Secure Privacy-Preserving Association Rule Mining With Single Cloud Server

ZHIYONG HONG$^{1,2}$, ZHILI ZHANG$^{3}$, PU DUAN$^{4}$, BENYU ZHANG$^{4}$, BAOCANG WANG$^{5}$, WEN GAO$^{6}$, AND ZHEN ZHAO$^{5,7}$

$^{1}$Facility of Intelligence Manufacture, Wuyi University, Jiangmen 529020, China
$^{2}$Yue-Gang-Ao Industrial Big Data Collaborative Innovation Center, Wuyi University, Jiangmen 529020, China
$^{3}$School of Information Engineering, Xuchang University, Xuchang 461000, China
$^{4}$Ant Group, Hangzhou 310000, China
$^{5}$State Key Laboratory of Integrated Service Networks, Xidian University, Xi’an 710071, China
$^{6}$School of Cyberspace Security, Xi’an University of Posts and Telecommunications, Xi’an 710121, China
$^{7}$Guangxi Key Laboratory of Cryptography and Information Security, Guilin 541004, China

Corresponding author: Zhili Zhang (zljzhangxc@163.com)

This work was supported in part by the Guangxi Key Laboratory of Cryptography and Information Security under Grant GCIS202106; in part by the National Natural Science Foundation of China under Grant U19B2021, Grant 61902303, and Grant 62002288; in part by the Key Research and Development Program of Shaanxi under Grant 2020ZDLGY08-04; in part by the National Cryptography Development Fund under Grant MMJ20180219; in part by the Natural Science Basic Research Program of Shaanxi under Grant 2020JQ-832; in part by the Key Technologies Research and Development Program of He’nan Province under Grant 212102210084 and Grant 192102210295; and in part by the Technicians Troop Construction Projects of Henan Province.

ABSTRACT
To preserve the privacy of data uploaded on the cloud, it is widely accepted to encrypt the data before uploading it. This leads to the challenge of data analysis, especially association rule mining while protecting data privacy. As one of the solutions, homomorphic encryption is presented allowing encrypted data processing without decryption. In particular, the twin-cloud structure is frequently applied in the privacy-preserving association rule mining schemes based on asymmetric homomorphic encryption, which contradicts the reality that most of the practical applications applied the single cloud server. However, the existing related single cloud server schemes suffer from privacy leakage problems. To fill this gap in the literature, in this paper, we first present a universal secure multiplication protocol with the garbled circuit and additive homomorphic encryption. Based on this multiplication protocol, we construct the inner product protocol, comparison protocol, frequent itemset protocol, and the final association rule mining protocol that is secure against privacy leakage. Finally, we give the theoretical security analysis of the proposed protocols and show its performance analysis.

INDEX TERMS
Privacy-preserving association rule mining, single cloud server, garbled circuit, homomorphic encryption.

I. INTRODUCTION
In the information era, people’s life is transferred into data and stored on the network, such as shopping preference, traffic line, and hobbies. This promotes the industry field to focus on data mining techniques for the great economic value that the mining results will bring. In particular, association rule mining is one of the directions being concerned, which is to search out the related dependencies between items in the given dataset. For example, it can analyze the customers’ shopping baskets, and the supermarket can accordingly adjust the placement of commodity shelves to implicitly obtain more benefits. Association rule mining has been widely applied in many fields, such as business [1], health analysis [2], and website analysis [3]. Along with the increasing amount of data needed to be analyzed, the cloud that can support strong computing and storage power is widely accepted and applied [4]. As a natural extension, outsourced association rule mining allows the cloud server to perform the data mining work on the uploaded database. However, the cloud platform which is run by the not fully-trusted company in the practical applications may lead to the privacy leakage problem or even economic loss. Various solutions are, therefore, proposed to further realize privacy-preserving against

The associate editor coordinating the review of this manuscript and approving it for publication was Mansoor Ahmed$^D$. 

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/
the malicious cloud server in the outsourced association rule mining, such as data perturbation [5], [6], anonymization [7], [8], differential privacy [9], [10], and cryptography-based methods [11], [12].

Data perturbation [5], [6] adds noise to the data to create perturbed data, where the noise is large enough to mask the record’s specific attribute value. When performing computing tasks, the perturbed data is iteratively reconstructed to obtain an approximate original data distribution. It is easy to lose the original data and decrease the effectiveness of the results. As a classic technique of anonymization [7], \( k \)-anonymity [8] helps the data to be invisible by constructing the equivalence class which contains at least \( k \) pieces of indistinguishable data. This method may cause privacy disclosure since the sensitive attributes in the equivalence class are not constrained. Anonymization relies too much on background knowledge assumptions, and association attacks, link attacks, and multi-source attacks may break its security. Differential privacy [9], [10] allows the cloud server to analyze the entirety of a data set without revealing individual privacy, which does not care about the background knowledge possessed by the adversary. Even if the adversary has obtained the information of all the records except a certain record, it cannot get the privacy of this record. However, when the dimension of the data is high, the noise of the differential privacy processing result will be very large, resulting in the low usability of the result.

Compared with the abovementioned techniques, we have cryptography-based methods that can usually obtain an accurate and effective mining result at a high-security level, which means encrypting and then uploading the data to the cloud to further protect data privacy against curious or malicious cloud server. The universal system model in the outsourced association rule mining contains the cloud server, the data owner, and the rule miner. The existing related schemes always design their system model based on non-collusion cloud servers, where cloud servers are responsible for the computations generated during the mining process and the data owner and miner can be offline. Such schemes guarantee a high level of privacy security for each participant in the entire mining process, while they are based on a strong assumption that the two cloud servers cannot collude with each other. In addition, one of the cloud servers in the twin-cloud setting schemes usually owns the decrypting ability, which leads to hidden security risks. The single-cloud-server setting [13], [14] can avoid the aforementioned trouble, while most of the existing schemes suffer from the privacy leakage problem. To fill this gap in the literature, we propose a secure privacy-preserving association rule mining scheme based on a single-cloud-server setting.

A. RELATED WORK

As a classical topic, privacy preservation of data has been put a lot of attention on [15]–[18]. In particular, various research has been conducted on privacy-preserving outsourced association rule mining to further protect data privacy against the curious or malicious cloud server while correctly performing the association rule mining process in cloud computing. Vaidya and Kantarcioglu [19], [20] first considered the privacy-preserving requirements in data mining and proposed the idea of defining and measuring privacy. They [19], [20] presented the privacy-preserving association rule mining schemes in vertically and horizontally partitioned databases, respectively.

Data Perturbation and Anonymization: To resist background attacks in the mining process, Wong et al. [21] proposed one-to-many item mapping, which can randomly convert transactions. The core idea of the scheme is to add noise to the transaction database. In 2010, Tai et al. [5] proposed the conception of \( k \) support anonymity in which \( k \)-1 fake items with similar support will be added to protect the privacy of given items. Giannotti et al. [5] improved the security of this primitive to \( k \) privacy, where a converted item set is indistinguishable from at least \( k \)-1 other item sets by adding fake transactions to the database. This scheme trades off the privacy-preserving security against computation efficiency.

Outsourced Privacy-Preserving Association Rule Mining: Lai et al. [22] proposed the first outsourced privacy-preserving association rule mining scheme with semantic security based on predicate encryption, while its efficiency is still undesirable in practice. Yi et al. [23] proposed a privacy-preserving association rule mining protocol based on Elgamal homomorphic encryption algorithm. In their protocol, a data miner can outsource all computations to cloud servers, and data owners only perform some encryption operations. However, this protocol requires \( n \) servers to perform distributed homomorphic computations, which will bring a lot of communication between cloud servers.

Outsourced Privacy-Preserving Association Rule Mining With Twin-Cloud Setting: In 2017, Qiu et al. [24] proposed a twin-cloud structure, which is currently a commonly used system model framework. They constructed a scheme for frequent itemset mining with three privacy levels using Paillier [25] and [26] homomorphic encryption. But none of the three privacy levels can guarantee the security of mining results. Ma et al. [27] improved the mining efficiency of this work [24] and designed a block algorithm for frequent itemset mining, where encrypted transactions can be divided into multiple blocks based on the sparse attributes of the transaction database. Kim et al. [28] proposed a privacy-preserving association rule mining on the twin-cloud structure without adding fake transactions. Besides, Liu et al. [29] proposed a privacy-preserving association rule mining scheme for outsourced cloud data in a multi-key environment based on the BCP homomorphic encryption [30], [31]. Pang and Wang [32] presented a privacy-preserving association rule mining scheme in a multikey environment under the twin-cloud setting.

