7-2018

Privacy-preserving mining of association rule on outsourced cloud data from multiple parties

Lin LIU

Jinshu SU

Rongmao CHEN

Ximeng LIU
Singapore Management University, xmliu@smu.edu.sg

Xiaofeng WANG

See next page for additional authors

DOI: https://doi.org/10.1007/978-3-319-93638-3_25

Privacy-Preserving Mining of Association Rule on Outsourced Cloud Data from Multiple Parties

Lin Liu[1], Jinshu Su[1,2(✉)], Rongmao Chen[1(✉)], Ximeng Liu[3,4], Xiaofeng Wang[1], Shuhui Chen[1], and Hofung Leung[5]

[1] School of Computer, National University of Defense Technology, Changsha, China
{liulin16,sjs,chromao,xf_wang,shchen}@nudt.edu.cn
[2] National Key Laboratory for Parallel and Distributed Processing, National University of Defense Technology, Changsha, China
[3] School of Information Systems, Singapore Management University, Singapore, Singapore
snbnix@gmail.com
[4] College of Mathematics and Computer Science, Fuzhou University, Fuzhou, China
[5] Department of Computer Science and Engineering, Chinese University of Hong Kong, Shatin, Hong Kong
lhf@cuhk.edu.hk

Abstract. It has been widely recognized as a challenge to carry out data analysis and meanwhile preserve its privacy in the cloud. In this work, we mainly focus on a well-known data analysis approach namely association rule mining. We found that the data privacy in this mining approach have not been well considered so far. To address this problem, we propose a scheme for privacy-preserving association rule mining on outsourced cloud data which are uploaded from multiple parties in a twin-cloud architecture. In particular, we mainly consider the scenario where the data owners and miners have different encryption keys that are kept secret from each other and also from the cloud server. Our scheme is constructed by a set of well-designed two-party secure computation algorithms, which not only preserve the data confidentiality and query privacy but also allow the data owner to be offline during the data mining. Compared with the state-of-art works, our scheme not only achieves higher level privacy but also reduces the computation cost of data owners.

Keywords: Association rule mining · Frequent itemset mining
Privacy preserving outsourcing · Cloud computing

1 Introduction

Cloud computing has attracted more and more attentions due to its capability of supporting real-time and massive data storing and processing. For a long time, it

W. Susilo and G. Yang (Eds.): ACISP 2018, LNCS 10946, pp. 431–451, 2018.
https://doi.org/10.1007/978-3-319-93638-3_25

has been a growing interest in the paradigm of data mining as a service in cloud computing [1–5]. Since internet giants such as Google and Amazon can collect large-scale data from millions of users and devices, mining on cloud data can also dramatically improve the accuracy and effectiveness of mining. However, when uploading data to cloud service provider, users lose control of their data. Therefore, even though outsourcing data storage and data mining benefit from the scale of economy, it comes with the privacy and security issues.

In this work, we mainly consider the security and privacy problems existing in mining association rule on the outsourced cloud data. Frequent itemset mining, key of association rule, is a popular data mining approach, which is usually employed to discover frequently co-occurring data items and relationships between data items in large transaction databases. These techniques have been widely used in market prediction, intrusion detection, network traffic management and so on. For instance, if customers are buying bread, how likely are they going to buy beer (and what kind of beer) on the same trip to the supermarket? Such information can help retailers do selective marketing and arrange their shelf space for increasing sales. Kantarcioglu and Clifton [6] and Vaidya and Clifton [7] first identified and addressed privacy issues in horizontally and vertically partitioned databases. Due to the increase of data security and privacy demanding, researchers have proposed various methods on privacy-preserving association rule mining. These works can be roughly divided into randomization-based schemes and cryptography-based schemes. Despite the high efficiency in randomization-based schemes, they suffer from the inaccuracy of mining result for adding random noise to the raw data. Compared with the randomization-based scheme, the cryptography-based scheme can apply stronger security level and accurate mining result. Recently, Yi *et al.* [4] have proposed a privacy-preserving association rule mining scheme on the outsourced cloud data encrypted by using ElGamal homomorphic encryption scheme [8]. However, the communication cost was huge due to the fact that their scheme needs n cloud servers to cooperate with each other. Qiu *et al.* [1] proposed a framework for privacy-preserving frequent itemset mining on encrypted cloud data in the twin-cloud architecture. Both of Yi *et al.* [4] and Qiu *et al.* [1] designed three different privacy level protocols, which achieved item privacy, transaction privacy and database privacy respectively. However, even in the highest security level, the mining result was still in plaintext form to the cloud server. Li *et al.* [9] proposed a privacy-preserving association rules mining system on vertically partitioned databases via a symmetric homomorphic encryption scheme. Their scheme achieved high efficiency, but the data owners in that scheme need to stay online during the mining process and some information about the raw data may be revealed.

Motivating Scenario. In this paper, we mainly consider a scenario where a higher privacy level is required. In most cases, the mining result is miner's personal property, which should be kept secret to any other entities including the untrusted cloud server. For example, if the mining result from business data is enterprise's market prediction, leaking this information to competitors will damage this enterprise's profits. In our scenario, it is required that both the raw

data outsourced by the data owners and the mining result for the miner are confidential to the cloud server. Moreover, we consider a large number of data owners and miners in our system, and hence supporting offline users is desirable for improving the system's scalability. In addition, we insist that the frequent itemset mining is the cornerstone for association rule mining. Only mining frequent itemset is not enough to get the strong association rule, which is the key to find the relationship among itemsets. Overall speaking, in this work, we aim at designing a secure scheme, which supports that, (1) the raw data and the mining result are protected from other entities; (2) offline users and; (3) mining both the frequent itemset and association rule simultaneously.

Our Contributions. In this paper, we propose a privacy-preserving association rule mining scheme in the twin-cloud architecture. The contributions of this paper are four-fold, namely:

- To our best knowledge, this is the first work that studies privacy-preserving association rule mining on encrypted data under different keys. Our proposed scheme allows different data owners to outsource their data with different encryption keys to the cloud server for secure storage and processing.
- We build a set of cryptographic blocks for privacy-preserving association rule mining based on BCP cryptosystem [10], which play the cornerstone of our system.
- Based on the cryptographic blocks proposed, we construct a privacy-preserving association rule scheme with multiple keys. And we also prove that our scheme is secure under the semi-honest model.
- We show that our scheme can indeed achieve higher privacy level than most of the recent works [1,4,9]. And also, we fully prove the security of our scheme under the semi-honest mode.

We make a comparison between our work and the most recent works [1,4,9], which is shown in Table 1. In Qiu *et al.*'s work [1] and Yi *et al.*'s work [4], they proposed three different privacy level protocols. Here, we just compare their highest privacy level protocol with ours. Yi *et al.*'s work [4] and Qiu *et al.*'s work [1] can only support frequent itemset mining. Both of their works cannot protect the miner's mining result privacy. Moreover, the data owners' computation cost is highest. Li *et al.*'s [9] algorithm is the most efficient but cannot support the offline data owners. More importantly, their work can only achieve partial data privacy.

Related Work. Data perturbation is widely used to protect sensitive information when outsourcing data mining of association rule. This randomization-based approach can be used to protect the raw data but cannot protect the mining results. Randomization-based approach [3,5] may have unpredictable impacts on data mining precision, due to the random noise added to the raw data. Differential privacy is used to protect privacy mining the association rule. However, the key limitation of such solutions is that the mining results are not accurate with 100%.

Table 1. Comparison summary

Algorithm	Support FIM[a]	Support ARM[b]	Support Offline	D. Privacy[c]	M.R. Privacy[d]	DO Cost[e]	Support multi-key
[4]	Yes	No	Yes	Yes	No	Medium	No
[9]	Yes	Yes	No	Partial	Yes	Low	No
[1]	Yes	No	Yes	Yes	No	High	No
Ours	Yes	Yes	Yes	Yes	Yes	Medium	Yes

[a]FIM means Frequent Itemset Mining.
[b]ARM means Association Rule Mining.
[c]D.Privacy means Data Privacy.
[d]M.R.Privacy means Mining Result Privacy.
[e]DO Cost means Data owner's computation cost.

