Wireless Networks
https://doi.org/10.1007/s11276-022-03185-5

ORIGINAL PAPER

A new method for privacy preserving association rule mining using homomorphic encryption with a secure communication protocol

S. Zehtabchi[1] · N. Daneshpour[1] · M. Safkhani[1,2]

Accepted: 26 October 2022

Abstract

With the enormous amount of data growing exponentially, data owners aim to share data with each other to acquire an enhanced analytic view of their data. Distributed association rule mining over cloud computation helps data owners to extract knowledge from multiple databases. However, privacy is an important concept for data owners who want to share data and extract knowledge from aggregated data. This study proposes an outsourcing method to securely share and mine association rules from multiple parties protecting all parties' privacy. In this research, we utilize the properties of homomorphic encryption and propose a custom secure communication protocol. In our study, privacy of the shared data of all parties is ensured and we will not perform decryption in any step of the proposed method. We show that previous schemes could not properly guarantee the integrity of messages, furthermore, we propose our innovative communication protocol that uses an integrity checking function based on homomorphic encryption to authorize each party's shared data. We have implemented our scheme and showed even though an additional communication burden exists in our method, this method outperforms its competitors in terms of elapsed time and security. We have achieved more security especially in confidentiality and integrity of messages due to use of our proposed secure communication protocol. We also measured our proposed method's precision of mining parameters with different datasets to show the accuracy of our method on different scales will not change.

Keywords Data mining · Privacy · Rule mining · Security

1 Introduction

Data mining is being widely used for the purpose of finding interesting patterns from the usual large-scaled databases. Data owners take advantage of analyses from their data using properties of data mining techniques. However, as the data become more valuable, data owners tend to aggregate their data with each other to have a better perception of the analyzed aggregated data. On the other hand, the privacy of every data owner is an obstacle for them to share data.

Privacy-Preserving Data Mining (PPDM) refers to the methods which help data owners to discover interesting patterns in their data without privacy violation. The Secure Multiparty Computation (SMC) [1] can be used as a solution in response to these concerns. SMC-based privacy-preserving data mining algorithms can securely outsource computation from clients to a cloud system and take back the result to each user. Traditional SMC methods are not acceptable due to inefficient performance with large-scaled data analytics [2]. Hence, data analyzers need a new method to reduce the problem of the high computational burden of the traditional SMC methods.

PPDM methods are different from each other based on the method they use to achieve different levels of privacy. Some of the methods use encryption to make

✉ N. Daneshpour
ndaneshpour@sru.ac.ir

S. Zehtabchi
s.zehtabchi@sru.ac.ir

M. Safkhani
safkhani@sru.ac.ir

[1] Faculty of Computer Engineering, Shahid Rajaee Teacher Training University, Tehran, Iran

[2] School of Computer Science, Institute for Research in Fundamental Sciences (IPM), Tehran, P.O. Box 19395-5746, Iran

Published online: 21 November 2022

confidentiality during data transfer. Homomorphic encryption is also proposed in some papers to calculate the result with encrypted data as input. Hence, the data mining process has no knowledge of the data as it is being mined. Data owners tend to get knowledge of all the data from all participants without revealing their private data and knowledge, therefore, PPDM schemes that use homomorphic cryptography systems can be acceptable by data owners.

Association rule mining is a data mining method for discovering interesting relations between variables in large datasets. Association rules are if-then statements that help to show the probability of relationships between data items within large datasets in various types of datasets. Association rule mining has a wide variety of applications in market basket analysis, social healthcare, anomaly detection and etc. This paper proposes a new method for privacy-preserving association rule mining on distributed databases using cloud computation.

Nowadays, privacy-preserving association rule mining has gained lots of attention by researchers. In [3] Hammai et al. proposed a method for association rule mining using homomorphic encryption. In [4] Li et al. proposed a method for privacy-preserving association rule mining over transactional datasets. In [5] a method for association rule mining using homomorphic encryption has been investigated. However, despite other schemes that do not provide the integrity of messages using a secure communication protocol, we propose a new method for this aim. By using our secure communication protocol, when a replay attack occurs, all of the received messages would not be acceptable, since our proposed Integrity Checking Function (ICF) whose aim is to check the secure value of all the messages, would detect it and will ignore all of the incoming messages from that sender. Replay attacks can lead data mining process to wrong knowledge.

The problem of PPDM always faces various challenges against privacy, efficiency and feasibility. The main aim of our study is to propose a new method for solving the problem of association rule mining over distributed datasets that enhances privacy whilst keeping the performance acceptable and preserving all participants' privacy. Furthermore, the method must be feasible for all participants in the case of communication protocol and computational costs. On the other hand, despite most of our competitors, we aim to propose a method that will not be dependent on the number of participants and will be feasible for multiple participants to mine knowledge from their aggregated data.

Data owners' privacy is an important factor for them, but they also need to be assured that the extracted knowledge after the data-mining operation has the most available precision as it is the main benefit of it. We show our method does not change or remove any data to achieve the purpose of privacy, hence we prevent losing useful knowledge.

In the following sections, we propose a new method for sharing data which is based on a homomorphic cryptosystem. This method uses our proposed communication scheme for authentication and ensuring data integrity and confidentiality so that all parties can ensure the integrity and accessibility of their data. The proposed data sharing scheme is not dependent on any data-mining scheme, hence it can be used for any cloud-based data mining system. A method is proposed to securely mine association rules over distributed databases using the properties of homomorphic encryption for privacy-preserving data mining based on SMC. The proposed method for mining association rules is based on homomorphic encryption so none of the parties can get information about other parties local databases, however, the final results of mining association rules will be shared with all parties. Therefore, this paper aims to make a contribution to helping all parties securely perform association rule mining using new integrity checking function that is based on homomorphic encryption.

1.1 Main contributions

The main contributions of this paper are summarized as follows:

- Proposing a communication protocol for distributed association rule mining using properties of homomorphic encryption and investigating how this method can avoid replay attacks by checking the integrity of received messages.
- Proposing a new method for privacy preserving association rule mining over horizontally distributed datasets based on the method proposed in [3], which uses our proposed communication scheme.
- Implementing our proposed scheme and also some of recent related schemes for comparison.

1.2 Paper organization

The organization of this paper is summarized as follows. We present backgrounds for privacy preserving association rule mining over distributed datasets in Sect. 2. Then, we have a review of previous works on privacy preserving data mining in Sect. 3. In Sect. 4, we propose our method for privacy preserving association rule mining. Sect. 5 focuses on analyzing the results of our proposed method compared to other related methods. In Sect. 6, we have concluded the proposed method and the following results.