Outsourced Privacy-Preserving Association Rule Mining With Single Cloud-Server Setting: In 2016, Li et al. [13] constructed one symmetric fully homomorphic encryption...
algorithm and designed a privacy-preserving association rule mining scheme on a vertical partition database based on this algorithm. The system model of this solution uses a single cloud server structure. Although the scheme is very efficient, the original data may be leaked. Wu et al. [14] proposed an association rule mining scheme called SecEDMO under the single cloud server structure. They constructed a secure symmetric homomorphic encryption algorithm. But if there is a collision between two data owners, it may lead to the disclosure of sensitive information of others.

B. OUR CONTRIBUTION

To remove the strong assumption that two cloud servers will not collude with each other in the twin-cloud setting and fill the gap that the single-cloud setting always leads to privacy disclosure, we propose a secure privacy-preserving association rule mining scheme based on a single-cloud-server setting.

Based on the garbled circuit and additive homomorphic encryption, we construct a universal secure multiplication protocol that can securely compute the product of the secret bits between the data owner and the data miner with the single-cloud setting. To further facilitate understanding, we instantiate it into a concrete construction by applying the Paillier cryptosystem as the additive homomorphic encryption. Based on this protocol, we construct our inner product protocol with the inner product of the mining query vector and transaction record vector can be securely computed. By proposing another comparison protocol, we construct the frequent itemset mining protocol and the final association rule mining protocol. Finally, we give the theoretical security of the proposed protocols which has proven that our protocol is secure in the defined security models and show its practical performance analysis.

C. ORGANIZATION

The rest of this paper is organized as follows. Section 2 introduces the preliminary knowledge for designing our protocols. In Section 3, we define the system model and attack model. In Section 4, we give a concrete construction including a multiplication protocol, inner product protocol, comparison protocol, frequent itemset protocol, and association rule mining protocol. In Section 5, we give the security proof of our protocols. In Section 6, we show the performance analysis. Finally, in Section 7, we give the conclusion.

II. PRELIMINARIES

In this section, we recall the definitions of some concepts, including frequent itemset mining and association rule mining, Paillier cryptosystem, and garbled circuit.

To begin with, we list the basic notations throughout this paper in Table 1. It is worth noting that both the transaction records \( t_i \in T \), \( i = \{1, 2, \cdots, m\} \) and the mining query \( q \) are represented as binary bit strings with bitlength \( l \), where \( l \) denotes the number of items in the database.

| Table 1. Notations. |
|---------------------|
| Notations          | Definitions                  |
| \( I \)             | Item set \( I = \{item_1, item_2, \cdots, item_l\} \) |
| \( T \)             | Transaction record set \( T = \{t_1, t_2, \cdots, t_m\} \) |
| \( t_i \) \( i \in T \) | Where \( t_i \) \( \in \{0, 1\} \) for \( i \in \{1, 2, \cdots, l\} \) |
| \( q \)             | The support of mining query \( q \) |
| \( \|q\|_1 \)       | The 1-number of \( q \) |
| \( supp_{\text{min}} \) | The support threshold |
| \( X \implies Y \)  | The association rule from \( X \) to \( Y \) |
| \( conf(X \implies Y) \) | The confidence of association rule \( X \implies Y \) |
| \( conf_{\text{min}} \) | The confidence threshold |
| \( Z_N \)           | \( Z_N = \{0, 1, \cdots, N - 1\} \) |
| \( Z_{N^2} \)       | \( Z_{N^2} = \{i | gcd(i, N^2) = 1, i \in [1, N^2 - 1]\} \) |
| \( |M| \)           | The Paillier encryption of \( M \in Z_N \) |
| \( B(x) \)          | The bitlength of \( x \) |

A. FREQUENT ITEMSET MINING

Frequent itemset mining [33] finds frequently occurring item sets, sequences, or substructures from the database and provides some support for possible decisions. It is also the basic concept of association rule mining.

We take an example to explain the definition of frequent itemset mining with the given item set \( I = \{item_1, item_2, \cdots, item_l\} \) and transaction record set \( T = \{t_1, t_2, \cdots, t_m\} \). For a mining query \( q \) and a component \( t_i \in T \), where \( q, t_i \in \{0, 1\}^l \), if

\[
q \cdot t_i = \|q\|_1
\]

holds, the transaction record \( t_i \) is said to support the mining query \( q \), which means the items in the mining query \( q \) are also contained in the component \( t_i \), where \( \|q\|_1 \) denotes the 1-number of \( q \). The total number of transaction records supporting the mining query \( q \) in \( T \) is called the support of the mining query \( q \), namely \( supp(q) \). Compared with the support threshold \( supp_{\text{min}} \), if

\[
supp(q) \geq supp_{\text{min}},
\]

the mining query \( q \) is defined as a frequent itemset in \( T \).

B. ASSOCIATION RULE MINING

Association rule mining is to find the currently unknown and valuable relationship between some items in the database under the premise of frequent itemset mining. The association rule between the itemsets \( X \) and \( Y \) is defined as \( X \implies Y \), where \( X \subseteq I, Y \subseteq I, X \neq \emptyset, X \cap Y = \emptyset \). The index that measures the close relationship of the association rule \( X \implies Y \) is called confidence, namely \( conf(X \implies Y) \), we have

\[
conf(X \implies Y) = \frac{supp(X \implies Y)}{supp(X)} = \frac{supp(X \cup Y)}{supp(X)}.
\]
TABLE 2. Location information dataset T.

| Location | ID  | t1 | t2 | t3 | t4 | t5 |
|----------|-----|----|----|----|----|----|
| School   | 0   | 1  | 0  | 1  | 1  |    |
| Home     | 1   | 1  | 1  | 1  | 0  |    |
| Restaurant| 1   | 1  | 1  | 1  | 0  |    |
| Market   | 1   | 0  | 0  | 0  | 0  |    |
| Train    | 0   | 0  | 1  | 0  | 1  |    |

It can also be used to describe the proportion of Y in the presence of X.

In particular, taking into a threshold \( \text{conf}_{\min} \) of the confidence into consider, \( X \Rightarrow Y \) is called a strong association rule if \( \text{conf}(X \Rightarrow Y) \geq \text{conf}_{\min} \) and \( X \cup Y \) is a frequent itemset. And it should be known that in practical applications, the industry field always concentrates on finding strong association rules to pursue more economic benefits. Our association rule mining protocol, therefore, is to find strong association rules.

We still take an example to illustrate this definition. As shown in Table 2, people’s location records in the daily life are listed. Each location record can be represented as a binary vector, as is the mining query. If a location is included in a person’s location record, it is represented by 1. Otherwise, it is represented by 0. Assuming that the mining query is \( q = (0, 1, 1, 0, 0) \), to determine whether the location information record \( t_i \) supports \( q \) or not, we perform the following.

- Compute the inner product of two vectors \( t_i, q \) to obtain \( p_i \).
- Check whether \( p_i = \| q \|_1 \) holds or not.
  - If \( p_i = \| q \|_1 \), \( t_i \) supports \( q \).
  - Otherwise, \( t_i \) does not support \( q \).

Therefore, the support for mining query \( q \) is calculated as

\[
\text{supp}(q) = \sum_{i=1}^{5} \frac{q \cdot t_i}{\| q \|_1},
\]

where \( q \cdot t_i \) denotes the inner product of \( q \) and \( t_i \).

For a rule \( X \Rightarrow Y \), to find whether it is a strong association rule, we perform the following.

- Compute \( \text{supp}(X \cup Y) \) and \( \text{supp}(X) \).
- Given the defined \( \text{supp}_{\min} \), if either \( \text{supp}(X \cup Y) < \text{supp}_{\min} \) or \( \text{supp}(X) < \text{supp}_{\min} \), \( X \Rightarrow Y \) is not a strong association rule.
- Otherwise, compute \( \text{conf}(X \Rightarrow Y) = \frac{\text{supp}(X \cup Y)}{\text{supp}(Y)} \).
- Given the defined \( \text{conf}_{\min} \), if \( \text{conf}(X \Rightarrow Y) \geq \text{conf}_{\min} \), the rule \( X \Rightarrow Y \) is defined as a strong association rule. Otherwise, it is not.