Compared with randomization-based approaches, cryptography-based approaches usually provide a well-defined security model and an exact mining result for privacy-preserving data mining. Earlier works [6,7] are not efficient enough for the practical requirement facing the prevalent of large scale datasets. Dong and Chen [11] employed an efficient inner product protocol [12] for evaluating association rule mining. But this solution is a two-party protocol, which involves extensive interactions. Lai *et al.* [13] first proposed a semantically secure solution for outsourcing association rule mining with both privacy and mining privacy, but the efficiency is still undesirable for the practice. Yi *et al.* [4] proposed a privacy-preserving association rule mining in cloud computing. To mine association rule from its data, the user outsources the task to $n(\geq 2)$ "semi-honest" servers, which cooperate to perform mining algorithm on encrypted data and return encrypted association rules to the user. In his work $n(\geq 2)$ servers are needed which cause huge communication cost. Li *et al.* [9] proposed a privacy-preserving outsourced association rule mining on vertically partitioned databases. However, their solution still leaks information about the raw data. Most recently, Qiu *et al.* [1] proposed a privacy-preserving frequent itemset mining scheme on outsourced encrypted cloud data. In their work, they proposed three different privacy level protocols. In their privacy level I protocol, only the transaction database in the cloud is encrypted while the miner's query is in plaintext. This protocol work quite efficiently but without protecting the query's privacy. In their protocol II and protocol III, the miner's query is protected or partial protected, but the mining result is known to cloud. For adopting time consuming homomorphic cryptosystem BGN [14], the computation cost of data owners is quite large in protocol II.

2 Preliminaries

In this section, we introduce essential preliminary concepts which serve as the basis of our scheme. Table 2 lists the key notations used throughout this paper.

2.1 Frequent Itemset Mining and Association Rule Mining

Frequent itemset mining, the key of association rule mining, is first proposed by Agrawal *et al.* [15]. Given a set of items, and a transaction databases over these items, frequent itemsets are items which appear with frequency more than a given number. In the following, we give the specific definition of this concept.

Table 2. Notation used

Notations	Definition
pk_{DO_i}/sk_{DO_i}	Public/private key of data owner i
pk_M/sk_M	Public/private key of miner
pk_Σ	The product of all the data owners and miner's public key
$[\![x]\!]_{pk}$	Encrypted data x under pk
MK	Master key of BCP cryptosystem
$\mathbf{mDec}_{(pk,\mathbf{MK})}(X)$	Decrypt X with the master key
$\|x\|$	Bit length of x
$supp(X)$	Support of X
$conf(X)$	Confidence of X
SMAD	Secure multiplication across domain
SCAD	Secure comparison across domain
SC	Secure comparison
SIP	Secure inner product
SFIM	Secure frequent itemset mining

Definition 1 (Frequent Itemset). *Let $I = \{i_1, \cdots, i_m\}$ be a set of items. A transaction T is a set of items. A transaction database is denoted as $T = \{t_1, \cdots, t_m\}$, where m is the total number of transactions. An itemset $X \subseteq I$ is a set of items from I. If $X \subseteq t_i$, X is contained by a transaction t_i. The support of itemset X, is the number of transactions containing X in T, which is referred as $supp(X)$. $supp_{min}$ is the user-defined minimum threshold. If $supp(X) \geq supp_{min}$, X is the frequent itemset.*

The purpose of the frequent itemset mining is to discover the frequency of the item/itemsets, which will further be used to find the relationship of two items. Generally, the relationship between two items are measured by *support* and *confidence*. An association rule is of the form $X \Rightarrow Y$ where $X, Y \subset I$ and $X \cap Y = \emptyset$. The $supp(X \Rightarrow Y)$, support of the rule $X \Rightarrow Y$, is the number of the transactions containing $X \cup Y$. The *confidence* of rule $X \Rightarrow Y$ is a measure of the relation between two items, denoted by $conf(X \Rightarrow Y) = supp(X \Rightarrow Y)/supp(X)$.

Definition 2 (Strong Association Rule). *Assume a minimum support threshold* $supp_{min}$ *and a minimum confidence threshold* $conf_{min}$ *are given. The rule* $X \Rightarrow Y$ *is strong iff* $supp(X \Rightarrow Y) \geq supp_{min}$ *and* $conf(X \Rightarrow Y) \geq conf_{min}$.

Here, we illustrate the above two definition by the following example. A transaction dataset T is given in Table 3. All the items are presented as boolean types, i.e., an item is described as absent by 0, otherwise by 1. Suppose that, if $X = \{Coke\}$, and $Y = \{Milk\}$, we can represent $X \cup Y$ as $\boldsymbol{q} = (0, 1, 1, 0)$. We want to find out that whether $Coke \Rightarrow Milk$ is a strong association rule or not. First, we make an inner product $v_i = \boldsymbol{q} \cdot \boldsymbol{t_i}$, where $\boldsymbol{t_i}, i \in (1, \cdots, 5)$ is the row in the table. It can be easily got that only v_1 and v_3 are equal to 2. Therefore, $supp(X \Rightarrow Y) = 2$. If $supp(X \Rightarrow Y) < supp_{min}$, we can conclude that $X \Rightarrow Y$ is not the strong rule, because $X \cup Y$ is not a frequent itemset. Here, assume that $supp_{min} = 2$, thus $X \cup Y$ is a frequent itemset. Next, we can calculate $supp(X)$ in the same way. In Table 3, it can be easily calculated that $supp(X \Rightarrow Y) = 2$ and $supp(X) = 3$. Therefore, we can easily get $conf(X \Rightarrow Y) = 2/3$. If the $conf(X \Rightarrow Y) \geq conf_{min}$, $X \Rightarrow Y$ is the strong association rule. Otherwise, it's not.

Table 3. Market-basket transaction dataset T

ID	Bread	Coke	Milk	Beer
1	1	1	1	0
2	1	0	0	1
3	0	1	1	1
4	1	1	0	1
5	0	0	1	0

2.2 BCP Cryptosystem

BCP Cryptosystem is an additively homomorphic cryptosystem, proposed by Bresson et al. [10]. BCP is a double decryption mechanism, meaning that it offers two independent decryption mechanisms. The most prominent characteristic of such scheme is that if given the master key of this cryptosystem, any given ciphertext can be successfully decrypted. The BCP cryptosystem works as follows:

Setup(κ): Given a security parameter κ, choose a safe-prime RSA-modulus $N = pq$ (i.e., $p = 2p' + 1$ and $q = 2q' + 1$ for distinct primes p' and q', respectively) of bitlength κ. In the following, we use $|N|$ to denote the length of N. Then a random element $g \in \mathbb{Z}^*_{N^2}$ with order $pp'qq'$ is picked, such that $g^{p'q'} \bmod N^2 = 1 + \lambda N$ for $\lambda \in [1, N-1]$. Thus, the algorithm outputs the public parameter **PP** and the master key **MK** as follows, $\mathbf{PP} = (N, \lambda, g)$ and $\mathbf{MK} = (p', q')$.

KeyGen(**PP**): Randomly pick $a \in \mathbb{Z}^*_{N^2}$ and compute $h = g^a \bmod N^2$. Then, output the public key $\mathbf{pk} = h$ and secret key $\mathbf{sk} = a$.

$\mathbf{Enc}_{(\mathbf{PP},pk)}(m)$: For a given plaintext $m \in \mathbb{Z}_N$, randomly pick $r \in \mathbb{Z}_{N^2}$, then output the ciphtext (A, B) as $A = g^r \bmod N^2$, $B = h^r(1 + mN) \bmod N^2$.

$\mathbf{Dec}_{(\mathbf{PP},sk)}(A, B)$: The plaintext of the given ciphtext (A, B) and secret key $\mathbf{sk} = a$, can be calculated as $m = (B/(A^a) - 1 \bmod N^2)/N$.