2 Background and definitions

In the following section, we look at some basic concepts of privacy preserving data mining.

2.1 Data partitioning

Outsourced data mining techniques can be categorized based on the method of data partitioning between parties. We have three different database fragmentation methods which are explained as follows:

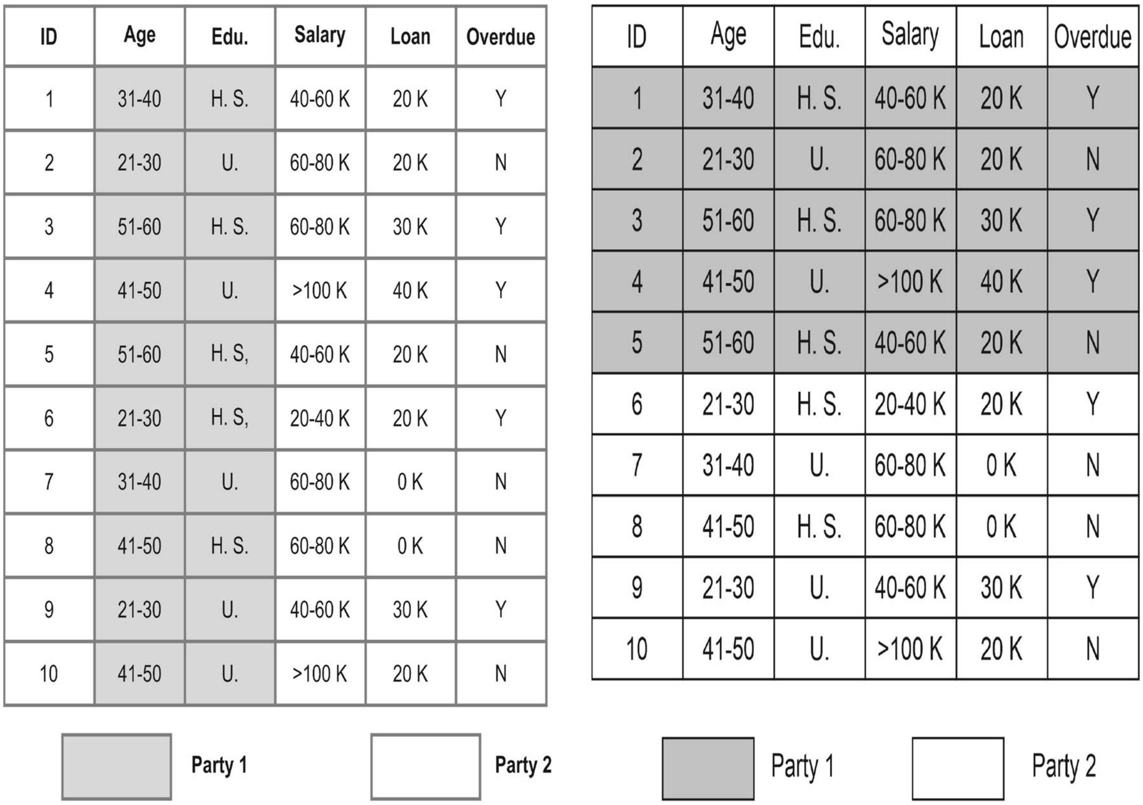
- Vertically data partitioning. All the parties have the same records, however, attributes about those records are saved exclusively. Figure 1a illustrates a partitioned database consists of vertically partitioned datasets.
- Horizontally data partitioning. In this method, all the parties have same attributes but every party has different records. Figure 1b illustrates a partitioned database consists of vertically partitioned datasets.
- Hybrid partitioning. This method is a combination of vertically and horizontally data partitioning. In the hybrid partitioning, data are distributed either first vertically and then horizontally or vice-versa.

In this paper, we propose a scheme for privacy-preserving data mining based on horizontally data partitioning. Our scheme solves the problem of privacy-preserving association rule mining where all the participants have the same database structure and possibly different records.

2.2 Association rule mining

As mentioned before, association rule mining is a data mining technique to extract interesting associations between attributes in large databases. This is used in a variety of applications like anomaly detection, medical diagnosis, finding patterns in data and market basket analysis.

Association rule mining is defined in [6]. Assume $A = \{a_1, a_2, a_3, ..., a_n\}$ is a set of binary-value attributes of size n. $DB = \{t_1, t_2, t_3, ..., t_m\}$ is a set of transactions of size m. Each transaction t is called an itemset if $t \subset A$. For an itemset $P \subseteq A$, a transaction t contains P if and only if $P \subseteq t$. An association rule is an implication $P \Rightarrow Q$ where $P \subseteq A$, $Q \subseteq A$ and $P \cap Q = \theta$. The support value of an association rule $P \Rightarrow Q$ can be derived as shown in Definition 1. This rule has support value S if the probability of a transaction in a database DB containing both P and Q is



ID	Age	Edu.	Salary	Loan	Overdue
1	31-40	H. S.	40-60 K	20 K	Y
2	21-30	U.	60-80 K	20 K	N
3	51-60	H. S.	60-80 K	30 K	Y
4	41-50	U.	>100 K	40 K	Y
5	51-60	H. S,	40-60 K	20 K	N
6	21-30	H. S,	20-40 K	20 K	Y
7	31-40	U.	60-80 K	0 K	N
8	41-50	H. S.	60-80 K	0 K	N
9	21-30	U.	40-60 K	30 K	Y
10	41-50	U.	>100 K	20 K	N

▨ Party 1 □ Party 2

(a) Vertically distributed dataset

ID	Age	Edu.	Salary	Loan	Overdue
1	31-40	H. S.	40-60 K	20 K	Y
2	21-30	U.	60-80 K	20 K	N
3	51-60	H. S.	60-80 K	30 K	Y
4	41-50	U.	>100 K	40 K	Y
5	51-60	H. S.	40-60 K	20 K	N
6	21-30	H. S.	20-40 K	20 K	Y
7	31-40	U.	60-80 K	0 K	N
8	41-50	H. S.	60-80 K	0 K	N
9	21-30	U.	40-60 K	30 K	Y
10	41-50	U.	>100 K	20 K	N

▨ Party 1 □ Party 2

(b) Horizontally distributed dataset

Fig. 1 Vertically distributed dataset and Horizontally distributed dataset

S. The confidence of this rule is C if the probability of a transaction in database DB containing P and then Q is C as shown in Definition 2.

Definition 1 **Support** $(P \rightarrow Q) = \frac{|P \cup Q|}{|DB|}$ where $P \subseteq A$, $Q \subseteq A$ and A is a set of n binary values attributes.