C. PAILLIER CRYPTOSYSTEM

The Paillier homomorphic encryption [25] is one of the most practical addition homomorphic encryption algorithms. We recall its definition as follows.

- **Keygen(\( k \)):** Taking as input a secure parameter \( k \), two large prime numbers \( p, q \), the key generation algorithm calculates \( N = pq \) and \( \lambda = lcm(p - 1, q - 1) \), randomly choose \( g \in \mathbb{Z}_{N^2}^* \), satisfying \( gcd(L(g^\lambda \mod N^2), N) = 1 \) for \( L(x) = (x - 1)/N \), and sets the public and secret key pair as
  \[
pk = (N, g), \ sk = \lambda.
\]
- **Enc(\( pk, M \)):** Taking as input the public key \( pk \) and a message \( M \in \mathbb{Z}_N \), the encryption algorithm randomly selects \( r \in \mathbb{Z}_N \) and computes
  \[
  C = g^M r^N \mod N^2.
  \]
- **Dec(\( sk, C \)):** Taking as input a secret key \( sk = \lambda \) and a ciphertext \( C \), the decryption algorithm computes
  \[
  M = \frac{L(C^{\lambda} \mod N^2)}{L(g^\lambda \mod N^2)} \mod N.
  \]

In terms of the homomorphism, Paillier encryption owns several interesting features as follows.

- \([M_1] \cdot [M_2] = [M_1 + M_2] \). Given two ciphertexts \( C_1 = g^{M_1 r_1^N} \mod N^2, C_2 = g^{M_2 r_2^N} \mod N^2 \), we can compute
  \[
  C_3 = C_1 C_2 = g^{M_1 + M_2 (r_1 r_2)^N} \mod N^2
  \]
  which is an encryption of \( M_1 + M_2 \).
- \([M]^s = [s M] \). Given a number \( s \in \mathbb{Z}_N \) and a ciphertext \( C = g^M r^N \mod N^2 \), we can obtain
  \[
  C^s = g^{sM (r^s)^N} \mod N^2,
  \]
  which is an encryption of \( s M \).
- \([M]^{-1} = [-M] \). Given a ciphertext \( C = g^M r^N \mod N^2 \) encrypted with \( M \in \mathbb{Z}_N \), we have
  \[
  C^{-1} = g^{-M} r^{-N} \mod N^2,
  \]
  which is an encryption of \( -M \).
- Given a ciphertext \( C_1 = g^{M_1 r_1^N} \mod N^2 \), we can compute
  \[
  C_2 = C_1^{M_2 r_2^N} = g^{M_1 M_2 (r_1^{M_2} r_2^N)} \mod N^2,
  \]
  which is an encryption of \( M_1 M_2 \).

D. GARBLED CIRCUIT

Garbled circuit [34] enables two-party secure computation, where two mistrusting parties can jointly evaluate a function over their private inputs and the function is described as a Boolean circuit.

Below, we take the scenario in Fig.1 as an example to illustrate the workflow of garbled circuit, where Alice owing
Z. Hong et al.: Secure Privacy-Preserving Association Rule Mining With Single Cloud Server

bit \( u \in \{0, 1\} \) and Bob owing bit \( v \in \{0, 1\} \) intends to perform the logical AND privately on \( u, v \) using the garbled circuit. The AND gate has two input lines \( x, y \) and one output line \( z \), and each line has two possible values of 0 and 1.

- Alice sets two random key-values for each line representing 0 and 1 respectively, namely \( K_{0,i}, K_{1,i} \) for line \( i \in \{x, y, z\} \).
- Alice uses the key-value to encrypt the truth table as Table 3. It then sends \( K_{0,x}, K_{0,y}, K_{0,z} \) and the disordered encrypted truth table to Bob.
- Bob applies \( K_{u,i}, K_{v,i} \) to decrypt each row of the disordered truth table and returns the decryption results \( K_{w,i} \) to Alice.
- Since only one row can be successfully decrypted, there will be only one \( w \in \{1, 2, 3, 4\} \) satisfying \( K_{w,i} = K_{k,z}, k \in \{0, 1\} \). The logical AND of \( u \) and \( v \) is, therefore, \( k \).

Since the received key-values are all randomly chosen numbers from the point of view of Bob, he will not obtain any additional information.

### III. SYSTEM MODEL AND ATTACK MODEL

#### A. SYSTEM MODEL
We are committed to achieving privacy-preserving association rule mining with the single-cloud setting in the scenario where a miner submits the mining query to the cloud server that has collected a large set of encrypted transaction records from data owners.

As shown in Fig. 2, the system model consists of a cloud service provider (CSP), data owner (DO), and data miner (DM). Specifically, the CSP in our system is considered to be honest but curious, namely the semi-honest model.

- Data Owner (DO): Data owners upload encrypted transaction records to the cloud server to perform association rule mining.
- Data Miner (DM): The data miner intends to mine potentially unknown association rules of the items by outsourcing mining queries to the cloud server.
- Cloud Service Provider (CSP): The cloud server receives the mining query from data miner, performs association rule mining on the encrypted transaction records uploaded by the data owner, and sends the mining result back to the data miner.

#### B. ATTACK MODEL
We mainly consider external attackers and internal attackers. For external attackers, the active attack method is relatively expensive, while the passive attack cost is extremely low and does not leave traces. Therefore, the main attack method of the external attacker we consider is the passive attack. Besides, internal attackers can collect intermediate results in the computation process. The ability of attacker \( A \) is defined as follows:

- \( A \) can obtain communication data between all entities by eavesdropping on the public channel.
- \( A \) can corrupt the CSP to try to obtain privacy or mining results of DO and DM.
- \( A \) can corrupt the DM and the CSP meanwhile to try to obtain the privacy of DO.

Note that, the CSP is not assumed to collude with the DO since this will directly lead to the privacy disclosure of DO.

To resist the attacks defined in the above attack models, our protocol is supposed to protect the mining query and mining result privacy of DM and data privacy of DO.

### IV. PROTOCOLS
In this section, we construct a privacy-preserving association rule mining protocol under the single-cloud setting by invoking four protocols, namely the multiplication protocol, inner product protocol, comparison protocol, and frequent itemset mining protocol, separately. In particular, the proposed multiplication protocol can be regarded as a general construction basing on the garbled circuit and any additive homomorphic encryption system. In this protocol, to facilitate understanding, we instantiate it by applying the Paillier cryptosystem. In the following discussion,
we refer the encryption and decryption to the Paillier cryptosystem unless specified otherwise. Besides, to further clarify how the invoked four protocols support the final association rule mining protocol, we illustrate their relations in Fig. 3.

Next, we show a rough overview of the protocols. Combined with the definition of association rule mining given in Sec. II, to compute the association rule of \( X \implies Y \) in the Association Rule Mining Protocol, we must compute out \( supp(X \cup Y) \) and \( supp(X) \) which needs the join of Frequent Itemset Mining Protocol, and to compare the ratio value of them with the given threshold \( conf_{min} \) which needs to invoke the Comparison Protocol. Also, according to the definition of frequent itemset mining given in Sec. II, to compute the support of a mining query in the transactions, we need to compute out the inner product of this query and each transaction record that requires Inner Product Protocol, and to compare the value with the support threshold \( supp_{min} \) that requires Comparison Protocol. As defined, the transaction records and mining queries involved in the computation are represented as binary vectors. By splitting the inner product operation on vectors into multiplications and additions, we have that the Inner Product Protocol can invoke the Multiplication Protocol to compute the inner product value.

Suppose that the DM runs the Paillier homomorphic encryption to generate its public/secret key pair \((pk, sk)\) and publishes the public key \(pk\).

### A. Multiplication Protocol

To clarify the applied symbols, we summarize the symbols used in the multiplication protocol to Table 4.

Suppose that the DM owns bit \( x \in \{0, 1\} \) and the DO owns bit \( y \in \{0, 1\} \), they interact with each other and the CSP to obtain the privacy-preserving multiplication result of \( x \) and \( y \). Since \( x \cdot y \) equals to the result of logical AND between \( x \) and \( y \), we then combine the AND gate in the garbled circuit and homomorphic encryption to design our multiplication protocol as follows.