$\mathbf{mDec}_{(\mathbf{PP},pk,\mathbf{MK})}(A, B)$: Using the master secret key **MK** of this cryptosystem, the plaintext of the above ciphtertext (A, B) can be calculated as follows. First compute $a \bmod N$ as $a \bmod N = (h^{p'q'} - 1 \bmod N^2)/N \cdot k^{-1} \bmod N$, where k^{-1} denotes the inverse of k modulo N. Then $r \bmod N$ can be computed as $r \bmod N = (A^{p'q'} - 1 \bmod N^2)/N \cdot k^{-1} \bmod N$. Therefore, the when a and r is obtained, the plaintext can be easily get by the following equation, $m = ((B/g^{ar})^{p'q'} - 1 \bmod N^2)/N \cdot (p'q')^{-1} \bmod N$, where $(p'q')^{-1}$ is the inverse of $p'q'$ modulo N.

The BCP cryptosystem is additively homomorphic, which can be verified as $\mathbf{Dec}_{sk}\ (\llbracket m_1 \rrbracket_{pk} \cdot \llbracket m_2 \rrbracket_{pk}) = m_1 + m_2$. Note that for any given $m, k \in \mathbb{Z}_N$, we can easily get $(\llbracket m \rrbracket_{pk})^k = \llbracket k \cdot m \rrbracket_{pk}$. Moreover, if $k = N - 1$, we can get $(\llbracket m \rrbracket_{pk})^{N-1} = \llbracket -m \rrbracket_{pk}$. In this paper, for simplicity we use $\llbracket m \rrbracket_{pk}$ instead of $\mathbf{Enc}_{(\mathbf{PP},pk)}(m)$. More proofs of the correctness and semantic security of the BCP cryptosystem can be found in [10].

3 System Model and Design Goal

3.1 Problem Statement

Suppose that the cloud service provider has collected a large set of encrypted transactions from data owners. A miner, who has limited transactions, wants to mine the frequent itemsets. If mining from his own transaction database, the mining results may not be accurate. Therefore, he need make some queries to cloud to find out whether the itemsets in his own database are frequent or not in cloud's database which is much larger. We follow the same assumption in previous sections that each transaction is represented as a binary vector, and a mining query is represented as another binary vector.

3.2 System Model

In our system, we focus on preserving privacy association rule mining on the cloud. Specifically, we define the system model by dividing this system into five parties: Key Generation Center (KGC), Evaluator, Cloud Service Provider (CSP), Data Owners (DO) and Miner. The overall system model of our preserving privacy association rule mining system can be found in Fig. 1.

(1) **Key Generation Center:** The trusted KGC is responsible for generating and managing both public and private keys for every party in our system (See ①).

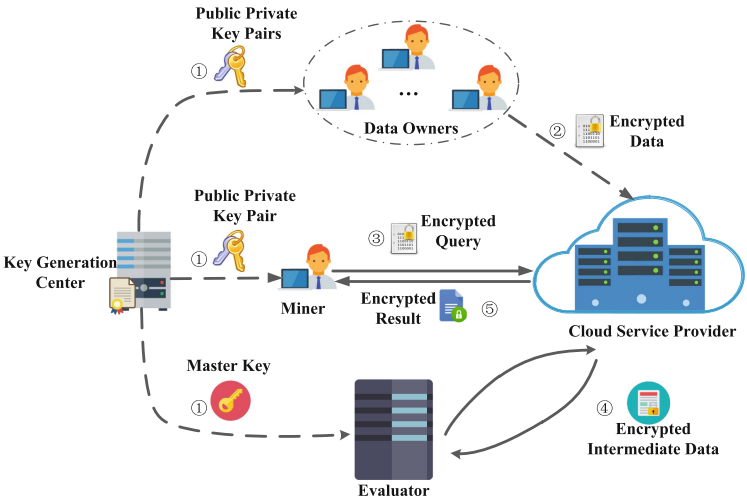


Fig. 1. System model

(2) **Data Owners:** Generally, the DOs use their public key to encrypt their sensitive data, before uploading them to the CSP (See ②).
(3) **Cloud Service Provider:** CSP has massive storage space. It could store and manage data outsourced from all the DOs (See ②). In addition, CSP has some computation abilities to perform some calculations over the outsourced data. In our system, the CSP provides the service of association rule mining for the miners through cooperating with Evaluator (See ④).
(4) **Evaluator:** Evaluator provides online computation in our system. It has the master key of the BCP cryptosystem. In our system, the CSP need cooperate with the Evaluator to mine the frequent itemsets and association rules (See ④).
(5) **Miner:** In our system, Miner is the data mining service user. Data owner can also be a miner. The miner has some transaction itemsets. The goal of the miner is to find the frequent itemsets and strong association rules for his limited dataset. To achieve this purpose, he sends the encrypted itemsets to the CSP to find out whether they are frequent or not (See ③). The mining results obtained from the CSP can only be decrypted by miner himself (See ⑤).

Note that the Evaluator is an essential part in our system. On one hand, since BCP cryptosystem is not fully homomorphic, a CSP alone cannot perform various compute operations. On the other hand, this twin-cloud architecture composed by CSP and Evaluator, can minimize the interactions between the request users and the cloud servers while the one cloud cannot [16]. In this scheme, the Miner only sends encrypted queries and then remains offline until receiving the encrypted mining results.

3.3 Threat Model

In our threat model, we assume the KGC is fully trusted by all the entities. On the other hand, CSP, Evaluator, DOs and Miner are *curious-but-honest* entities, which means that these entities intend to follow the protocols strictly and return correct computation results, but may try to infer the private information of other parties according to the data received and held. In addition, we also assume that the CSP and Evaluator don't conclude with each other. Now, we introduce an active adversary $\mathcal{A}$ in this model. The goal of $\mathcal{A}$ is to get the original data from the DOs and the Miner. What's more, $\mathcal{A}$ also wants to know the Miner's final mining results. Such an adversary has the following capabilities:

(1) $\mathcal{A}$ may eavesdrop all communication to obtain the encrypted data.
(2) $\mathcal{A}$ may compromise CSP and try to obtain all the plaintext value of the ciphertext uploaded by the DOs and all the intermediate results sent by Evaluator during the executing an interactive protocol.
(3) $\mathcal{A}$ may compromise one or more DOs to obtain their decryption abilities.

The adversary $\mathcal{A}$ is restricted from comprising (1) Evaluator, (2) all the DOs and (3) the Miner. Here we remark that such restrictions are typical and widely used in adversary model used in cryptographic protocols [1,16,17].

3.4 Design Goals

Under the aforementioned system model and attack model, our design goal is the following four objects.

(1) *The security and privacy should be guaranteed.* The data uploaded by the DOs, the query information from the Miner and the mining result from the encrypted data contains sensitive data of themselves which could not be disclosed to the CSP, Evaluator or $\mathcal{A}$. Meanwhile, the access pattern shouldn't be revealed and inferred by CSP, Evaluator or $\mathcal{A}$ either. Access pattern is defined as the original encrypted input corresponding to the computed value, e.g., the comparison result, the most frequent class label, etc.
(2) *Data query result's accuracy should be guaranteed.* It is also really important that the mining accuracy must be guaranteed when applying the privacy-preserving strategy. Therefore, the proposed system should achieve same accuracy compared with the non-privacy-preserving data mining system.
(3) *Low communication overhead and efficiency of computation should be guaranteed.* Consider the real-time requirements of online service and the diversity of terminals, the proposed scheme should have low overhead in terms of communication and computation. Especially, the DOs and the Miners in our system are usually resource-constrained users, their computation and communication cost should be as small as possible.
(4) *Offline DOs and miners should be supported.* After outsourcing the encrypted data, the DOs should be offline. There are many miners involved in our system. Therefore, supporting offline DOs and miners is rather necessary in terms of the system's scalability.

4 Privacy-Preserving Frequent Itemset Mining and Association Rule Mining

4.1 Setup

Recall that in Sect. 3 we have stated that the CSP holds a set of encrypted transactions from multiple DOs. Suppose we have η DOs in our system. The KGC generates pairs of the public and private keys (pk_{DO_i}, sk_{DO_i}), $i = 1, 2, \cdots, \eta$ and pk_M, sk_M. Then, KGC distributes the individual public-private key pair (pk_{DO_i}, sk_{DO_i}) to the DO i and (pk_M, sk_M) to the miner, respectively. Meanwhile, the strong private key is sent to the Evaluator. Moreover, all the entities' public keys are known to the others.