Definition 2 **Confidence** $(P \rightarrow Q) = \frac{|P \cup Q|}{|P|}$ where $P \subseteq A$, $Q \subseteq A$ and A is a set of n binary values attributes.

If an itemset's support value is greater than or equal to the user-defined minimum support threshold s, then it is called a frequent itemset. Association rule mining works in two steps, based on Apriori algorithm [7].

As our proposed method is based on Secure Multiparty Computation (SMC), we explain SMC in the next section.

2.3 Secure multiparty computation

Mostly, the main problem with data sharing is about data owners privacy concerns about their sensitive data. Secure Multiparty Computation (SMC) is a subfield of cryptography that allows multiple participants to outsource their data without any privacy leakage. The main concept of SMC is to compute a function $F(i_1, i_2, ..., i_N)$ where the inputs $i_1...i_N$ are taken from N parties and will be kept as secrets. Figure 2 illustrates the concept of SMC.

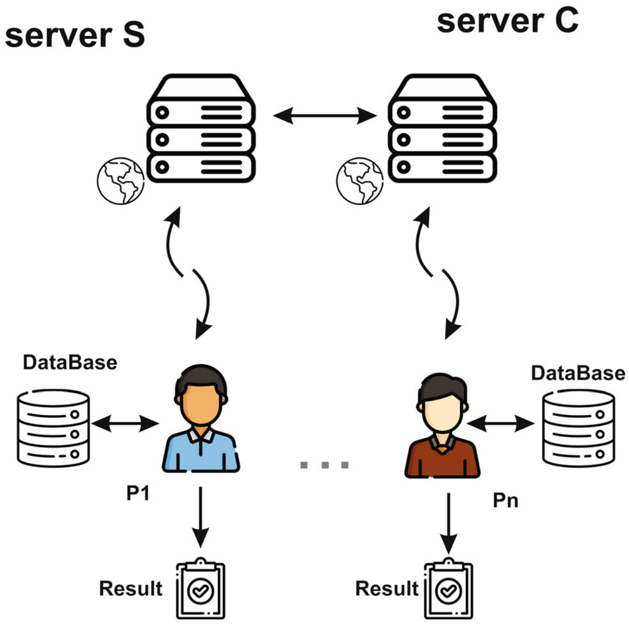


Fig. 2 The concept of secure multiparty computation

To achieve the goal of SMC; our proposed scheme, uses homomorphic encryption which satisfies the goal of privacy during all the steps of computation.

2.4 Homomorphic encryption

Homomorphic encryption is a form of encryption with an additional evaluation capability for computing over encrypted data without access to the secret key. The result of such a computation remains encrypted. Definition 3 defines the main property of homomorphic encryption.

Definition 3 $Enc_{pk}(m1 + m2) = Enc_{pk}(m1) \cdot Enc_{pk}(m2)$ $Dec_{sk}(Enc_{pk}(m1) \cdot Enc_{pk}(m2)) = Dec_{sk}(Enc_{pk}(m1 + m2)) = m1 + m2$ where pk and sk are the public key and the secret key of legal party in the protocol.

3 Related work

Recently, privacy-preserving data mining techniques for extracting knowledge from securely shared sensitive data without revealing sensitive data have been proposed in a variety of techniques. These methods are classified into four major categories based on how they share sensitive data: **data anonymization**, **data randomization**, **data perturbation**, and **cryptography-based** methods. The following are the properties of these methods:

- **Data anonymization** Data identifiers may be removed or changed in data anonymization techniques. Personal identifiers may be converted into aggregated data, rendering sensitive data owners indistinguishable. Matatov et al. [8] employs a genetic algorithm to find the best features for achieving the highest level of k-anonymity. Prakash and Singaravel [9] proposes a method for achieving the required level of k-anonymity by dividing and partitioning the original dataset.
- **Data randomization** In these methods, the original data is distorted to introduce some noise into the data. Using an adapted binary firefly algorithm, sensitive knowledge is changed before data sharing in [10]. Wu et al. in [11] proposed a scheme to ensemble locally created random decision trees (RDT) that satisfies local differential privacy on edge servers and utilized their scheme efficiency by using pruning and merging techniques. In [12], Shen et al. presented a survey over the schemes proposed for statistical learning under local differential privacy.
- **Data perturbation** In the data perturbation method, original data will be modified to achieve the aim of privacy. The knowledge extracted from perturbed data

has to be similar to the original one. Reversible perturbation methods are used to make secure data communication protocol between parties. In [13, 14], image steganography schemes are used to securely share sensitive data and mine knowledge from shared data. Du et al. in [15] proposed two different algorithms, Output Perturbation and Objective Perturbation, to increase privacy of training datasets in wireless big data scenarios. Wu et al. [16] also utilized multiple special data structures called Sketch to design various locally private frequency estimation schemes of physical symptoms for infectious disease analysis. Zhan et al. in [17] proposed a Continuous Reversible Privacy-Preserving (CRP) algorithm which has two phases for data hiding and data recovery. It also solves the ID3 decision tree problem for vertically partitioned datasets between n parties.

- **Cryptography-based** These methods are designed for achieving high levels of privacy during data sharing and computing results. In this paper, we concentrate on the schemes based on cryptography. In [18], Li et al. proposed a new method for generating a random decision tree using BCP homomorphic encryption algorithm. Also in another work, Li et al. [19], constructed the C4.5 decision tree using BCP homomorphic encryption. In [5], a new method has been proposed for mining association rules over distributed parties using homomorphic encryption. Li et al. in [4], proposed a method for mining association rules over transactional datasets using custom homomorphic encryption. It adds some fictitious transactions to provide data noise during data communication and detects fictitious transactions before computation. In [20], Domadiya and Reo proposed a method that is based on a custom encryption algorithm for performing association rule mining over horizontally distributed datasets. Bahrami et al. in [21] proposed a cryptographic approach that uses Pseudo-Random Permutation method for mobile clients to store data on one or multiple clouds. Vaidya et al. in [22] used homomorphic encryption for proposing a scheme for k-means clustering working on vertically datasets with data perturbation. In [23] Tan et al. proposed a scheme to implement a lightweight edge-based KNN classification mechanism over encrypted cloud data using Paillier encryption. In [24], a new scheme for constructing ID3 decision tree algorithm over horizontally distributed datasets and cloud systems is proposed based on EPOM homomorphic encryption. Li et al. in [25] used homomorphic encryption for computing $xLnx$ to construct an ID3 decision tree for two parties. Huang et al. in [26] used AES-256 encryption for mining association rules. They also used timestamps for authentication of the received messages. Hammani et al. in [3] proposed a method for association rule mining using homomorphic encryption. In [27] Domadiya et al. proposed a new method for privacy-preserving association rule mining over horizontally partitioned datasets using Shamir's secret. Rajesh et al. proposed a method based on a revised RSA and neural network in [28] which exchanges the dataset using revised RSA and then performs a feed-forward backpropagation neural network. In this paper, we propose a new Integrity Checking Function (ICF) method in order to secure communication protocol between our parties.