- **Initialization of DM.** With the public key \( pk \) generated using Paillier homomorphic encryption, the DM generates three ciphertexts \( C_1, C_2, C_3 \) of message 0 and one ciphertext \( C_4 \) of message 1 as

\[
[0] = \{C_1, C_2, C_3\}, \quad [1] = C_4.
\]

picks random numbers

\[
k_a^0, k_a^1, k_b^0, k_b^1 \in \mathbb{Z}_{N^2}
\]

representing its and DO’s choises of 0 and 1 respectively, and chooses a random secret number \( r_A \in \mathbb{Z}_N \) and a public cryptographic hash function \( H : \{0, 1\}^* \rightarrow \{0, 1\}^N \). The DM computes

\[
C_i' = C_i \times r_A^N \mod N^2_{|i=1,2,3,4|}.
\]

It then generates the secret truth table as Table 5 and shuffles the order of the four secret values to obtain the disordered secret truth values \( T_1, T_2, T_3, T_4 \).

- **DM→DO.** With the \( x \in \{0, 1\} \), the DM selects \( k_a^x \), computes

\[
T_i' = k_a^x \oplus T_i|_{i=1,2,3,4},
\]

and sends \( k_b^0, k_b^1, T_i', C_i' \) to the DO.

- **DO→CSP.** With the \( y \in \{0, 1\} \), the DO chooses \( k_b^y \), computes

\[
k_b^y \oplus T_i'|_{i=1,2,3,4},
\]

and randomly shuffles the order of results to obtain \( T_i'' |_{i=1,2,3,4} \). It then sends \( T_i'' |_{i=1,2,3,4} \) to CSP.

### Table 4. Notations in the multiplication protocol.

| Notations | Definitions |
|-----------|-------------|
| \( x, y \) | \( x, y \in \{0, 1\} \) |
| \( C_1, C_2, C_3, C_4 \) | Ciphertexts of - using Paillier homomorphic encryption |
| \( k_a^0, k_a^1, k_b^0, k_b^1 \) | Random numbers in \( \mathbb{Z}_{N^2} \) |
| \( r_A, R \) | Random number in \( \mathbb{Z}_N \) |
| \( H \) | Hash function \( H : \{0, 1\}^* \rightarrow \{0, 1\}^N \) |
| \( C_i' \) | \( C_i' = C_i \times r_A^N \mod N^2 \) |
| \( T_i' \) | Disordered secret truth values of \( C_i' \) |
| \( T_i'' \) | Disordered secret truth values of \( k_b^y \oplus T_i' \) |

\[
C_i = C_i \times r_A^N \mod N^2_{|i=1,2,3,4|}.
\]

### Table 5. The secret truth table of logical and.

| X | Y | Z | Secret Truth Value |
|---|---|---|-------------------|
| 0 | 0 | 0 | \( k_a^0 \oplus k_b^0 \oplus C_1 \) |
| 1 | 0 | 0 | \( k_a^0 \oplus k_b^0 \oplus C_2 \) |
| 0 | 1 | 0 | \( k_a^0 \oplus k_b^1 \oplus C_3 \) |
| 1 | 1 | 1 | \( k_a^0 \oplus k_b^1 \oplus C_4 \) |
TABLE 6. Notations in the inner product protocol.

| Notations | Definitions |
|-----------|-------------|
| $\mathbf{x}$ | $[\mathbf{x}] = ([x_1], [x_2], \cdots, [x_l])$ |
| $\mathbf{y}$ | $[\mathbf{y}] = ([y_1], [y_2], \cdots, [y_l])$ |
| $W_i$ | $W_i = [x_i \times y_i]$ |
| $V$ | $V = \prod_{i=1}^{l} W_i$, the inner product of $\mathbf{x}$ and $\mathbf{y}$ |

- **CSP→DM.** The CSP randomly chooses a number $R \in \mathbb{Z}_N$, generates four ciphertexts of $R$ as

$$[R] = C_{R,1}, C_{R,2}, C_{R,3}, C_{R,4},$$

and computes

$$\begin{cases} C_i'' = C_i' \times C_{R,i} \mod N^2 \\ h_i = H(C_i'') \\ \tilde{T}_i = \tilde{T}_i'' \times C_{R,i} \mod N^2 \end{cases}_{i \in \{1,2,3,4\}}$$

Finally, the CSP sends $\{\tilde{T}_i\}_{i \in \{1,2,3,4\}}$ to the DM.

- **DM→CSP.** After receiving $\{\tilde{T}_i\}_{i \in \{1,2,3,4\}}$, the DM computes

$$\{U_i = \tilde{T}_i \times r_A^N \mod N^2\}_{i \in \{1,2,3,4\}}$$

and uploads $\{U_i\}_{i \in \{1,2,3,4\}}$ to the CSP.

- **CSP.** The CSP computes

$$\{u_i = H(U_i)\}_{i \in \{1,2,3,4\}},$$

compares $\{u_i\}_{i \in \{1,2,3,4\}}$ and $\{h_i\}_{i \in \{1,2,3,4\}}$, finds the only duplicate data $u_k$, where $k \in \{1, 4\}$. It then computes

$$W = U_k \times g^{-R} \mod N^2$$

and reserves $W$ as the output of the multiplication protocol, which means the underlying bit of $W$ is the multiplication result of $x$ and $y$, namely $W = [x \times y]$.

Note that, the output of the multiplication protocol, $W$, usually will not be returned to the DM since $W$ is just an intermediate result and this will enable the DM to obtain $y$ by decrypting $W$ with $sk$.

B. INNER PRODUCT PROTOCOL

The symbols used in the inner product protocol are summarized in Table 6.

By invoking the above multiplication protocol, we construct the following inner product protocol, where the DM sends the ciphertext of a mining query, $[\mathbf{x}] = ([x_1], [x_2], \cdots, [x_l])$, to the CSP who owns the ciphertext of a transaction record, $[\mathbf{y}] = ([y_1], [y_2], \cdots, [y_l])$.

To obtain the privacy-preserving inner product result of $\mathbf{x}$ and $\mathbf{y}$:

- the CSP invokes the Multiplication Protocol on the same-dimensional component of $\mathbf{x}$ and $\mathbf{y}$, namely $x_i$ and $y_i$, to obtain the ciphertext $W_i = [x_i \times y_i]$ of the operation result, where $i \in \{1, 2, \cdots, l\}$;

C. COMPARISON PROTOCOL

The symbols used in the comparison protocol are summarized in Table 7.

Since the association rule mining consists of the comparison operation, such as between $\text{supp}(q)$ and $\text{supp}_{\text{min}}$, $\text{conf}(X \Rightarrow Y)$ and $\text{conf}_{\text{min}}$, we construct the comparison protocol as follows.

Suppose that the CSP owns two ciphertexts $[s], [t]$, it intends to output the privacy-preserving comparison result of $s$ and $t$ without decryption.

- **CSP→DM.** With $[s], [t]$, the CSP computes

$$[f] = [s - t] = [s] \times [t]^{N-1},$$

selects random numbers $a, b \in \mathbb{Z}_N, c \in \{0, 1\}$ satisfying $a \gg b$ and $B(a) < B(N)/4$, and computes

$$[\alpha] = [(-1)^c a(2f + 1) + b] = ([f]^2 \times [1])^a \times [N-1]^b \times C_b$$

and sends it to the DM.

- **DM→CSP.** The DM decrypts the received ciphertext with $sk$ to obtain $\alpha$.

  -- If $B(\alpha) < B(N)/2$, the DM sets $u = 1$.

  -- Otherwise, it sets $u = 0$.

The DM encrypts $u$ with $pk$ to obtain $[u]$ and sends it to the CSP.

- **CSP.** The CSP performs the following:

  -- If $c = 0$, the CSP outputs $B = [u]$.

  -- Otherwise, it outputs $B = [1 - u] = [1] \times [u]^{N-1}$.

Also, the comparison result $B$ will be returned or not to the DM in terms of the actual scenario and privacy requirement.

D. FREQUENT ITEMSET MINING PROTOCOL

The symbols used in the frequent itemset mining protocol are summarized in Table 8.

Given a transaction record database $T = \{t_1, \cdots, t_m\}$, where $t_i = \{t_{i,1}, \cdots, t_{i,l}\}$, where $m$ is the total number of location records in the database and $l$ is the length of the location record. The DM generates and encrypts a mining query $q$ and a support threshold $\text{supp}_{\text{min}}$ and sends them to
the CSP who will output supp(q) and determine whether q is a frequent itemset.