After receiving the public-private key pair from the KGC, the DOs encrypt every record $\boldsymbol{p_i}$ in his own database, and outsource these encrypted data to the CSP. So far, the work of the DOs' is over, meaning that all the DOs can remain offline from now on.

4.2 Privacy-Preserving Building Blocks

In this section, we propose a set of privacy-preserving building blocks, including secure multiplication accross domains algorithm, secure inner product calculation algorithm, secure comparison accross domains algorithm and secure comparison. In Andreas *et al* 's work [18], they have proposed **KeyProd** and **TransDec** algorithm in the similiar system model based on BCP. **KeyProd** and **TransDec** can be used to transform the encryptions under pk_{DO_i} or pk_M into encryption under $pk_\Sigma = \prod_{i=1}^{m} pk_{DO_i} pk_M$ or vice verse. For more details of these algorithms, please see [18]. These cryptographic blocks, proposed in this paper and Andreas *et al.*'s work [18], serve as the basic constructions of our privacy-preserving association rule mining system.

Secure Multiplication Across Domains. Note that Andreas *et al.* [18] have proposed a secure multiple protocol (i.e., **Mult**.) based on BCP cryptosystem. Here, we present the secure multiplication across different encryption domains with the similar idea. For simplicity and readability, we use $[\![x]\!]_{pk_{DO}}$ instead of $[\![x]\!]_{pk_{DO_i}}$ in the following context. Suppose that CSP has encrypted data $[\![x]\!]_{pk_{DO}}$ and $[\![y]\!]_{pk_M}$. The goal of secure multiplication across domains (**SMAD**) algorithm is to calculate $[\![xy]\!]_{pk_\Sigma}$. We introduce the details of our **SMAD** algorithm as follows.

Step 1 (CSP): (1) $a, b, c, d \xleftarrow{R} \mathbb{Z}_N$.
(2) $X_0 = [\![x]\!]_{pk_{DO}} \cdot [\![a]\!]_{pk_{DO}}$, $Y_0 = [\![y]\!]_{pk_M} \cdot [\![b]\!]_{pk_M}$, $X_1 = [\![x]\!]^b_{pk_{DO}} \cdot [\![c]\!]_{pk_{DO}}$, $Y_1 = [\![y]\!]^a_{pk_M} \cdot [\![d]\!]_{pk_M}$.
(3) Send X_0, Y_0, X_1 and Y_1 to Evaluator.
Step 2 (Evaluator): (1) $z_0 \leftarrow \mathbf{mDec}_{(pk_{DO}, MK)}(X_0)$, $z_1 \leftarrow \mathbf{mDec}_{(pk_M, MK)}(Y_0)$, $z_2 \leftarrow \mathbf{mDec}_{(pk_{DO}, MK)}(X_1)$, $z_3 \leftarrow \mathbf{mDec}_{(pk_M, MK)}(Y_1)$.

(2) $Z_1 \leftarrow [\![z_0 \cdot z_1]\!]_{pk_\Sigma}$, $Z_2 \leftarrow [\![z_2]\!]_{pk_\Sigma}^{N-1}$, $Z_3 \leftarrow [\![z_3]\!]_{pk_\Sigma}^{N-1}$.
(3) Send Z_1, Z_2, Z_3 to CSP.
Step 3 (CSP): (1) $S_1 \leftarrow ([\![a \cdot b]\!]_{pk_\Sigma})^{N-1}$, $S_2 \leftarrow [\![c]\!]_{pk_\Sigma}$, $S_3 \leftarrow [\![d]\!]_{pk_\Sigma}$.
(2) $[\![xy]\!]_{pk_\Sigma} \leftarrow Z_1 \cdot Z_2 \cdot Z_3 \cdot S_1 \cdot S_2 \cdot S_3$.

REMARK. The basic idea of **SMAD** is based on the following equation, i.e., $xy = (x + a)(y + b) - (bx + c) - (ay + d) - ab + (c + d)$.

Secure Inner Product. Suppose that CSP has an encrypted data vector $[\![\boldsymbol{x}]\!]_{pk_{DO}} = ([\![x_1]\!]_{pk_{DO}}, \cdots, [\![x_n]\!]_{pk_{DO}})$ and an encrypted data vector $[\![\boldsymbol{y}]\!]_{pk_M} = ([\![y_1]\!]_{pk_M}, \cdots, [\![y_n]\!]_{pk_M})$. For every $[\![x_i]\!]_{pk_{DO}}$ and $[\![y_i]\!]_{pk_M}$, CSP and Evaluator run **SMAD** algorithm to get $[\![x_i y_i]\!]_{pk_\Sigma}$. Then, CSP multiplies all the encrypted data. Thus, CSP can obtain $[\![\boldsymbol{x} \cdot \boldsymbol{y}]\!]_{pk_\Sigma} = (x_1 y_1 + \cdots + x_n y_n)_{pk_\Sigma}$.

Secure Comparison Across Domains. Suppose that CSP has two encrypted data $[\![x]\!]_{pk_M}$ and $[\![y]\!]_{pk_\Sigma}$, where where $x, y \leq 2^l$, $l < |N|/2 - 1$. The purpose of CSP is to find out whether $[\![x]\!]_{pk_M}$ is larger than $[\![y]\!]_{pk_\Sigma}$ or not, without leaking the original value of x and y to Evaluator.

Step 1 (CSP): (1) $A \leftarrow ([\![x]\!]_{pk_M})^2 \cdot [\![1]\!]_{pk_M}$, $B \leftarrow ([\![y]\!]_{pk_\Sigma})^2$.
(2) Randomly pick $a \xleftarrow{R} \{0, 1\}$, $C \leftarrow A^{a(N-1)}$, $D \leftarrow B^{(1-a)(N-1)}$.
(3) Randomly choose $r_a, r_b \xleftarrow{R} \mathbb{Z}_N$, and calculate $C' \leftarrow C \cdot [\![r_a]\!]_{pk_M}$, $D' \leftarrow D \cdot [\![r_b]\!]_{pk_\Sigma}$. Send C' and D' to Evaluator.
Step 2 (Evaluator): (1) $c' \leftarrow \mathbf{mDec}_{(pk_M, MK)}(C')$, $d' \leftarrow \mathbf{mDec}_{(pk_\Sigma, MK)}(D')$.
(2) Calculate $E \leftarrow [\![c' + d']\!]_{pk_\Sigma}$, then send E to CSP.
Step 3 (CSP): (1) $F \leftarrow E \cdot ([\![r_a + r_b]\!]_{pk_\Sigma})^{N-1}$.
(2) Randomly choose r_1, r_2, where $r_1, r_2 \xleftarrow{R} \{1, \cdots, 2^l\}$, $r_2 \ll r_1$, and calculate $F' \leftarrow F^{r_1} \cdot [\![r_2]\!]_{pk_\Sigma}$. Send F' to Evaluator.
Step 4 (Evaluator): (1) $z \leftarrow \mathbf{mDec}_{(pk_\Sigma, MK)}(F')$.
(2) If $z < N/2$, $\delta \leftarrow 1$ else $\delta \leftarrow 0$. Send $[\![\delta]\!]_{pk_\Sigma}$.
Step 5 (CSP): If $a = 0$, $[\![t]\!]_{pk_\Sigma} = [\![\delta]\!]_{pk_\Sigma}$. Else, $[\![t]\!]_{pk_\Sigma} \leftarrow [\![1]\!]_{pk_\Sigma} \cdot ([\![\delta]\!]_{pk_\Sigma})^{N-1}$.

Finally, CSP gets the encrypted comparison result $[\![t]\!]_{pk_M}$. If $t = 1$, it means $x \geq y$. Otherwise, it shows $x < y$.

