and can be extended for use in the wireless sensor-networks [36, 37].

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported by National Key Research and Development Program of China (no. 2017YFB0803002), Basic Research Project of Shenzhen, China (no. JCYJ20160318094015947), National Natural Science Foundation of China (nos. 61872109 and 61771222), and Key Technology Program of Shenzhen, China (no. JSGG20160427185010977).

References

[1] Y. Lindell and B. Pinkas, "Privacy preserving data mining," *Journal of Cryptology*, vol. 15, no. 3, pp. 177–206, 2002.

[2] R. Agrawal and R. Srikant, "Privacy-preserving data mining," *ACM SIGMOD Record*, vol. 29, no. 2, pp. 439–450, 2000.

[3] D. Liu, E. Bertino, and X. Yi, "Privacy of outsourced *k*-means clustering," in *Proceedings of the 9th ACM Symposium on Information, Computer and Communications Security, ASIA CCS 2014*, pp. 123–133, June 2014.

[4] P. Li, J. Li, Z. Huang, C.-Z. Gao, W.-B. Chen, and K. Chen, "Privacy-preserving outsourced classification in cloud computing," *Cluster Computing*, pp. 1–10, 2017.

[5] F. Emekci, O. D. Sahin, D. Agrawal, and A. El Abbadi, "Privacy preserving decision tree learning over multiple parties," *Data & Knowledge Engineering*, vol. 63, no. 2, pp. 348–361, 2007.

[6] P. Lory, "Enhancing the Efficiency in Privacy Preserving Learning of Decision Trees in Partitioned Databases," in *Privacy in Statistical Databases*, vol. 7556 of *Lecture Notes in Computer Science*, pp. 322–335, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[7] J. Vaidya and C. Clifton, "Privacy-preserving *K*-means clustering over vertically partitioned data," in *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '03)*, pp. 206–215, Washington, DC , USA, August 2003.

[8] J. Zhan, S. Matwin, and L. W. Chang, "Privacy-Preserving Naive Bayesian Classification over Horizontally Partitioned Data," in *Data Mining: Foundations and Practice*, vol. 118 of *Studies in Computational Intelligence*, pp. 529–538, Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[9] S. Han and W. K. Ng, "Multi-party privacy-preserving decision trees for arbitrarily partitioned data," *International Journal of Intelligent Control and Systems*, vol. 12, no. 4, 2007.

[10] T. Li, J. Li, Z. Liu, P. Li, and C. Jia, "Differentially private Naive Bayes learning over multiple data sources," *Information Sciences*, vol. 444, pp. 89–104, 2018.

[11] C. Gao, Q. Cheng, P. He, W. Susilo, and J. Li, "Privacy-preserving Naive Bayes classifiers secure against the substitution-then-comparison attack," *Information Sciences*, vol. 444, pp. 72–88, 2018.

[12] J. Shen, T. Zhou, X. Chen, J. Li, and W. Susilo, "Anonymous and Traceable Group Data Sharing in Cloud Computing," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 4, pp. 912–925, 2018.

[13] Z. Cai, H. Yan, P. Li, Z.-A. Huang, and C. Gao, "Towards secure and flexible EHR sharing in mobile health cloud under static assumptions," *Cluster Computing*, vol. 20, no. 3, pp. 2415–2422, 2017.

[14] J. Li, Y. K. Li, X. Chen, P. P. C. Lee, and W. Lou, "A Hybrid Cloud Approach for Secure Authorized Deduplication," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 5, pp. 1206–1216, 2015.

[15] Z. Liu, Y. Huang, J. Li, X. Cheng, and C. Shen, "DivORAM: Towards a practical oblivious RAM with variable block size," *Information Sciences*, vol. 447, pp. 1–11, 2018.

[16] X. Chen, J. Li, J. Weng, J. Ma, and W. Lou, "Verifiable computation over large database with incremental updates," *Institute of Electrical and Electronics Engineers. Transactions on Computers*, vol. 65, no. 10, pp. 3184–3195, 2016.

[17] J. Shen, C. Wang, T. Li, X. Chen, X. Huang, and Z.-H. Zhan, "Secure data uploading scheme for a smart home system," *Information Sciences*, vol. 453, pp. 186–197, 2018.

[18] S. Chen, G. Wang, G. Yan, and D. Xie, "Multi-dimensional fuzzy trust evaluation for mobile social networks based on dynamic community structures," *Concurrency and Computation: Practice and Experience*, vol. 29, no. 7, p. e3901, 2017.