- **DM→CSP.** The DM encrypts a mining query q = \((q_1, \ldots, q_l)\) and a support threshold supp\(_{\text{min}}\) as \([\{q_1\}, \ldots, \{q_l\}], [\text{supp}\_{\text{min}}]\) and uploads them to the CSP.

- **CSP→DM.** The CSP invokes the Inner Product Protocol to compute \(v_i = [q \cdot t_i]\) for \(i \in \{1, 2, \ldots, m\}\) and obtain

\[
A = \{v_1, \ldots, v_m\}.
\]

The CSP computes \(n\) dummy ciphertexts encrypted with messages chosen from \([0, l]\) and concatenates them to the inner product ciphertext to obtain

\[
A = \{v_1, \ldots, v_m, v_{m+1}, \ldots, v_{m+n}\}.
\]

For any \(i \in \{1, m + n\}\), the CSP randomly selects \(d_i \in \mathbb{Z}_N\), computes

\[
[w_i] = [d_i(v_i - ||q||_1)] = \left([v_i] \times [||q||_1]^{N-1}\right)^{d_i}
\]

and obtains

\[
W = \{[w_1], \ldots, [w_m], \ldots, [w_{m+n}]\}.
\]

It then chooses a secret permutation \(\psi\) to shuffle the order of \(W\) to

\[
W' = \psi(W) = \{[w_1'], \ldots, [w_{m+n}]\}
\]

and sends \(W'\) to the DM.

- **DM→CSP.** The DM decrypts \([w_i']\) with \(sk\) and obtain \(w_i'\), where \(i \in \{1, 2, \ldots, m + n\}\). If \(w_i' = 0\), it sets \(s_i = 1\). Otherwise, it sets \(s_i = 0\). The DM, then, encrypts \(s_i\) to obtain

\[
S = \{[s_1], [s_2], \ldots, [s_{m+n}]\}
\]

and sends it to the CSP.

- **CSP→DM.** The CSP executes the following steps.
  - Perform the inverse permutation \(\psi^{-1}\) on \(S\) to obtain
    \[
    S' = \psi^{-1}(S)
    \]
  - and removes the last \(n\) components to have
    \[
    S'' = \{[s'_1], [s'_2], \ldots, [s'_{m}]\}.
    \]

In particular, we have

\[
s'_i = \begin{cases} 
0, & v_i < ||q||_1 \\
1, & v_i = ||q||_1.
\end{cases}
\]

The CSP computes the ciphertext of the mining query’s support degree as

\[
[supp(q)] = \prod_{i=1}^{m} [s'_i].
\]

- **Invoke the Comparison Protocol** on \(supp(q)\) and \(supp_{\text{min}}\) to obtain the comparison result \(B\) and send it to the DM.

  - **DM.** The DM decrypts \(B\) and obtain the comparison result.
    - If the result is 1, \(q\) is a frequent itemset.
    - Otherwise, \(q\) is not a frequent itemset.

### E. Association Rule Mining Protocol

We construct the association rule mining protocol by invoking the above proposed four protocols. Given a transaction record database \(T = \{t_1, \ldots, t_m\}\), where \(t_i = \{t_{i1}, \ldots, t_{il}\}\), where \(m\) is the total number of location records in the database and \(l\) is the length of the location record. Suppose that the DM intends to determine whether a rule \(X \Rightarrow Y\) is a strong association rule or not, it interacts with the CSP and DO to perform the following steps. Note that, taking the practical applications into consideration, this mining process bases on the fact that \(X\) is a frequent itemset and its support degree \(\text{supp}(X)\) is known.

- **CSP→DM.** Since

\[
\text{conf}(X \Rightarrow Y) = \frac{\text{supp}(X \cup Y)}{\text{supp}(X)},
\]

the CSP interacts with the DM to invoke the Frequent Itemset Mining Protocol on \(X \cup Y\) and each \(t_i\) for \(i \in \{1, 2, \ldots, m\}\) to obtain the ciphertext of support degree of \(X \cup Y\), namely \([\text{supp}(X \cup Y)]\). And the frequent itemset mining result is sent to the DM.

- **DM→CSP.** The DM performs the following.
  - If DM determine \(X \cup Y\) is not a frequent itemset, the rule \(X \Rightarrow Y\) is not a strong association rule.
  - Otherwise, \(X \cup Y\) is a frequent itemset. The DM generates the minimum confidence threshold \(\text{conf}_{\text{min}}\) and uploads it to the CSP.

1) **CSP.** By converting the comparison between

\[
\text{conf}(X \Rightarrow Y) = \frac{\text{supp}(X \cup Y)}{\text{supp}(X)}
\]

and \(\text{conf}_{\text{min}} = \alpha/\beta\) to that between \(\beta \times \text{supp}(X \cup Y)\) and \(\alpha \times \text{supp}(X)\). The CSP sets

\[
\text{conf}_{\text{min}} = \alpha/\beta,
\]

and computes

\[
[\beta \times \text{supp}(X \cup Y)] = [\text{supp}(X \cup Y)]^\beta,
\]

\[
[\alpha \times \text{supp}(X)] = [\text{supp}(X)]^\alpha
\]
and invokes the Comparison Protocol on \([\beta \times \text{supp}(X \cup Y)]\) and \([\alpha \times \text{supp}(X)]\) to obtain its comparison result \(B\). It then sends \(B\) to the DM.

2) DM. The DM decrypts \(B\) to obtain the comparison result. If it is 1, the rule \(X \implies Y\) is supposed to be a strong association rule. otherwise, it is not a strong association rule.

This completes our association rule mining protocol.

V. SECURITY ANALYSIS

In this section, we give the theoretical security analysis of the proposed protocols as below based on the defined attack model in section III-B. Since our protocol is assumed to be constructed under the semi-honest model, we first recall the definition of security in the semi-honest model as below.

Given a protocol \(\pi\), let \(a_i\) be the input of participant \(P_i\) and \(b_i\) be the output of \(P_i\); let \(\text{REAL}^{\pi}_i(\pi)\) be the view of \(P_i\) in the real-world execution of the protocol \(\pi\); and let \(\text{IDEAL}^{\pi}_i(\pi)\) be the view of \(P_i\), which is simulated from \(a_i\) and \(b_i\), in the ideal world execution of the protocol \(\pi\). If \(\text{REAL}^{\pi}_i(\pi)\) is computationally indistinguishable from \(\text{IDEAL}^{\pi}_i(\pi)\), the protocol \(\pi\) is said to be secure in the semi-honest model.

Below, we give the security proof of the proposed multiplication protocol, inner product protocol, comparison protocol, frequent itemset mining protocol, and association rule mining protocol in Lemma 1, 2, 3, 4, and 5 separately.

Lemma 1: The proposed multiplication protocol is secure under the semi-honest model.

Proof: We can have the execution view of data miner \(W\) in the real world as \(\text{REAL}^{\pi}_i((\tilde{T}_i)_i)\in\{1,2,3,4\}\), where \(\tilde{T}_i = T'_i \times C_{R,i}, C_{R,i} = [R]\) for \(R \in \mathbb{Z}_N\) randomly chosen by the cloud service provider. We assume that \(\text{IDEAL}^{\pi}_i(W)((\tilde{T}_i)_i)\in\{1,2,3,4\}\) is the execution view of a simulated data miner in an ideal world. Among them, \(T'_i\) is a random number in \(\mathbb{Z}_{N^2}\). Since the data miner cannot extract the information of random number \(R, \tilde{T}_i\) and \(T'_i\) are indistinguishable from its point of view. We, therefore, have that \(\text{REAL}^T_i\in\{1,2,3,4\}\) and \(\text{IDEAL}^{\pi}_i(W)((T'_i, \bar{T}_i)_i)\in\{1,2,3,4\}\) are computationally indistinguishable.