Table 1 shows the discussed methods based on the proposed data mining schemes and their security preserving category, and Fig. 3 is a schematic of the above-mentioned schemes' categorization.

4 Proposed method

In this section, we will first explain some preliminary steps required for our proposed method. Then, we propose a protocol for communication between users and cloud computing services. Our proposed method consists of a protocol that allows parties to communicate with one another. As a result, before being accepted by the receiver, each message will be authenticated. The properties of homomorphic encryption are used in our communication protocol. This communication protocol also adds a Secure Message Value (SMV) to each message that is going to be transmitted; thus, when the transmissions are complete, we check their integrity. Then, we describe our scheme for securely mining association rules over horizontally partitioned datasets between multiple parties, which is also based on homomorphic encryption and can be used in conjunction with our communication protocol.

4.1 Preliminaries

To formally describe the problem, we first define association rule mining on horizontally partitioned datasets, and then introduce our notations and roles.

4.1.1 Association rule mining on distributed databases

Distributed association rule mining is defined as follows. Given a transactional database with a number of transactions, DB is distributed to n sites $Site_1, Site_2, \ldots, Site_n$ where

Table 1 Categorization of reviewed PPDM schemes in this paper

Paper	Data mining algorithm	Security model	Integrity mechanism
[8]	–	Anonymization	–
[9]	–	Anonymization	–
[10]	–	Randomization	–
[11]	Random decision tree	Randomization	–
[13]	–	Data perturbation	–
[15]	–	Data perturbation	–
[16]	–	Data perturbation	–
[14]	–	Data perturbation	–
[17]	ID3	Data perturbation	–
[18]	Classification	Encryption	–
[19]	Classification	Encryption	–
[5]	Association rule mining	Encryption	–
[22]	K-means	Encryption & perturbation	–
[4]	Association rule mining	Homomorphic encryption	–
[24]	ID3	Homomorphic encryption	–
[25]	ID3	Homomorphic encryption	–
[26]	Association rule mining	Symmetric encryption	Timestamp
[3]	Association rule mining	Homomorphic encryption	Aggregate signature
[20]	Association rule mining	Custom protocol	–
[27]	Association rule mining	Shamir's secret key	–
[28]	Association rule mining	Encryption	–
[21]	–	Encryption	–
[23]	KNN	Homomorphic encryption	–
Our work	Association rule mining	Homomorphic encryption	Integrity checking function

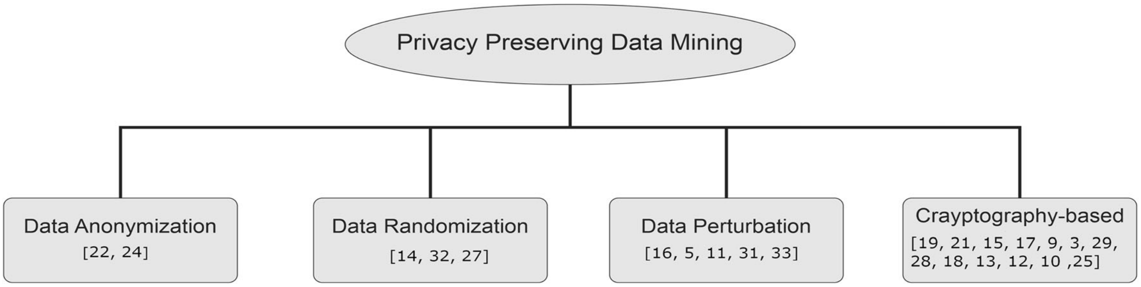


Fig. 3 Privacy Preserving Data Mining Schemes

$Site_i$ has database size $|DB|_i$ and $i = 1, 2, \ldots n$. Now $P.sup$ is global support count of all datasets and $P.sup_i$ is local support count of itemset P in party i, respectively. If user-defined minimum support threshold is $|s|$, itemset P is globally frequent with the following condition: $P.sup \geq |s| * |DB| = |s| * \Sigma_{i=1}^{n} |DB|_i$; similarly, itemset P is locally frequent at $site_i$, if $P.sup_i \geq |s| * |DB|_i$; and $P.sup$ i.e. the global support count of itemset P will be $P.sup = \Sigma_{i=1}^{n} P.sup_i$. Cheung et al. [29].

If we have the global support count of itemset P and Q, the global confidence of a rule $P \rightarrow Q$ is as described in Definition 4 :

Definition 4 Confidence($P \rightarrow Q$) $= \dfrac{|P \cup Q|.sup}{|P|.sup}$

Since we use homomorphic encryption in our *ICF* function to guarantee the integrity of the communication between cloud servers and parties, in the next subsection we review Paillier encryption which is a homomorphic encryption algorithm.

Table 2 Notations throughout the paper

Notation	Definition
LMFI	Local most frequent itemset in party's database
GMFI	Global most frequent itemset in all parties' databases
$\|DB\|_i$	Party$_i$ database size
$\|DB\|_G$	Global database size of all parties
P_{sup_i}	Local support count of itemset P
P_{sup}	Global support count of itemset P
sk_i^{RSA}	Party$_i$ RSA encryption private key
pk_i^{RSA}	Party$_i$ RSA encryption public key
sk_i	Party$_i$ Paillier encryption private key
pk_i	Party$_i$ Paillier encryption public key
SMV	Secure Message Value which is used in the proposed communication protocol
SIV	Secure Initial Value which is used in the proposed communication protocol
$GenSIVs()$	The operation of generating SIV for all parties
$GenRSAs()$	The operation of generating RSA key pairs for all parties and cloud server
$GenPaillierKeys()$	The operation of generating Paillier key pairs for all parties and cloud server
ICF	Integrity checking function which is described in the proposed communication protocol
$GenerateMessage(m)$	The operation of generating a message using our proposed protocol
$MineLocalRules()$	The operation of association rule mining of a party's dataset
$Dec_{sk_i}(c)$	Paillier decryption function of ciphertext c with the private key sk of party$_i$
$Enc_{pk_i}(m)$	Paillier encryption function of message m with the public key pk of party$_i$
$Dec_{sk_i^{RSA}}(c)$	RSA decryption function of ciphertext c with the private key sk^{RSA} of party$_i$
$Enc_{pk_i^{RSA}}(m)$	RSA encryption function of message m with the public key pk^{RSA} of party$_i$
CalculateGlobalFrequentItemSet()	Calculates global most frequent itemset
FindGlobalSupportCount()	Calculates global support count of an itemset
CalculateGlobalDatabaseSize()	Calculates global database size