Discussion. In the secure comparison algorithm of Qiu *et al* 's work [1], the CSP sends $[\![r(x-y)]\!]$ directly to Evaluator ($[\![x]\!]$ means the encryption of x under paillier [19]). There are several problems. First, if the decryption is 0, Evaluator could easily know $x = y$. Second, according to the decryption is smaller than $N/2$, the evaluator can infer whether x is smaller than y or not. Thus, we can conclude that, the comparison result is leaked to Evaluator in the secure comparison algorithm in Qiu *et al*'s work [1]. Moreover, if $x - y$ is a small number, the adversary $\mathcal{A}$ may infer the relationship of x and y according to the factoring result of $r(x-y)$, i.e., one large prime and a small number. Therefore, we can conclude the comparison algorithm in Qiu *et al.*'s work [1] is not secure to the adversary either. On one hand, in order to avoid showing the relationship of x and y, CSP should send $[\![r_1(x'-y')]\!]_{pk_\Sigma}$ or $[\![r_1(y'-x')]\!]_{pk_\Sigma}$ randomly, where

$x' = 2x + 1$ and $y' = 2y$. If $x > y$, it is obvious that $x' > y'$ or vice verse. On the other hand, to keep the comparison result from the factoring of $r_1(x' - y')$, CSP also blinds $r_1(x' - y')$ with a small random number r_2, i.e., $r_1(x' - y') + r_2$ before sending it to Evaluator. Since $r_2 \ll r_1$, blinding such a number dose not influence the comparison result of x and y.

Secure Comparison. We follow the same idea of **SCAD** to design the **SC** algorithm. Suppose that CSP has two encrypted data $[\![x]\!]_{pk_\Sigma}$ and $[\![y]\!]_{pk_\Sigma}$, where $x, y \leq 2^l$, $l < |N|/2 - 1$. The purpose of CSP is to find out whether $[\![x]\!]_{pk_\Sigma}$ is larger than $[\![y]\!]_{pk_\Sigma}$ or not, without leaking the original value of x and y to Evaluator. The details of the **SC** is as follows.

Step 1 (CSP): (1) Calculate $A \leftarrow ([\![x]\!]_{pk_\Sigma})^2 \cdot [\![1]\!]_{pk_\Sigma}$, $B \leftarrow ([\![y]\!]_{pk_\Sigma})^2$.
(2)Randomly pick $a \xleftarrow{R} \{0, 1\}$, $C \leftarrow A^{a(N-1)} \cdot B^{(1-a)(N-1)}$.
(3) Randomly choose r_1, r_2, where $r_1, r_2 \xleftarrow{R} \{1, \cdots, 2^l\}$, $r_2 \ll r_1$, and calculate $D \leftarrow C^{r_1} \cdot [\![r_2]\!]_{pk_\Sigma}$. Send D to Evaluator.
Step 2 (Evaluator): (1) $z \leftarrow \mathbf{mDec}_{(pk_\Sigma, MK)}(D)$.
(2) If $z < N/2$, $\delta \leftarrow 1$ else $\delta \leftarrow 0$. Send $[\![\delta]\!]_{pk_\Sigma}$.
Step 3 (CSP): If $a = 0$, $[\![t]\!]_{pk_\Sigma} \leftarrow [\![\delta]\!]_{pk_\Sigma}$. Else, $[\![t]\!]_{pk_\Sigma} \leftarrow [\![1]\!]_{pk_\Sigma} \cdot ([\![\delta]\!]_{pk_\Sigma})^{N-1}$.

At the end of the algorithm, CSP gets the encrypted comparison result, i.e., $[\![t]\!]_{pk_\Sigma}$. If $t = 1$, it means $x \geq y$. Otherwise, we can conclude $x < y$.

4.3 Secure Frequent Itemset Mining

CSP, Evaluator and Miner together run this secure frequent itemset mining algorithm. At the end of the algorithm, Miner gets the encrypted mining results. If the decrypted data is 1, it means that the query itemset is frequent. Otherwise, it is not. Assume that CSP holds m encrypted transactions data $\mathbf{C} = \{\boldsymbol{C_1}, \cdots, \boldsymbol{C_m}\}$, where $\boldsymbol{C_j} = ([\![c_{j,1}]\!]_{pk_{DO_i}}, \cdots, [\![c_{j,n}]\!]_{pk_{DO_i}})$, $i \in (1, \cdots, \eta)$, $j \in (1, \cdots, m)$. Miner has the encrypted mining request $\boldsymbol{Q}$ and $[\![z]\!]_{pk_M}$ as well as the encrypted minimum support $[\![supp_{min}]\!]_{pk_M}$, where $\boldsymbol{Q} = ([\![q_1]\!]_{pk_M}, \cdots, [\![q_n]\!]_{pk_M})$, z is the number of the 1s in $\boldsymbol{Q}$. Evaluator has the master key **MK**.

Step 1 (DO): Each DO encrypts his transactions with his own public key and sends the encrypted data to CSP. Thus, CSP gets m encrypted transactions data $\mathbf{C} = \{\boldsymbol{C_1}, \cdots, \boldsymbol{C_m}\}$, where $\boldsymbol{C_j} = ([\![c_{j,1}]\!]_{pk_{DO_i}}, \cdots, [\![c_{j,n}]\!]_{pk_{DO_i}})$, $i \in (1, \cdots, \eta)$, $j \in (1, \cdots, m)$.
Step 2 (Miner): The miner uses pk_M to encrypt his mining quest and minimum support, thus obtaining $\boldsymbol{Q}$, $[\![z]\!]_{pk_M}$ and $[\![supp_{min}]\!]_{pk_M}$, where $\boldsymbol{Q} = ([\![q_1]\!]_{pk_M}, \cdots, [\![q_n]\!]_{pk_M})$, z is the number of the 1s in $\boldsymbol{Q}$. Miner sends $\{\boldsymbol{Q}, [\![z]\!]_{pk_M}, [\![supp_{min}]\!]_{pk_M}\}$ to CSP.
Step 3 (CSP): CSP selects a *dummy* transactions set $\mathbf{D} = \{\boldsymbol{D_1}, \cdots, \boldsymbol{D_k}\}$, where $\boldsymbol{D_l} = (d_{l,1}, \cdots, d_{l,n})$, $d_{l,t} \in \{0, 1\}$, $l \in \{1, \cdots, k\}$ and $t \in \{1, \cdots, n\}$. CSP randomly chooses a DO's public key pk_{DO_i} to encrypt every D_l. Then, CSP

combines the transactions $\mathbf{C}$ uploaded by DOs with the dummy transaction set $\mathbf{D}$, which can be denoted as $\mathbf{E} = \mathbf{C} \bigcup \mathbf{D}$, and $E = \{\boldsymbol{E_1}, \cdots, \boldsymbol{E_k}\}$. Finally, CSP runs a secret permutation function on E, $E' = \pi(E)$.

Step 4 (CSP and Evaluator): CSP and Evaluator run **Keyprod** together on $[\![z]\!]_{pk_M}$ to get $[\![z]\!]_{pk_\Sigma}$. After that, CSP and Evaluator run **SIP** together on every transaction in the permuted database and miner's query. Thus, CSP gets $[\![x_i]\!]_{pk_\Sigma}, i \in (1, \cdots, m+k)$ at the end of every round of **SIP**.

Step 5 (CSP): For every $[\![x_i]\!]_{pk_\Sigma}$, CSP randomly chooses an α_i from $\mathbb{Z}_n$, and calculates $[\![w_i]\!]_{pk_\Sigma} \leftarrow \alpha_i([\![x_i]\!]_{pk_\Sigma} \cdot ([\![z]\!]_{pk_\Sigma})^{(N-1)})$. Then, CSP sends $\boldsymbol{W} = \{[\![w_1]\!]_{pk_\Sigma}, \cdots, [\![w_{m+k}]\!]_{pk_\Sigma}\}$ it to Evaluator.

Step 6 (Evaluator): Given $\boldsymbol{W}$, the Evaluator uses **MK** to decrypt every $[\![w_i]\!]_{pk_\Sigma}$. If $w_i = 0$, set $v_i = 1$, else $v_i = 0$. Then, he encrypts every v_i, before sending $\boldsymbol{V} = ([\![v_1]\!]_{pk_\Sigma}, \cdots, [\![v_{m+k}]\!]_{pk_\Sigma})$ to CSP.