For the data owner, we represent its execution view in the real world as \(\text{REAL}^{\pi}_i(k_0^b, k_1^b, (T'_i, \bar{T}_i, C_i)_i)\in\{1,2,3,4\}\), where \(k_0^b, k_1^b\) are random numbers defined by the data miner, and \(C_i\) is also random from the point of view of the data owner. Since \(C_i = C_i \times r_a \mod N^2\) with random number \(r_a\) chosen by the data miner and \(\{C_i\}_i\in\{1,2,3\} = [0]\). \(C_4 = [1]\) are encrypted with Paillier encryption scheme, we have \(C_i\) and \(C_i'\) are random from the point of view of the data owner for \(i \in \{1, 2, 3, 4\}\). Since \(T'_i = k_2^b \oplus \bar{T}_i\), where \(\bar{T}_i\) is confused by \(C_i\), we have \(\{T'_i\}_i\in\{1,2,3,4\}\) are random for the data owner. We assume that \(\text{IDEAL}^{\pi}_i(k_0^b, k_1^b, (T'_i, \bar{T}_i, C_i)_i)\in\{1,2,3,4\}\) is the execution view of the simulated data owner in the ideal world, where \(k_0^b, k_1^b, \bar{T}_i, C_i \in \mathbb{Z}_{N^2}\) are random chosen. This means \(k_0^b, k_1^b\) and \(k_0^p, k_1^p\) are indistinguishable from the point of view of the data owner. Also, \(T'_i, C_i\) and \(\bar{T}_i, \bar{C}_i\) are computationally indistinguishable for each \(i \in \{1, 2, 3, 4\}\).

Based on the above analysis, we have that \(\text{REAL}^{\pi}_i(k_0^b, k_1^b, (T'_i, C_i)_i)\in\{1,2,3,4\}\) and \(\text{IDEAL}^{\pi}_i(k_0^b, k_1^b, (\bar{T}_i, \bar{C}_i)_i)\in\{1,2,3,4\}\) are computationally indistinguishable.

Proofs about the cloud service providers are similar to that of the data miner and data owner. Combining these three results, we have that the proposed multiplication protocol is secure under the semi-honest model.

Lemma 2: The proposed inner product protocol is secure under the semi-honest model.

Proof: Since the proposed inner product protocol comes directly from the constructed multiplication protocol and additive homomorphic encryption whose security has been proved in Lemma 1 and [25] respectively, we have that our proposed inner product protocol is secure under the semi-honest model.

Lemma 3: The proposed comparison protocol is secure under the semi-honest model.

Proof: According to the setting of the proposed comparison protocol, we have the execution view of the data miner in the real world is \(\text{REAL}^{\pi}_i((\alpha, \alpha'), (\tilde{\alpha}, \tilde{\alpha}'))\), where \(\alpha = (-1)^{f_1} \cdot a \cdot (2^f + 1) + b, a, b \in \mathbb{Z}_N, c \in \{0, 1\}\) are randomly chosen by the CSP. This leads to a fact that \(\alpha\) is random from the point of view of the data miner. Assume that \(\text{IDEAL}^{\pi}_i(W)((\alpha', \tilde{\alpha}')\) is the execution view of a simulated data miner in an ideal world, where \(\alpha', \tilde{\alpha}' \in \mathbb{Z}_{N^2}, \alpha', \tilde{\alpha}' \in \mathbb{Z}_N\) are randomly chosen. With the semantic security of the encryption, \(\{\alpha\}\) and \(\{\alpha'\}\) are computationally indistinguishable. Upon that both \(\alpha\) and \(\alpha'\) are two random numbers from the point of view of the data miner, they are computationally indistinguishable. Based on the above analysis, we have that \(\text{REAL}^{\pi}_i(A, \alpha, a, \alpha)\) and \(\text{IDEAL}^{\pi}_i(W'(A', \alpha', u))\) are computationally indistinguishable.

The security proof of cloud service provider is similar to that of the data miner. Combining these two results, we have that the proposed comparison protocol is secure under the semi-honest model.

Lemma 4: The proposed frequent itemset mining protocol is secure under the semi-honest model.

Proof: Note that, the invoked inner product and comparison protocols in the proposed frequent itemset mining protocol have been proved to be secure in the above lemmas. Besides, we show the detailed security analysis of the frequent itemset mining protocol except for these two protocols as below.

According to the setting of the proposed frequent itemset mining protocol, we have the execution view of the data miner in the real world is \(\text{REAL}^{\pi}_i(W', B)\), where \(W' = \psi(W) = \{[w'_1], \cdots, [w'_{m+n}]\} \in [0, 1, \cdots, l]^{m+n}\) and \(B \in \{0, 1\}\). Assume that \(\text{IDEAL}^{\pi}_i(W'(\tilde{W}', B'))\) is the execution view of a simulated data miner in an ideal world, where \(\tilde{W}' \in \{0, 1, \cdots, l\}^{m+n}\) and \(B' \in \{0, 1\}\) are randomly chosen. Since \(\psi\) is a permutation randomly chosen by the cloud service provider, \(W = \{[w_1], \cdots, [w_{m+n}]\}, [w_i] = [d(N_i - \lceil q \rceil)]\) and \(v_i \in \{0, 1, \cdots, [l]\}\) are dummy ciphertexts computed by the cloud service provider, and \(d_i \in \mathbb{Z}_N\) are randomly chosen by the cloud service provider,
we can have that although the data miner can decrypt $W'$ to \{$w'_1, \cdots , w_{m+n}$\}, this vector is indistinguishable from a random vector from \{0, 1, \cdots, l\}^{m+n}. Since $B \in \{[0], [1]\}$ is the encryption of the comparison result, the data miner cannot extract the additional information from $B$ except the comparison result. Based on the above analysis, we have that $\text{REAL}_{\text{W}}^1(W', B)$ and $\text{IDEAL}_{\text{W}}^1(W', B')$ are computationally indistinguishable. The security proof of cloud service provider is similar to that of the data miner.

Next, we discuss the data confidentiality and query privacy against an external adversary $A$. Assume that $A$ can eavesdrop on the transmission link between each entity such that it can obtain all the uploaded data. Since the uploaded data is encrypted by the Pailler cryptosystem whose security has been proved in [25], $A$ cannot extract any information of the original data. If $A$ can comprise all the data owners, they still cannot obtain the miner’s query since all the intermediate results are ciphertext. As long as the cloud service provider does not cooperate with the data miner, all the data confidentiality and query privacy will be protected. The security proof of cloud service provider is similar to that of the data miner.

The security of the Paillier encryption algorithm comes from finding two large prime numbers. The encryption algorithm mainly consists of two modular exponentiation operations. The computational complexity of the modular exponentiation operation is $O(k^3)$. Similarly, the computational complexity of the decryption algorithm is also $O(k^3)$. The additive homomorphic operation is a modular multiplication operation.

2) COMMUNICATION OVERHEAD

Based on the above assumptions, the modulus $N$ used in the scheme is $2k$ bits. In the following, we estimate the sub-protocol communication overhead. The multiplication protocol requires three-party communication. Among them, the communication traffic between the data miner and the data owner is about $40k$ bits, between the data miner and cloud service provider is about $32k$ bits, and between the data owner and cloud service provider is about $16k$ bits. The inner product protocol is designed based on the secure multiplication protocol, and its communication traffic depends on the length of the position information record vector and the communication traffic of the secure multiplication protocol. The comparison protocol is executed by the data miner and CSP. The data miner uploads about $4k$ bits of data to the cloud service provider who needs to send about $4k$ bits of data to the data miner. The amount of communication between them is about $8k$ bits.

B. EXPERIMENTAL ANALYSIS

The experimental simulation of this scheme was tested on a notebook computer running Windows 10 with Intel Core i7-8750H 2.20GHz CPU and 8GB RAM. We implemented the Pailler encryption algorithm with a large integer $N$ of 1024 bits in Visual Studio through the Miracle library in the C++ development kit and tested the proposed scheme using this encryption algorithm. In our experiments, we first tested the performance of the scheme and the efficiency of the sub-protocols, including time cost and communication cost. Then, we conducted simulation experiments on the secure frequent itemset mining protocol on an original database provided by Roberto Bayardo. There are 3196 location information records in the original database, and the record vector length is 75.

In order to facilitate the experiment, we selected some transaction records from the original database to form a test database. In addition, we analyze the performance of this program by changing the parameters. It can be seen from the agreement that in the time overhead of the participants, encrypting the data occupies the main part of the time

VI. PERFORMANCE ANALYSIS

In this section, we analyze the proposed protocols in terms of theoretical and experimental aspects as below.