4.1.2 Paillier encryption

Homomorphic encryption is a type of encryption that allows computation on the ciphertexts without decrypting them. We can use this property to perform a variety of data mining computations [25, 26]. Our scheme is not dependent on any specific homomorphic cryptosystem; it can be used with any homomorphic cryptosystem that has the property listed in the Definition 3. In this paper, we use the Paillier homomorphic cryptosystem proposed by Paillier in [30], which is based on the decisional composite residuosity problem for security. The Paillier cryptosystem operates in three stages:

- Phase 1: Key Generation

  1. Choose two large prime numbers p and q randomly and independently of each other such that $gcd(pq, (p-1), (q-1)) = 1$. This property is assured if both primes are of equal length.
  2. $N = p \cdot q$ and $\lambda = lcm(p-1, q-1)$, where lcm represents least common multiple.
  3. Select a random number g where $g \in \mathbb{Z}_{n^2}^*$.
  4. Ensure N divides the order of g by checking the existence of the following modular multiplicative inverse: $\mu = \left(L(g^{\lambda} \bmod n^2)\right)^{-1} \bmod n$ where function L is defined as $L(x) = \frac{x-1}{n}$
  5. (N, g) is the public key.
  6. (p, q) is the private key.

- Phase 2: Encryption

  1. Let m be a message to be encrypted where $0 \leq m < n$
  2. Pick a random number $0 < r < n$ and $r \in \mathbb{Z}_{\mathbb{N}}^*$ (i.e., ensure $gcd(r, n) = 1$) For a plaintext $Enc_{pk}(m) = C = g^m \cdot r^N \bmod N^2$ where pk is the generated public key.

- Phase 3: Decryption

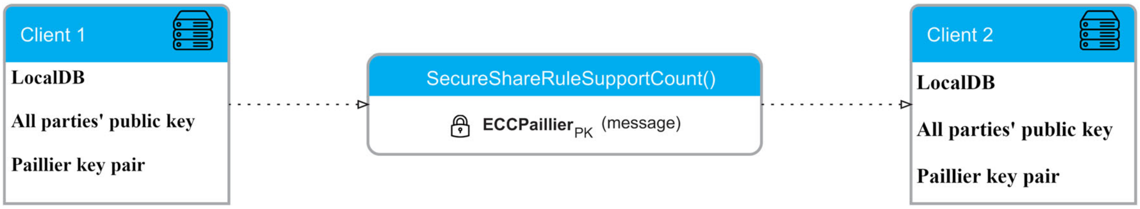


Fig. 4 Communication protocol proposed in [5]

1. In order to decrypt a ciphertext $C \in \mathbb{Z}_{\mathbb{N}}^{*}$, we compute the plaintext message as: $m = L(c^{\lambda} \mathrm{mod} n^{2}) \cdot \mu \mathrm{mod} n$

We also use the RSA public-key encryption algorithm to ensure the confidentiality of messages while they are being transmitted between parties.

4.1.3 Notations

Table 2 shows the notations we used to explain how our protocol works.

4.1.4 Roles

We have three different roles in our proposed scheme, which are as follows:

- A master server S that is trusted by all parties and is responsible for generating secrets and encryption keys.
- Parties that outsource encrypted data and receive the computed results.
- Cloud servers that receive the encrypted information from the clients, perform the computation and then send the constructed decision tree to the users.

4.2 Details of our proposed scheme

Our proposed privacy-preserving rule mining scheme is divided into two parts: the first is concerned with the communication protocol between parties, and the second is concerned with secure rule mining.

We use an innovative communication scheme based on homomorphic encryption, and we use this communication scheme throughout the entire process of our proposed method. In this section, we demonstrate that our scheme outperforms its competitors when the integrity of all participants' messages is one of the main security principles. We divide the distributed association rule problem into four phases, each with a specific outcome required for association rule mining.

4.2.1 Communication protocol

The term "message integrity" refers to the fact that a message has not been tampered with or altered. Integrity is one of the main principles of information security introduced by [31], there are many types of attacks that will threaten the integrity of the information that exists in the system, and our semi-trustworthy system should be hardened against them. Man In The Middle (MITM) attacks [32], can occur from any adversary, and the attacker can resend, manipulate, and even block communication between our semi-honest parties. We propose a custom communication protocol that checks the integrity of transmitted messages, and we demonstrate in this section that our communication protocol is safe against replay attacks that threaten message integrity.

Here, we describe our communication protocol which authenticates transferred messages based on homomorphic encryption. Precisely, we use RSA public key encryption to add an encryption layer to ensure the security of messages from each party to the server even when they have been eavesdropped. We also check the integrity of all of received messages, by a special function named ICF which is designed in this paper.

One of our scheme's advantages in comparison to its competitors is our designed communication protocol. The communication protocol aims to share a group of numeric values and check the integrity alongside preserving privacy during sharing values between all parties. Our communication protocol consists of four sequential and procedural steps including (1) Setup, (2) Generate Message (GenMessage()), (3) Integrity checking through ICF and (4) Verify.

Setup In this step, we initialize each party with its necessary secrets and RSA public keys of all parties. We also share a Secure Initial Value called $SIV \in Z_p$ between each party and trusted server S.

Generate message A sender generates Secure Message Value (SMV) for each message and keeps it alongside the original message. SMV is calculated as

$$\{SMV | x \in Z_p, x \leq SIV_i, SMV = Enc_{pk_i}(x), \Sigma_{k=1}^{n} SMV_k = SIV_i\}$$

where SIV_i is the sender's SIV generated in the setup phase. Then the sender generates the message's value which is encrypted by a homomorphic encryption algorithm and after all the message structures is prepared, the sender encrypts all the generated message using the receiver RSA public key encryption and sends it to the receiver i.e $Enc^{RSA}_{pk_i}(Enc_{pk_i}(value), SMV_i, data)$ where $value$ can be support count, database size or rule frequency and $data$ can be the details about $value$ like a rule.