[19] E. Luo, Q. Liu, J. H. Abawajy, and G. Wang, "Privacy-preserving multi-hop profile-matching protocol for proximity mobile social networks," *Future Generation Computer Systems*, vol. 68, pp. 222–233, 2017.

[20] X. F. Chen, J. Li, J. Ma, Q. Tang, and W. Lou, "New algorithms for secure outsourcing of modular exponentiations," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 9, pp. 2386–2396, 2014.

[21] R. Bost, R. A. Popa, S. Tu, and S. Goldwasser, "Machine Learning Classification over Encrypted Data," in *Proceedings of the Network and Distributed System Security Symposium*, San Diego, CA.

[22] A. Peter, E. Tews, and S. Katzenbeisser, "Efficiently outsourcing multiparty computation under multiple keys," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 12, pp. 2046–2058, 2013.

[23] G. Jagannathan, K. Pillaipakkamnatt, and R. N. Wright, "A Practical Differentially Private Random Decision Tree Classifier," in

Proceedings of the 2009 IEEE International Conference on Data Mining Workshops (ICDMW), pp. 114–121, Miami, FL, USA, December 2009.

[24] B. Gilburd, A. Schuster, and R. Wolff, "Privacy-preserving data mining on data grids in the presence of malicious participants," in *Proceedings of the Proceedings. 13th IEEE International Symposium on High performance Distributed Computing, 2004.*, pp. 225–234, Honolulu, HI, USA.

[25] D. Shah and S. Zhong, "Two methods for privacy preserving data mining with malicious participants," *Information Sciences*, vol. 177, no. 23, pp. 5468–5483, 2007.

[26] P. Li, J. Li, Z. Huang et al., "Multi-key privacy-preserving deep learning in cloud computing," *Future Generation Computer Systems*, vol. 74, pp. 76–85, 2017.

[27] Q. Lin, H. Yan, Z. Huang, W. Chen, J. Shen, and Y. Tang, "An ID-based linearly homomorphic signature scheme and its application in blockchain," *IEEE Access*, vol. PP, no. 99, pp. 1-1, 2018.

[28] J. Xu, L. Wei, Y. Zhang, A. Wang, F. Zhou, and C. Gao, "Dynamic Fully Homomorphic encryption-based Merkle Tree for lightweight streaming authenticated data structures," *Journal of Network and Computer Applications*, vol. 107, pp. 113–124, 2018.

[29] J. Shen, Z. Gui, S. Ji, J. Shen, H. Tan, and Y. Tang, "Cloud-aided lightweight certificateless authentication protocol with anonymity for wireless body area networks," *Journal of Network and Computer Applications*, vol. 106, pp. 117–123, 2018.

[30] X. Zhang, Y. Tan, C. Liang, Y. Li, and J. Li, "A Covert Channel Over VoLTE via Adjusting Silence Periods," *IEEE Access*, vol. 6, pp. 9292–9302, 2018.

[31] Y. Wang, T. Li, H. Qin et al., "A brief survey on secure multi-party computing in the presence of rational parties," *Journal of Ambient Intelligence and Humanized Computing*, vol. 6, no. 6, pp. 807–824, 2015.

[32] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Advances in Cryptology—EUROCRYPT '99*, vol. 1592, pp. 223–238, Springer, 1999.

[33] L. Li, R. Lu, K.-K. R. Choo, A. Datta, and J. Shao, "Privacy-Preserving-Outsourced Association Rule Mining on Vertically Partitioned Databases," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 8, pp. 1547–1861, 2016.

[34] P. Mohassel, M. Rosulek, and Y. Zhang, "Fast and Secure Three-party Computation," in *Proceedings of the the 22nd ACM SIGSAC Conference*, pp. 591–602, Denver, Colorado, USA, October 2015.

[35] M. Naor, "Bit commitment using pseudorandomness," *Journal of Cryptology*, vol. 4, no. 2, pp. 151–158, 1991.

[36] H. Cheng, Z. Su, N. Xiong, and Y. Xiao, "Energy-efficient node scheduling algorithms for wireless sensor networks using Markov Random Field model," *Information Sciences*, vol. 329, pp. 461–477, 2016.

[37] H. Cheng, N. Xiong, A. V. Vasilakos, L. Tianruo Yang, G. Chen, and X. Zhuang, "Nodes organization for channel assignment with topology preservation in multi-radio wireless mesh networks," *Ad Hoc Networks*, vol. 10, no. 5, pp. 760–773, 2012.