A. THEORETICAL ANALYSIS

1) ENCRYPTION OVERHEAD

The security of the Paillier encryption algorithm comes from the difficulty of decomposing large integers. We assume that the modulus $N = pq$ used in this scheme is $2k$ bitlength. Next, we conduct a theoretical analysis of the encryption’s performance. The public key of the Paillier encryption is $(N, g)$ and the secret key is $\lambda = \text{lcm}(p - 1, q - 1)$. The bitlength of $g$ is approximately equal to the bitlength of $N^2$, that is, $4k$. The bitlength of the public key is about $6k$, and the bitlength of the secret key $\lambda$ is slightly less than $2k$. The size of the plaintext space is $N$, and the bitlength of ciphertext is about $4k$, so the ciphertext/plaintext length expansion ratio is 2.

The time cost of its key generation algorithm mainly comes from finding two large prime numbers. The encryption algorithm mainly consists of two modular exponentiation operations. The computational complexity of the modular exponentiation operation is $O(k^3)$. Similarly, the computational complexity of the decryption algorithm is also $O(k^3)$. The additive homomorphic operation is a modular multiplication operation.
TABLE 9. Performance analysis of the Paillier encryption.

| Key generation time | Encryption time | Decryption time | Homomorphic operation time | Ciphertext length |
|---------------------|----------------|----------------|--------------------------|------------------|
| 89.49ms             | 11.62ms        | 18.94ms        | 0.01ms                   | 0.2495KB         |

TABLE 10. Time cost of sub-protocol.

| Sub-protocol             | Data miner (ms) | CSP (ms) | Data owner (µs) |
|--------------------------|-----------------|----------|-----------------|
| Multiplication Protocol  | 48.22           | 48.91    | 23.24           |
| Inner Product Protocol   | 3.63            | 3.68     | 1.75            |
| Comparison Protocol      | 30.76           | 46.43    | N/A             |

TABLE 11. Communication cost of sub-protocol.

| Sub-protocol             | Data miner (KB) | CSP (KB) | Data owner (KB) |
|--------------------------|-----------------|----------|-----------------|
| Multiplication Protocol  | 3.496           | 0.999    | 0.999           |
| Inner Product Protocol   | 262.191         | 74.952   | 74.952          |
| Comparison Protocol      | 0.249           | 0.249    | N/A             |

overhead, while in the communication overhead, the bit length of the ciphertext data is greater than the bit length of the plaintext data. However, in general, the computing and communication resources of data miners and data owners are limited. Therefore, the performance of the encryption scheme is an important factor affecting the performance of this scheme.

We use 512-bit large prime numbers $p$ and $q$ and 1024-bit large integer $N$ to implement 1000 simulation tests of the Paillier encryption. As listed in Table 9, the key generation time cost is about 89.49ms, the encryption time cost is about 11.62ms, the decryption time cost is about 18.94ms. The homomorphic operation is only one modular multiplication, and the time consumption cost is about 0.01 ms. The average length of the ciphertext is about 0.2495KB.

When examining the performance of the sub-protocol in this scheme, the number of 1s in the mining query vector we randomly generated is 15. We use the above encryption and evaluate the three sub-protocols 200 times respectively and list the average time cost and the average communication cost of the sub-protocols in Table 10 and 11.

As listed in the table, we obtained the following results. In the multiplication protocol, the data miner runs about 48.22ms to complete the computation and needs to send about 3.496KB of data to the other two parties. Among them, about 0.999KB of data is sent to the cloud service provider and about 2.497KB of data is sent to the data owner. The data owner only needs to run about 23.24µs and sends about 0.999KB to the cloud service provider. And the service provider runs about 48.91ms and sends about 0.999KB to the data miner. In the inner product protocol, the computation cost and communication cost of the parties are about 75 times that of the parties in the multiplication protocol. The computational time costs of the data miner, data owner, and cloud service provider are about 3.63s, 3.68s, and 1.75ms respectively. Note that the computation cost of the data owner is much lower than that of the data miner. Their communication costs are about 262.191KB, 74.952KB, and 74.952KB, respectively. In the comparison protocol, participants are the data miner and cloud service provider whose computational time costs and communication costs are 30.76ms, 0.249KB, and 46.43ms, 0.249KB, respectively.

In order to better observe the total computation and communication costs of the frequent itemset mining protocol, we used different parameters to simulate the protocol. Suppose that when the protocol is executed, the number of records in the database is $m$, and the number of dummy ciphertexts added by CSP is $n$. After randomly generating the mining query vector, we performed the secure frequent itemset mining protocol on the mining query vector in the test database and tested the total computation time cost and total communication cost of the protocol in the cases of $m = 1000$, $n = 500$, $m = 2000$, $n = 1000$, and $m = 3000$, $n = 1500$.

As listed in Table 12, the total computation time costs for the three cases are 154.48min, 315.73min, and 461.32min, respectively, and the total communication cost is 366.23MB, 732.37MB, and 1098.70MB. Through analysis, we have that the factors affecting the total communication cost of the secure frequent itemset mining protocol are mainly the number of data records in the database and the length of each record vector. In the association rule mining protocol which is designed based on the frequent itemset mining protocol, the number of 1s in the mining query vector determines the number of times performing the frequent itemset mining protocol.

After simple reasoning, it can be obtained that the total computation time and total communication cost of the security association rule mining protocol are approximately proportional to that of the secure frequent itemset mining protocol, and exponentially related to the number of 1 in the mining query. It is worth noting that to show the performance more intuitively, we do not perform the mining job in a distributed way which will effectively reduce the time cost.

VII. CONCLUSION

In this paper, we propose a privacy-preserving association rule mining protocol on encrypted cloud data under a single cloud server. To realize the security under the single-cloud setting, we construct the multiplication protocol which is the
base of association rule mining protocol using garbled circuit and additive homomorphic encryption system. Compared with the state-of-art works, our protocol reduces the number of cloud servers and enhances the flexibility of cloud outsourcing association rule mining. In future work, we will focus on further improving the efficiency of our scheme.

REFERENCES

[1] T. Brijs, G. Swinnen, K. Vanhoof, and G. Wets, “Using association rules for product assortment decisions: A case study,” in Proc. 5th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD), 1999, pp. 254–260.

[2] S. E. Brossette, A. P. Sprague, J. M. Hardin, K. B. Waite, W. T. Jones, and S. A. Moser, “Association rules and data mining in hospital infection control and public health surveillance,” J. Amer. Med. Inform. Assoc., vol. 5, no. 4, pp. 373–381, Jul. 1998.

[3] B. Mobasher, N. Jain, E.-I. Han, and J. Srivastava, “Web mining: Pattern discovery from world wide web transactions,” 1996.

[4] P. Mell and T. Grance, “Draft NIS working definition of cloud computing,” vol. 50, no. 6, p. 50, 2009. [Online]. Available: http://csrc.nist.gov/groups/SNS/cloud-computing/index.html

[5] F. Giannotti, L. V. S. Lakshmanan, D. Pedreschi, H. Wang, and A. Monreale, “Privacy-preserving mining of association rules from outsourced transaction databases,” IEEE Trans. Knowl. Data Eng., vol. 7, no. 3, pp. 385–395, May 2013.

[6] C.-H. Tai, P.-S. Yu, and M.-S. Chen, “K-support anonymity based on pseudo taxonomy for outsourcing of frequent itemset mining,” in Proc. 16th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD), 2010, pp. 473–482.

[7] P. Samarati and L. Sweeney, “Generalizing data to provide anonymity when disclosing information (abstract),” in Proc. 17th ACM SIGACT-SIGMOD-SIGART Symp. Privac. Database Syst. (PODS), 1998, p. 185.

[8] L. Sweeney, “K-anonymity: A model for protecting privacy,” Int. J. Uncertainty, Fuzziness Knowl.-Based Syst., vol. 10, no. 5, pp. 557–570, 2002.

[9] C. Dwork, “Differential privacy,” in Automata, Languages and Programming (Lecture Notes in Computer Science), vol. 4052, M. Bugliesi, B. Preneel, V. Sassone, and I. Wegener, Eds. Berlin, Germany: Springer, 2006, pp. 1–12.

[10] C. Dwork, “The promise of differential privacy: A tutorial on algorithmic techniques,” in Proc. IEEE 52nd Annu. Symp. Found. Comput. Sci., Oct. 2011, pp. 1–2.

[11] M. Bellare, V. T. Hoang, and P. Rogaway, “Foundations of garbled circuits,” in Proc. ACM Conf. Comput. Commun. Security (CCS), 2012, pp. 784–796.