Integrity checking After all the messages from a sender is received by the receiver, $Enc^{RSA}_{pk_i}(Enc_{pk_i}(value), SMV_i, data)$, firstly, it decrypts all received messages using its RSA private key, i.e $Dec^{RSA}_{sk_i}(Enc^{RSA}_{pk_i}(Enc_{pk_i}(value), SMV_i, data)) = Enc_{pk_i}(value)$, $SMV_i, data$. Then, it calculates ICF of all the received SMV_i as $ICF = \Sigma^n_{k=1} SMV_k = SMV_1 + SMV_2 + \ldots + SMV_n \quad = Enc_{pk_i}(x_1) * Enc_{pk_i}(x_2) * \ldots * Enc_{pk_i}(x_n)$. Because of homomorphic properties we can state $ICF = Enc_{pk_i}(x_1 + x_2 + \ldots + x_n)$ where n is the messages count, x_k is the k^{th} random generated in the Generate Message's phase and SMV_k is k^{th}'s message's Secure Message Value.

Verify: Finally, the receiver can validate the ICF by comparing it to the $Enc_{pk_i}(SIV_i)$ received from the setup phase as the sender's SIV and determining whether or not these two values are equal.

The sequences of our communication protocol are depicted in Algorithm 1.

In our communication protocol, the ICF begins checking the sum of all secrets generated by the sender in the Generate Message phase of our protocol after receiving all messages from the sender. As a result, the ICF can detect any message manipulation and will reject it.

Remark 1 Our privacy-preserving rule mining scheme is inspired by the [5] scheme, which employs an Elliptic-curve-based Paillier encryption method. In [5], all parties mine their local rules first, then send the homomorphic encrypted value of each rule's support count to a combiner. A replay attack, on the other hand, may result in a calculation error. The adversary only needs to do the following to launch a replay attack against the scheme that is proposed in [5]:

1. Eavesdropping one run of the transferred messages including $ECCPaillier_{PrivateKeyUser}$ (*ECCEncrypted-Data*) as M.
2. The adversary party sends two instances of M to its target party.
3. The target party receives the replied message M' and the original message M where $M' = M$ and then, the signature of message will be verified because it is signed with a valid user private key.
4. The calculation of summation for support count or database size would be incorrect because the replied message will be calculated more than once. For example, if an adversary party performs a replay attack on the second phase of the protocol of [5] which calculates global support, the replied message M' would be like $P_{sup_{m'}}$ which is equal to P_{sup_m}. Therefore, the calculation will be

Algorithm 1 Proposed communication protocol

1. Setup (For all parties): Server S generates the Paillier encryption key pair as (pk_i, sk_i) and RSA encryption key pair as (pk^{RSA}_i, sk^{RSA}_i) and broadcasts public parameters to all parties and sends the secret keys to its owner. Then, the server S generates a random number $SIV_i \in Z_p$ for each party p_i. After that, server S broadcasts all public keys to all parties and sends each secret key and SIV_i to its related party. Moreover, the server S broadcasts $Enc_{pk_i}(SIV_i)$ of the i^{th} user to all parties.
2. Generate Message: This procedure is used when a party starts to send a message to a receiver. Each message from a sender p_i to a receiver p_j has its random Secure Message Value $\mathbf{SMV} = \{SMV | x \in Z_p, x \leq SIV_i, SMV = Enc_{pk_i}(x), \Sigma^n_{k=1} SMV_k = SIV_i\}$ where SIV_i is the sender p_i Secure Initial Value and n is the count of all sending messages. Then, the sender encrypts the message value with the Paillier encryption. At the end, all the messages containing homomorphic encrypted of its value, its SMV and its data get encrypted using the receiver p_j RSA public key pk^{RSA}_j i.e. $Enc^{RSA}_{pk_j}(Enc_{pk_i}(value), SMV_i, data)$.
3. ICF: Receiver p_j first decrypts all received messages by its sk^{RSA}_j and retrieved $Enc_{pk_i}(value), SMV_i$ and $data$. Then it starts to calculate the ICF as $ICF = \Sigma^n_{k=1} SMV_k = SMV_1 + SMV_2 + \ldots + SMV_n = Enc_{pk_i}(x_1) * Enc_{pk_i}(x_2) * \ldots * Enc_{pk_i}(x_n)$ which based on homomorphic properties equals with $Enc_{pk_i}(x_1 + x_2 + \ldots + x_n)$ where n is the messages count, pk_i is the sender p_i's Paillier public key, x_k is the k^{th} random generated in the Generate Message's step and SMV_k is k^{th} message's Secure Message Value.
4. Verify: The receiver computes the ICF and verifies it by checking whether ICF equals the received $Enc_{pk_i}(SIV_i)$ of the user or not.