Step 7 (CSP): On receiving $\boldsymbol{V'}$, CSP computes $\boldsymbol{V} = \pi^{-1}(\boldsymbol{V})$, then he removes the dummy results and calculates $[\![u]\!]_{pk_\Sigma} = \prod_{i=1}^{m} v'_i$.

Step 8 (CSP and Evaluator): CSP and Evaluator run **SCAD** together on $[\![supp_{min}]\!]_{pk_M}$ and $[\![u]\!]_{pk_\Sigma}$ and obtain the encrypted comparison result $[\![t]\!]_{pk_\Sigma}$. After that, CSP gets $[\![t]\!]_{pk_M}$ through running **TransDec** with Evaluator. CSP sends it to Miner.

Step 9 (Miner): Miner decrypts the $[\![t]\!]_{pk_M}$. If $t = 1$, the query itemset is frequent, else it is not.

Remark. In our **SFIM**, the dummy transactions are needed. Without the dummy transactions, Evaluator can deduce the support of q by counting the number of 0s in $\boldsymbol{W}$. With these dummy transactions, the support of q will be covered. Since, CSP knows the inverse of the permutation function, he can use it to remove the dummy results thus getting the original support of q.

Discussion. In Step 6 of our **SFIM**, Evaluator encrypts v_i by pk_Σ rather than pk_M. If using pk_M, in Step 8, CSP and Evaluator run **SC** instead of **SCAD**. However, the miner in our system is "honest-but-curious". If v_i is encrypted by pk_M, it could be leaked to Miner, which shouldn't be known to him. To protect DOs' data privacy, all the intermediate data should be encrypted by pk_Σ. For the reason that no one has private key of pk_Σ, only Evaluator is capable of decrypting the data encrypted by pk_Σ.

4.4 Secure Association Rule Mining

Getting frequent itemsets is not enough for Miner to figure out the relationship between the itemset. In the following context, we will describe how to securely mine association rule from the frequent itemsets. In our algorithm, the Miner is supposed to have the threshold of confidence, i.e., $conf_{min}$. If the Miner expects to know whether $X \Rightarrow Y$ is strong or not, CSP just needs to give him $supp(X)$ and $supp(X \cup Y)$. Assume that CSP has m encrypted transactions data $\mathbf{C} = \{\boldsymbol{C_1}, \cdots, \boldsymbol{C_m}\}$, where $\boldsymbol{C_j} = ([\![c_{j,1}]\!]_{pk_{DO_i}}, \cdots, [\![c_{j,n}]\!]_{pk_{DO_i}})$, $i \in (1, \cdots, \eta)$, $j \in$

$(1, \cdots, m)$. The CSP also has the support of query $[\![u]\!]_{pk_\Sigma}$ from **SFIM**. Miner has the frequent itemset $\boldsymbol{f}$ and the threshold of confidence α/β, where $\boldsymbol{f} = ([\![f_1]\!]_{pk_M}, \cdots, [\![f_n]\!]_{pk_M})$. Please note that, for the easiness and convenience of comparison, we denote the threshold of confidence as α/β. Evaluator has the master key **MK**. The details of our **SARM** is given as follows.

Step 1 (Miner): (1) Get the sets of $\boldsymbol{f}$'s nonvoid proper subset H, where $H = \{\boldsymbol{h_1}, \cdots, \boldsymbol{h}_{2^z-2}\}$ [1]. Suppose that the number of 1s in $\boldsymbol{h_i}$ is k_i.
(2) Encrypt every $\boldsymbol{h_i}$, k_i, α and β, where $i \in (1, \cdots, 2^z - 2)$. Send them to CSP.
Step 2: For each $i = 1$ to $2^z - 2$,
(CSP and Evaluator): (1) The same procedure as in **SFIM** from Step 3 to Step 7. At the end, CSP gets $[\![u_i]\!]_{pk_\Sigma}$.
(2) $[\![\tau_i]\!]_{pk_\Sigma} \leftarrow \mathbf{SMAD}([\![\beta]\!]_{pk_M}, [\![u]\!]_{pk_\Sigma})$, $[\![\varepsilon_i]\!]_{pk_\Sigma} \leftarrow \mathbf{SMAD}([\![\alpha]\!]_{pk_M}, [\![u_i]\!]_{pk_\Sigma})$.
(3) $[\![\gamma_i]\!]_{pk_M} \leftarrow \mathbf{SC}([\![\tau_i]\!]_{pk_\Sigma}, [\![\varepsilon_i]\!]_{pk_\Sigma})$. Send $[\![\gamma_i]\!]_{pk_M}$ to the miner.
Miner: (1) $\gamma_i \leftarrow \mathbf{Dec}_{sk_M}([\![\gamma_i]\!]_{pk_M})$.
(2) If $\gamma_i = 1$, If $\gamma_i = 1$, $\boldsymbol{h_i} \Rightarrow (\boldsymbol{f} - \boldsymbol{h_i})$ is a strong association rule. Else, it is not.

5 Security Analysis

5.1 Security of Cryptographic Blocks

In this section, we prove the security of **SMAD**, **SIP**, **SCAD**, and **SC**. First, we give the definition of security in the semi-honest model in [16,20].

Definition 3 (Security in the Semi-Honest Model [20]). *Let a_i be the input of party P_i, $\Pi_i(\pi)$ be P_i's execution image of the protocol π and b_i be the output for party P_i computed from π. Then π is secure if $\Pi_i(\pi)$ can be simulated from a_i and b_i such that distribution of the simulated image is computationally indistinguishable from $\Pi_i(\pi)$ (More details can be found in [20]).*

From **Definition** 3, we can conclude that the simulated execution image and the actual execution image should be computational indistinguishable when proving the security of these cryptographic blocks. In our scheme, the execution image generally includes the data exchanged and the information computed from these data.

Theorem 1. *The **SMAD** proposed is secure under semi-honest model.*

Proof. Here, let the execution image of Evaluator be denoted by $\Pi_{Evaluator}(SMAD)$ which is given by $\Pi_{Evaluator}(SMAD) = \{(X_0, z_0), (X_1, z_1), (Y_0, z_2), (Y_1, z_3)\}$ where $z_0 = x + a$, $z_1 = y + b$, $z_2 = bx + c$ and $z_3 = ay + d$

[1] For example, if $\boldsymbol{f} = \{1, 1, 1, 0\}$ which means $\{X, Y, Z\}$. The sets of $\boldsymbol{f}$'s nonvoid proper subset is $H = \{\{X\}, \{Y\}\{Z\}, \{X, Y\}, \{X, Z\}, \{Y, Z\}\}$, which can be represent as

$$H = \{\{1,0,0,0\}, \{0,1,0,0\}, \{0,0,0,1\}, \{1,1,0,0\}, \{1,0,1,0\}, \{0,1,1,0\}\}.$$

.

are derived by decrypting X_0, X_1, X_2 and X_3 respectively. Note that a, b, c, d are random numbers in $\mathbb{Z}_N$. We assume that $\Pi^S_{Evaluator}(SMAD) = \{(X'_0, z'_0), (X'_1, z'_1), (Y'_0, z'_2), (Y'_1, z'_3)\}$ where all the elements are randomly generated from $\mathbb{Z}_N$. Since BCP is a semantic secure encryption scheme, (X_i, z_i) is computationally indistinguishable from (X'_i, z'_i), $i \in (0, 1, 2, 3)$. Meanwhile, as every z'_i is randomly chosen from $\mathbb{Z}_N$, z_i is computationally indistinguishable from z_i. Based on the above analysis, we can draw a conclusion that $\Pi_{Evaluator}(SMAD)$ is indistinguishable from $\Pi^S_{Evaluator}(SMAD)$.

The proof of CSP is analogous to Evaluator. Combining the above analysis, we can confirm that **SMAD** is secure under the semi-honest model.

Theorem 2. *The **SIP** is secure under semi-honest model.*

Proof. Our **SIP** is based on **SMAD**. Since we have proven the security of **SMAD**, we can conclude that **SIP** is secure too.