[12] H.-J. Kim, H.-I. Kim, and J.-W. Chang, “Privacy-preserving kNN classification algorithm using Yao’s garbled circuit on cloud computing,” in Proc. IEEE 10th Int. Conf. Cloud Comput. (CLOUD), Jun. 2017, pp. 766–769.

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

[14] J. Wu, N. Mu, X. Lei, J. Le, D. Zhang, and X. Liao, “SeeEDMO: Enabling efficient data mining with strong privacy protection in cloud computing,” IEEE Trans. Cloud Comput., early access, Jul. 30, 2019, doi: 10.1109/TCC.2019.2932065.

[15] S. Patil, S. Joshi, and D. Patil, “Enhanced privacy preservation using anonymization in IoT-enabled smart homes,” in Smart Intelligent Computing and Applications, 2020, pp. 439–454.

[16] S. Patil and S. Joshi, “Improved privacy preservation of personal health records via tokenization,” Int. J. Pure Appl. Math., vol. 118, no. 18, pp. 3035–3042, 2018.

[17] S. Patil and S. Joshi, “Demystifying user data privacy in the world of IoT,” Int. J. Innov. Technol. Exploring Eng., vol. 8, no. 10, pp. 4412–4418, 2019.

[18] A. Shaikh and S. Patil, “A survey on privacy enhanced role based data aggregation via differential privacy,” in Proc. Int. Conf. Adv. Commun. Comput. Technol. (iCACCESS), 2018, pp. 285–290.

[19] M. Kantarcioglu and C. Clifton, “Privacy-preserving distributed mining of association rules on horizontally partitioned data,” IEEE Trans. Knowl. Data Eng., vol. 16, no. 9, pp. 1026–1037, Sep. 2004.

[20] J. Vaidya and C. Clifton, “Privacy preserving association rule mining in vertically partitioned data,” in Proc. 8th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD), 2002, pp. 639–644.

[21] W. K. Wong, D. W. Cheung, E. Hung, B. Kao, and N. Mamoulios, “Security in outsourcing of association rule mining,” in Proc. 33rd Int. Conf. Very Large Data Bases., C. Koch, J. Gehrke, M. N. Garofalakis, D. Srivastava, K. Aberer, A. Deshpande, D. Florescu, C. Y. Chan, V. Ganti, C. Kanne, W. Klas, and E. J. Neuhold, Eds. Vienna, Austria: Univ. Vienna, Austria, Sep. 2007, pp. 111–122.

[22] J. Lai, Y. Li, J. Weng, C. Guan, Q. Yan, and R. H. Deng, “Towards semantically secure outsourcing of association rule mining on categorical data,” IEEE Trans. Cloud Comput., vol. 8, no. 1, pp. 312–323, Jan. 2020.

[23] S. Qiu, B. Wang, M. Li, J. Liu, and Y. Shi, “Toward practical privacy-preserving frequent itemset mining on encrypted cloud data,” IEEE Trans. Cloud Comput., vol. 10, no. 5, pp. 1–12.

[24] P. Paillier, “Public-key cryptosystems based on composite degree residuosity classes,” in Advances in Cryptology (Lecture Notes in Computer Science), vol. 1592, J. Stern, Ed. Berlin, Germany: Springer, May 1999, pp. 223–238.

[25] D. Boneh, E.-J. Goh, and K. Nissim, “Evaluating 2-DNF formulas on ciphertexts,” in Proc. 2nd TCC, vol. 3378, 2005, pp. 325–341.

[26] C. Ma, B. Wang, K. Jooste, Z. Zhang, and Y. Ping, “Practical privacy-preserving frequent itemset mining on supermarket transactions,” IEEE Trans. Syst. J., vol. 14, no. 2, pp. 1992–2002, Jun. 2020.

[27] H.-J. Kim, J.-H. Shin, Y.-H. Song, and J.-W. Chang, “Privacy-preserving association rule mining algorithm for encrypted data in computing,” in Proc. IEEE 12th Int. Conf. Cloud Comput. (CLOUD), Jul. 2019, pp. 487–489.

[28] L. Liu, J. Su, R. Chen, X. Liu, X. Wang, S. Chen, and H. Leung, “Privacy-preserving association rule mining of association rule on outsourced cloud data from multiple parties,” in Proc. ACISP in Lecture Notes in Computer Science vol. 10946, W. Susilo and G. Yang, Eds. Wollongong, NSW, Australia: Springer, Jul. 2018, pp. 431–451.

[29] E. Bresson, D. Catalano, and D. Pointcheval, “A simple public-key cryptosystem with a double trapdoor decryption mechanism and its applications,” in Advances in Cryptology—ASIACRYPT (Lecture Notes in Computer Science), vol. 2894, Berlin, Germany: Springer, 2003, pp. 37–54.

[30] A. Peter, E. Tews, and S. Katzenbeisser, “Efficiently outsourcing multiparty computation under multiple keys,” IEEE Trans. Inf. Forensics Security, vol. 8, no. 12, pp. 2046–2058, Dec. 2013.

[31] H. Pang and B. Wang, “Privacy-preserving association rule mining using homomorphic encryption in a multikey environment,” IEEE J. Sel. Areas Commun., vol. 15, no. 2, pp. 3131–3141, Jun. 2020.

[32] H. Agrawal and R. Srikant, “Fast algorithms for mining association rules,” in Proc. VLDB, Santiago de Chile, Chile, Sep. 1994, pp. 487–499.

[33] A. C. Yao, “How to generate and exchange secrets (extended abstract),” in Proc. 27th Annu. Symp. Found. Comput. Sci., Toronto, ON, Canada, Oct. 1986, pp. 162–167.

ZHIONG HONG received the B.S. and M.S. degrees in computer science and technology from the Shenyang Institute of Technology, Shenyang, China, in 2001 and 2004, respectively, and the Ph.D. degree in computer science and technology from Southwest Jiaotong University, Chengdu, China, in 2014. From 2006 to 2014, he was a Lecturer with the School of Computer, Wuyi University, Jiangmen, China. Since 2015, he has been an Associate Professor. His research interests include intelligent set theory.
ZHILI ZHANG received the Ph.D. degree in computer applied technology from the South China University of Technology, in 2006. He is currently a Professor with the School of Information Engineering, Xuchang University. His research interests include information security, the Internet of Things, and machine learning.

PU DUAN received the Ph.D. degree from Texas A&M University, in 2011. He has been a 20 veteran on cryptography, information security, and networking security. He joined Cisco after obtaining his Ph.D. degree. At Cisco, he led the research and development of new cryptographic algorithms for TLS.13 on Cisco firewall product. He is currently working with the Secure Collaborative Laboratory (SCI), Ant Group, as a Senior Staff Engineer, leading the team on research and implementation of privacy-preserving technologies. He has published more than 20 papers on areas of cryptography, networking security, and system security.

BENYU ZHANG has been engaged in AI research and development for two decades. Since, he went into industry, he has Initiated and led numbers of core AI systems R&D in advertising, search, and recommendation at Google and FB. He is currently leading the Secure Collaborative Intelligence Laboratory (SCI Lab), Ant Group, to build the next generation privacy-preserving data mining and AI platform. The vision is to form the “data internet” and enable data mining and AI applications on data from multiple parties securely, efficiently, and effectively. He published 49 peer-reviewed papers with more than 11,000 citations, has 70 U.S. patents approved and 84 U.S. patents pending.

BAOCANG WANG received the B.S. and M.S. degrees in mathematics and the Ph.D. degree in cryptography from Xidian University, in 2001, 2004, and 2006, respectively. He is currently a Professor with the School of Telecommunications Engineering, Xidian University. His research interests include public key cryptography, wireless network security, and data mining.

WEN GAO received the B.S. degree in electronic information engineering from the Henan University of Technology, China, in 2011, and the Ph.D. degree in cryptography from Xidian University, China, in 2017. She is currently a Lecturer with the College of Electrical and Information Engineering, Shaanxi University of Science and Technology, China. Her research interest includes lattice-based cryptography.

ZHEN ZHAO received the M.S. and Ph.D. degrees in cryptography from Xidian University, China, in 2016 and 2020, respectively. She is currently a Lecturer with the School of Cyber Engineering, Xidian University. Her research interests include the public-key cryptography; in particular, security proof, signature, and encryption schemes.