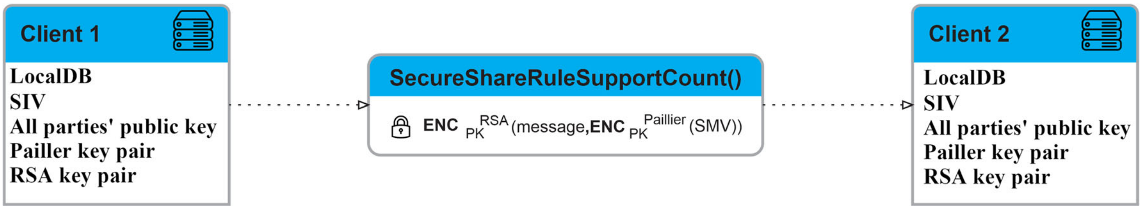


Fig. 5 Proposed communication protocol in our scheme

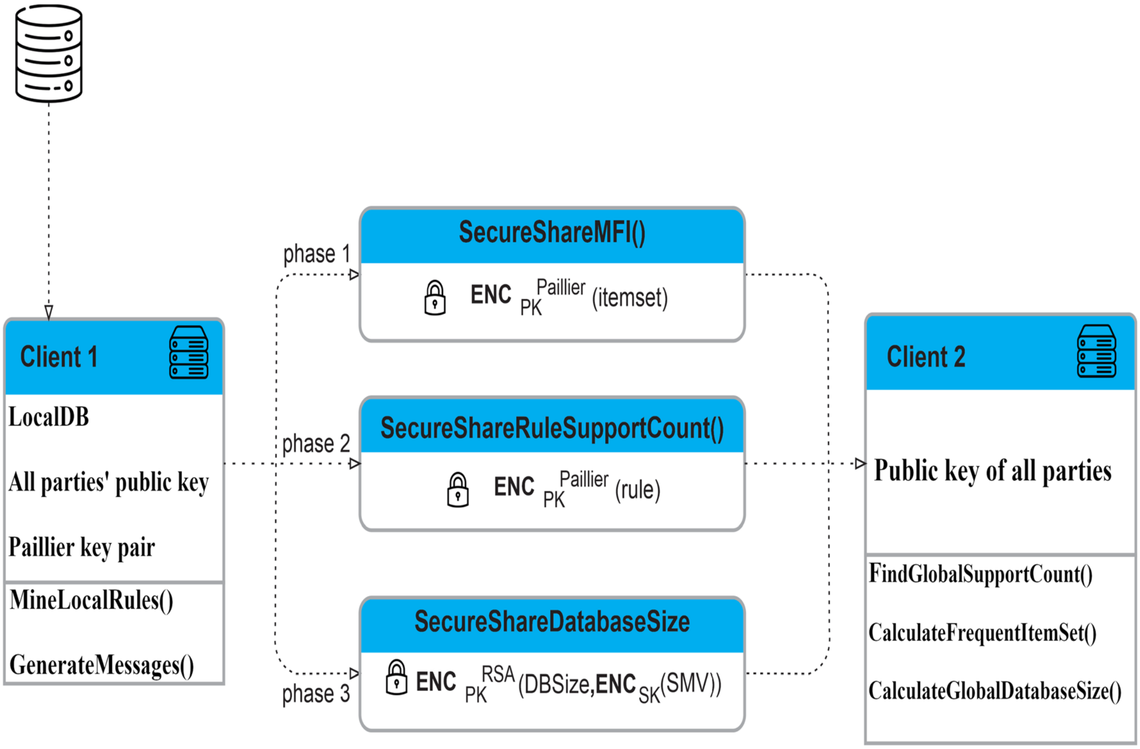

Client 1
LocalDB
All parties' public key
Paillier key pair
MineLocalRules()
GenerateMessages()
phase 1
phase 2
phase 3
SecureShareMFI()
ENC $_{PK}^{Paillier}$ (itemset)
SecureShareRuleSupportCount()
ENC $_{PK}^{Paillier}$ (rule)
SecureShareDatabaseSize
ENC $_{PK}^{RSA}$(DBSize,ENC$_{SK}$(SMV))
Client 2
Public key of all parties
FindGlobalSupportCount()
CalculateFrequentItemSet()
CalculateGlobalDatabaseSize()

Fig. 6 An abstract view of the proposed association rule mining scheme by [5]

$$Dec_{sk_i}(Enc_{pk_i}(P_{sup_1}) * Enc_{pk_i}(P_{sup_2}) * \dots * Enc_{pk_i}(P_{sup_m})* Enc_{pk_i}(P_{sup_{m'}}) * \dots * Enc_{pk_i}(P_{sup_n})) = Dec_{sk_i}(Enc_{pk_i}(P_{sup_1} + P_{sup_2} + \dots + P_{sup_m} + P_{sup_{m'}} + \dots + P_{sup_n})) = P_{sup_1} + P_{sup_2} + \dots + P_{sup_m} + P_{sup_{m'}} + \dots + P_{sup_n}.$$

It easily can be seen a miscalculation error occurs.

Our proposed communication protocol is capable of resolving the above-mentioned issue. Figure 5 depicts our proposed protocol, while Fig. 4 depicts the communication protocol in [5]. Figure 6 depicts an abstract representation of the method proposed in [5]. Our communication protocol is immune to replay attack because when a message m_i get received k times, $k > 1$, in the ICF step of our scheme, it will be calculated k times, hence the value of $ICF = \Sigma_{i=1}^{n+k} SMV_k$ will not be verified in Verify step where n is the original messages count.

4.2.2 Our method for privacy preserving association rule mining

Using the Paillier cryptosystem, we propose a method for mining association rules over distributed data. As previously stated, the proposed communication protocol is used to validate data communication between all parties.

Our proposed scheme for privacy-preserving association rule mining operates in three sequential phases, assuming that all parties have the same horizontally distributed database DB attributes and may have different transactions. We also believe that all parties involved in this protocol are semi-honest, which means that the parties must adhere to the exact prespecified protocol.

Phase 1 (*SecureShareMFI*()): In this phase, cloud server calculates maximal frequent itemset from all parties sites as:

- Each party calculates Local Maximal Frequent Itemset (LMFI) and signs the calculated itemset and sends it to cloud server through our proposed communication protocol.
- Cloud server calculates Global Maximal Frequent Itemset (GMFI) from all parties as Definition 5 and shares it with all parties.

Definition 5 $\cup_{i=0}^{n} MFI_{d_i}$ determines all global frequent where MFI_{d_i} is the most frequent itemset of party i.

Algorithm 2 defines the sequences of this phase.

Algorithm 2 The Calculation of Global Maximal Frequent Itemset (GMFI)

1. Each party calculates its LMFI.
2. Each party generates a SMV which is defined in the second step of Algorithm 1 for each item of its LMFI.
3. Cloud server decrypts all received messages from a party i using its RSA private key sk^{RSA} and verifies the integrity of all received items from the party by calculating ICF as $ICF = \Sigma_{k=1}^{n} SMV_k = SMV_1 + SMV_2 + \ldots + SMV_n = Enc_{pki}(x_1) * Enc_{pki}(x_2) * \ldots * Enc_{pki}(x_n) = Enc_{pki}(x_1 + x_2 + \ldots + x_n)$ where n is the messages count and SMV_i is the SMV of message i , then compares the calculated ICF with the secure initial value SIV_i of party i.
4. After verification, the cloud server calculates each user LMFI as $\cup_{i=0}^{k} item = LMFI_{d_i}$ where $item$ is each received item of party i which is received and verified by previous steps.
5. The cloud server generates GMFI from all its calculated LMFIs as $\cup_{i=0}^{n} LMFI_{d_i}$.

Phase 2 (*SecureShareRuleSupportCount*()): In this phase, the global support of an itemset from all parties is computed as the following steps:

- Each party generates all the non-empty subsets from global maximal frequent itemsets(GMFI); then each party finds the local support count of a candidate itemset P, then sends it to the cloud server with the proposed communication protocol. The local support count of candidate itemset P at $Party_i$ is denoted by $P.sup_i$.
- Cloud server verifies received messages from $party_i$ with the proposed ICF function. Then it calculates the Global support count of each itemset as defined in Definition 6.
- Cloud server securely shares the calculated global support count with all parties.

Definition 6 The global support count of an itemset P which is on $(n-1)$ parties can be calculated using homomorphic encryption properties as below:

$$Dec_{sk_i}(Enc_{pk_i}(P_{sup_1}) * Enc_{pk_i}(P_{sup_2}) * \ldots * Enc_{pk_i}(P_{sup_{n-1}})) = Dec_{sk_i}(Enc_{pk_i}(P_{sup_1} + P_{sup_2} + \ldots + P_{sup_{n-1}})) = P_{sup_1} + P_{sup_2} + \ldots + P_{sup_{n-1}}$$

.