Theorem 3. *The **SCAD** proposed is secure under semi-honest model.*

Proof. According to **SCAD**, the execution image of **SCAD** for Evaluator can be denoted by $\Pi_{Evaluator}(SCAD)$, which is $\Pi_{Evaluator}(SCAD) = \{(C', c'), (D', d'), (F', z), \delta\}$ where $c' = (-1)^a \cdot (2x + 1) + r_a$, $d' = (-1)^{1-a} \cdot (2y) + r_b$, $z = r_1((-1)^a \cdot (2x + 1) + (-1)^{1-a} \cdot (2y)) + r_2$ are separately derived from the decryption of C', D', F. Note that a is a random number from $(0, 1)$, r_a, r_b are random numbers form $\mathbb{Z}_N$, and r_1, r_2 is a random number from $\{1, \cdots, 2^l\}, 2^{2l+1} < N/2, r_1 \ll r_2$. In addition, δ is the comparison result from z. We assume $\Pi^S_{Evaluator}(SCAD) = \{(C'', c''), (D'', d''), (F'', z'), \delta'\}$ where (C'', c''), (D'', d''), (F'', z') are randomly generated from $\mathbb{Z}_N$, and δ' is set to 1 or 0 according to the randomly tossed coin. Since BCP is a semantically secure encryption scheme, (C', c'), (D', d'), (F', z) are computationally indistinguishable from (C'', c''), (D'', d''), (F'', z'). Furthermore, because the element a is randomly chosen from $\{0, 1\}$, δ is either 0 or 1 with equal probability. Thus, δ is computationally indistinguishable from δ'. Combining the above results, we can claim that $\Pi_{Evaluator}(SCAD)$ is computationally indistinguishable from $\Pi^S_{Evaluator}(SCAD)$.

On the other hand, the execution image of CSP, denoted by $\Pi_{CSP}(SCAD)$, is given by $\Pi_{Evaluator}(SCAD) = \{E, [\![\delta]\!]_{pk_\Sigma}\}$. Let the simulated image of CSP be given by $\Pi^S_{Evaluator}(SCAD) = \{E', \alpha\}$, where E', α are random numbers from $\mathbb{Z}_N$. Since BCP is semantically secure encryption scheme, E, and $[\![\delta]\!]_{pk_\Sigma}$ are computationally indistinguishable from E', and α. Thus, we can conclude that $\Pi_{CSP}(SCAD)$ is computationally indistinguishable from $\Pi^S_{CSP}(SCAD)$.

Based on the above analysis, we can claim that **SCAD** is secure under the semi-honest model.

Theorem 4. *The **SC** described is secure under semi-honest model.*

Proof. Since **SC** is designed by the similar idea of **SCAD**, we can easily get the proof from Theorem 3.

5.2 Security of SFIM and SARM

Theorem 5. *The **SFIM** proposed is secure under semi-honest model and also can preserve the data confidentiality and query privacy against active adversary.*

Proof. In the similar maner we can prove that our **SFIM** is secure under the semi-honest model firstly. In Step 1 to Step 2, DOs and Miner send C and Q, $[\![z]\!]_{pk_M}$, $[\![supp_{min}]\!]_{pk_M}$ to CSP. Due to the semantic security of BCP, the semi-honest CSP has no advantage to distinguish them from random numbers from $\mathbb{Z}_N$. In Step 3, the CSP randomly chooses a dummy transactions set and encrypts it with a random public key from DOs. Then, he mixes it with the original dataset uploaded from DOs. After that, CSP and Evaluator run the **SIP**. Since the Evaluator cannot distinguish the original dataset and the dummy data and the security proof of **SIP**, we can confirm the protocol is secure in Step 3 and Step 4. Furthermore, the data operation in Step 5 to Step 7 is similar to the process of **SMAD**, all the exchanged messages are in encrypted format, and each value deduced by CSP and Evaluator is blinded by random numbers. In Step 8, the **SCAD**, **TransDec** are adopted as the fundamental building blocks, which has been proved secure in previous section and [18]. In Step 9, CSP and Miner just deal with encrypted data, the security is from the semantic security of BCP. As a result, we can easily conclude that our **SFIM** is secure under the semi-honest model.

Next, we discuss the data confidentiality and query privacy against an active adversary $\mathcal{A}$. Assume that $\mathcal{A}$ eavesdrops the transmission link between DOs and CSP, the encrypted database and all the intermediate data is got by $\mathcal{A}$. Because all the data is encrypted by BCP, $\mathcal{A}$ cannot get the original data. If $\mathcal{A}$ comprises some DOs and gets their private keys, they still cannot decrypt the Miner's query since the encryption key is different. As long as the evaluator is not comprised all the data confidentiality and query privacy defined is satisfied.

As a result, we can claim that our **SFIM** is secure under semi-honest model and also can preserve the data confidentiality and query privacy against active adversary.

Theorem 6. *The **SARM** described in Sect. 4.4 is secure under semi-honest model and also can preserve the data confidentiality and query privacy against active adversary.*

It is worth noting that the proofs are similar to Theorem 5 and hence we omit it due to the space limitation.

6 Performance Analysis

In this section, we evaluate the performance of our scheme. In [10], the author also proposed a variant of the original BCP cryptosystem, where the randomness r is chosen in a smaller set, namely in $\mathbb{Z}_N$ rather than $\mathbb{Z}_{N^2}$. The variant of the original BCP cryptosystem is secure based on the *Small Decisional Diffie-Hellman Assumption* (S-DDH) over a squared composite modulus of the form

$N = pq$. (More details of S-DDH and the security analysis can be found in [10]) In this section, we will analyse the performance of our system based on BCP and the variant of BCP.

6.1 Experiment Analysis

The performance evaluations of the proposed system are tested on five laptop computers running Windows 8.1 with Intel Core I5-5200U 2.20 GHz CPU and 4 GB RAM. We implement BCP and its variant cryptosystem by BigInteger Class in Java development kit, and using this to implement our computation protocols. Specially, two of them are acted as the DOs, which encrypt the data and upload them to CSP; one is used as the Miner, and the rest of them are leveraged as the CSP and Evaluator respectively. In our experiment, we first test the efficiency of our cryptographic blocks. Then, we make an efficiency comparison with the most recent work [1] over the same chess database[2] as our transaction dataset, which totally has 3196 transactions and 75 attributes. Moreover, we analyse the performance of the schemes by varying parameters.

Table 4. Performance of cryptographic blocks (100-times for average, 80-bits security level)

Algorithm	CSP Compute.	Evaluator Compute.	CSP Commu.	Evaluator Commu.
SMAD	0.391 s	0.368 s	1.998 KB	1.499 KB
SCAD	0.398	0.214 s	1.498 KB	0.999 KB
SC	0.137	0.098 s	0.498 KB	0.499 KB
SIP (10 bits Vector)	3.951 s	3.822 s	19.991 KB	14.991 KB

Efficiency of Cryptographic Blocks. We first evaluate the performance of the basic cryptographic blocks, which can be seen in Table 4. For the BCP algorithm, we denote N as 1024 bits to achieve 80-bit security [21] levels. We can observe from Table 4 that in the **SMAD** algorithm the computation of CSP costs 0.391 s and he sends 1.998 KB data when communicating with Evaluator, while Evaluator needs 0.368 s to complete the computation and the communication will cost 1.499 KB. Moreover, in the **SCAD** algorithm, the CSP needs 0.398 s to compute and send 1.498 KB data to Evaluator, while the Evaluator needs 0.214 s to compute and send 0.999 KB data. In the **SC** algorithm, the CSP costs 0.137 s for computing and sends 0.498 KB data to Evaluator, while the Evaluator needs 0.098 s to compute and send 0.499 KB data. We also test **SIP** over two 10-bit vectors, we can see from Table 4, the cost of CSP and Evaluator is almost ten times of single **SMAD**.

We also test our scheme based on the variant of the BCP cryptosystem. The running result can be found in Table 5.

[2] http://fimi.ua.ac.be/data/.

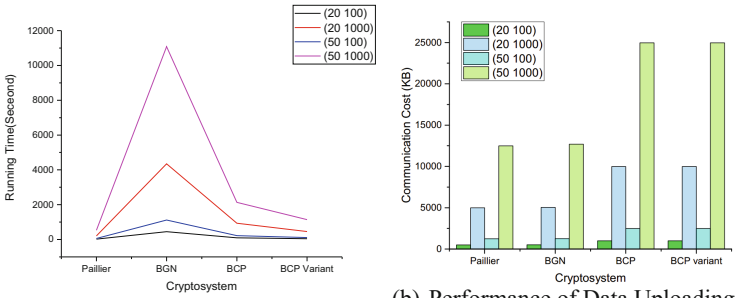
Table 5. Performance of cryptographic blocks based on the variant BCP (100-times for average, 80-bits security level)

Algorithm	CSP Compute.	Evaluator Compute.	CSP Commu.	Evaluator Commu.
SMAD	0.297 s	0.251 s	1.998 KB	1.498 KB
SCAD	0.254	0.171 s	1.499 KB	0.999 KB
SC	0.083	0.063 s	0.499 KB	0.499 KB
SIP (10 bits Vector)	2.301 s	3.102 s	19.981 KB	14.989 KB

Efficiency Comparison. For a fair comparison, we also implement Qiu *et al.*'s work [1] in Java by BigInteger Class in Java development kit and JPBC library[3]. We choose $| p |= 160$ bits with at least 80-bit security with Type A pairing in BGN and N as 1024 bits in Paillier [19]. We first make a comparison about the data encryption and uploading and then the frequent itemset mining protocol is compared.

Performance of Data Encryption and Uploading. Note that the data encryption is done in off-line by the DOs. In most conditions, the DOs are resource-constrained users. The performance of data encryption is shown in Fig. 2(a) and the uploading communication costs are shown Fig. 2(b).

As shown in Fig. 2, the running time of data encryption by BCP is much less than BGN, and the BCP variant's is more less, while both of them are higher than the Paillier's running time. The communication cost of BCP and BCP varinat is almost same which is larger than Paillier and BGN. Since most of the DOs are resource-constrained, our scheme extensively reduce the DOs' computation cost than [1]'s protocol 2, but with slight higher communication cost.



(a) Performance of Data Encryption

(b) Performance of Data Uploading

Fig. 2. Performance of data owner

[3] http://gas.unisa.it/projects/jpbc.

Table 6. Cloud computation time (in minutes) of frequent itemset mining

Protocol 2	Our protocol based on BCP	Our protocol based on BCP variant
1354.021	4321.612	2930.398

Performance of Frequent Itemset Mining. We test the cloud's (including CSP and Evaluator) running time in our scheme and [1]'s protocol on the Chess dataset. The overall running time is shown in Table 6. In our experiment, the size of dummy transactions in all of the protocols is $m/2$. From Table 6, we can conclude that our protocol is slower than [1]'s protocol 2. Since our protocol achieves higher privacy level, we think it is reasonable. In addition, if we use the BCP variant as the basic cryptosystem in our scheme, the running time can be largely reduced. What's more, the cloud is usually has "unlimited" computing resource and power, the running time of our scheme can be dramatically reduced in real cloud system.

7 Conclusions

In this paper, we propose a practical privacy-preserving frequent itemset mining and association rule mining protocol on encrypted cloud data. Compared with the state-of-art works, our scheme achieves higher privacy level, and also reduces the data owners' computation cost. The computation cost in cloud is higher than Qiu *et al.*'s work [1]. Since the cloud has massive computation resource, the computation time in real cloud service will be quite small. In our future work, we will focus on further improving the efficiency of our scheme.

Acknowledgement. The authors would like to thank Dr. Shuo Qiu for her generous feedback. The work is supported by the National Natural Science Foundation of China (No. 61702541, No. 61702105), the Young Elite Scientists Sponsorship Program by CAST (2017QNRC001), the Science and Technology Research Plan Program by NUDT (Grant No. ZK17-03-46), the national key research and development program under grant 2017YFB0802301, and Guangxi cloud computing and large data Collaborative Innovation Center Project.

References

1. Qiu, S., Wang, B., Li, M., Liu, J., Shi, Y.: Toward practical privacy-preserving frequent itemset mining on encrypted cloud data. IEEE Trans. Cloud Comput. (2017)
2. Sarawagi, S., Nagaralu, S.H.: Data mining models as services on the internet. ACM SIGKDD Explor. Newslett. **2**(1), 24–28 (2000)
3. Giannotti, F., Lakshmanan, L.V.S., Monreale, A., Pedreschi, D., Wang, H.: Privacy-preserving mining of association rules from outsourced transaction databases. IEEE Syst. J. **7**(3), 385–395 (2013)

4. Yi, X., Rao, F.Y., Bertino, E., Bouguettaya, A.: Privacy-preserving association rule mining in cloud computing. In: ACM Symposium on Information, Computer and Communications Security, pp. 439–450 (2015)
5. Tai, C.-H., Yu, P.S., Chen, M.-S.: k-Support anonymity based on pseudo taxonomy for outsourcing of frequent itemset mining. In: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 473–482. ACM (2010)
6. Kantarcioglu, M., Clifton, C.: Privacy-preserving distributed mining of association rules on horizontally partitioned data. IEEE Trans. Knowl. Data Eng. **16**(9), 1026–1037 (2004)
7. Vaidya, J., Clifton, C.: Privacy preserving association rule mining in vertically partitioned data. In: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 639–644. ACM (2002)
8. ElGamal, T.: A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE Trans. Inf. Theory **31**(4), 469–472 (1985)
9. Li, L., Lu, R., Choo, K.-K.R., Datta, A., Shao, J.: Privacy-preserving-outsourced association rule mining on vertically partitioned databases. IEEE Trans. Inf. Forensics Secur. **11**(8), 1847–1861 (2016)
10. Bresson, E., Catalano, D., Pointcheval, D.: A simple public-key cryptosystem with a double trapdoor decryption mechanism and its applications. In: Laih, C.-S. (ed.) ASIACRYPT 2003. LNCS, vol. 2894, pp. 37–54. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-40061-5_3
11. Dong, C., Chen, L.: A fast secure dot product protocol with application to privacy preserving association rule mining. In: Tseng, V.S., Ho, T.B., Zhou, Z.-H., Chen, A.L.P., Kao, H.-Y. (eds.) PAKDD 2014. LNCS (LNAI), vol. 8443, pp. 606–617. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-06608-0_50
12. Dong, C., Chen, L., Wen, Z.: When private set intersection meets big data: an efficient and scalable protocol. In: Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security, pp. 789–800. ACM (2013)
13. Lai, J., Li, Y., Deng, R.H., Weng, J., Guan, C., Yan, Q.: Towards semantically secure outsourcing of association rule mining on categorical data. Inf. Sci. **267**(2), 267–286 (2014)
14. Boneh, D., Goh, E.-J., Nissim, K.: Evaluating 2-DNF formulas on ciphertexts. In: Kilian, J. (ed.) TCC 2005. LNCS, vol. 3378, pp. 325–341. Springer, Heidelberg (2005). https://doi.org/10.1007/978-3-540-30576-7_18
15. Agrawal, R., Imieliński, T., Swami, A.: Mining association rules between sets of items in large databases. In: ACM SIGMOD Record, vol. 22, pp. 207–216. ACM (1993)
16. Cheng, K., Wang, L., Shen, Y., Wang, H., Wang, Y., Jiang, X., Zhong, H.: Secure k-NN query on encrypted cloud data with multiple keys. IEEE Trans. Big Data (2017)
17. Liu, X., Deng, R.H., Choo, K.-K.R., Weng, J.: An efficient privacy-preserving outsourced calculation toolkit with multiple keys. IEEE Trans. Inf. Forensics Secur. **11**(11), 2401–2414 (2016)
18. Peter, A., Tews, E., Katzenbeisser, S.: Efficiently outsourcing multiparty computation under multiple keys. IEEE Trans. Inf. Forensics Secur. **8**(12), 2046–2058 (2013)

19. Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 223–238. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48910-X_16
20. Goldreich, O.: Foundations of Cryptography: Volume 2, Basic Applications. Cambridge University Press, Cambridge (2009)
21. Barker, E., Barker, W., Burr, W., Polk, W., Smid, M.: NIST special publication 800–57. NIST Spec. Publ. **800**(57), 1–142 (2007)