The sequences of this phase is also defined in Algorithm 3.

Algorithm 3 Calculating Global Support Count

1. All parties calculate their local support counts from all subsets of GMFI.
2. Each party encrypts support of candidate itemset P using the cloud server Paillier homomorphic public key.
3. Each party sends the generated message containing encrypted support count in the previous step, using the proposed communication protocol.
4. Cloud server decrypts received messages using its RSA private key, then calculates the ICF as $ICF = \Sigma_{k=1}^{n} SMV_k = SMV_1 + SMV_2 + \ldots + SMV_n = Enc_{pki}(x_1) * Enc_{pki}(x_2) * \ldots * Enc_{pki}(x_n) = Enc_{pki}(x_1 + x_2 + \ldots + x_n)$ where n is the messages count and SMV_i is the SMV of message i , then verifies the received messages by checking it with the received SIV for the related party.
5. For all received itemsets: Cloud server calculates Global Support Count of an itemset P using homomorphic encryption properties $Dec_{sk}(Enc_{pk}(P_{sup_1}) * Enc_{pk}(P_{sup_2}) * \ldots * Enc_{pk}(P_{sup_n})) = Dec_{sk}(Enc_{pk}(P_{sup_1} + P_{sup_2} + \ldots + P_{sup_n})) = P_{sup_1} + P_{sup_2} + \ldots + P_{sup_n}$ where pk and sk are respectively cloud server's Paillier encryption public key and private key.
6. For all parties: Cloud server encrypts the calculated global support count by the party's RSA public key pk_i^{RSA} and sends it to the party.
7. Each party decrypts the received message from the cloud server by its RSA private sk_i^{RSA}.

Phase 3 (*SecureShareDatabaseSize*()): In this phase, global database size is calculated. The global database size is calculated by aggregating the encrypted local database sizes from all sites, as shown below:

- Each party generates n random numbers $r_k \in \mathbb{Z}\{\Sigma_{i=1}^{n} r_{ik} = |DB|_i\}$ where $|DB|_i$ is the $party_i$ local database size and n is random in $\mathbb{Z} < |DB|_i$. Then each party sends the encrypted value of r_k to the cloud server through the communication protocol.
- Cloud server computes the summation of the received numbers which is equal to the party's database size.
- Cloud server calculates the global database size as

$$\begin{aligned} &Dec_{sk}(Enc_{pk}(S_1) * Enc_{pk}(S_2) * \ldots * Enc_{pk}(S_n)) \\ &= Dec_{sk}(Enc_{pk}(S_1 + S_2 + \ldots + S_n)) = S_1 + S_2 + \ldots \\ &+ S_n \end{aligned}$$

.

The details of this phase are defined in Algorithm 4.

Algorithm 4 Global Database Size Calculation

```
1. Each party generates $n$ random numbers $r_k \in \mathbb{Z}\{\Sigma_{i=1}^{n} r_{ik} = |DB|_i\}$ where $|DB|_i$ is the $party_i$ local
   database size and $n$ is random in $\mathbb{Z} < |DB|_i$.
2. For each calculated $r_i$ from previous step: The party encrypts $r_i$ using Paillier homomorphic encryption
   with the cloud server public key and sends $Enc_{pk}(r_i)$ to the cloud server through the proposed
   communication protocol.
3. For all parties: Cloud server verifies the received message from the party by calculating ICF value
   of received message as $ICF = \Sigma_{k=1}^{n} SMV_k = SMV_1 + SMV_2 + \ldots + SMV_n = Enc_{pk_i}(x_1) * Enc_{pk_i}(x_2) * \ldots * Enc_{pk_i}(x_n) = Enc_{pk_i}(x_1 + x_2 + \ldots + x_n)$
   where $n$ is the messages count and $SMV_i$ is the SMV of
   message $i$ and $x_i$ is the random number generated in ICF generate message phase.
4. Cloud server calculates the local database size $Enc_{pk}(|DB|_i)$ for party $i$ using homomorphic encryption
   properties $(Enc_{pk}(r_1) * Enc_{pk}(r_2) * \ldots * Enc_{pk}(r_n)) = (Enc_{pk}(r_1 + r_2 + \ldots + r_n))$ and after
   decryption we obtain, $r_1 + r_2 + \ldots + r_n = |DB|_i$ where $pk$ is the cloud server's Paillier public key.
5. Cloud server calculates global database size $|DB|_G$ using homomorphic encryption properties
   $Dec_{sk}(Enc_{pk}(|DB|_1) * Enc_{pk}(|DB|_2) * \ldots * Enc_{pk}(|DB|_n)) = Dec_{sk}(Enc_{pk}(|DB|_1 + |DB|_2 + \ldots + |DB|_n)) = |DB|_1 + |DB|_2 + \ldots + |DB|_n = |DB|_G$.
6. Cloud server encrypts the calculated global database size i.e. $|DB|_G$ with each party's RSA public
   key $pk_i^{RSA}$ and sends $Enc_{pk_i^{RSA}}(|DB|_G)$ to the party.
7. Each party decrypts the received message $Enc_{pk_i^{RSA}}(|DB|_G)$ from cloud server by its RSA private
   $sk_i^{RSA}$ and obtains $|DB|_G$.
```

After deriving the global support count of all itemsets and the global size of the database, the cloud server derives the global association rules using Apriori algorithm [7]. Then it encrypts each rule by the related party's RSA public key. Figure 7 illustrates the communication scheme and association rule mining scheme between a party and a cloud server.

5 Evaluation and security analysis

In this section, the proposed scheme is analyzed with various evaluation metrics such as time complexity, security, communication cost and accuracy. We implemented our scheme on three real-world datasets from the UCI Machine Learning Repository [33], namely the Heart Disease dataset, the Adult dataset, and the Cardiotocography dataset that are widely used in the field of data mining. Table 3 displays the characteristics of the datasets that were tested.

We also implemented the proposed schemes in [3, 4, 20, 27, 28] and compared their security level and performance to our scheme. The results were obtained using a PC with 16 GB of RAM and a Core i7 CPU running Ubuntu.

All of the schemes chosen for this comparison proposed unique approaches to solve the problem of privacy-preserving association rule mining. All of the methods chosen for this comparison have innovations that improve the privacy of their participants, hence we choose them to compare with our scheme's security. We compared our method to its competitors based on their performance and security for their participants.

[3] proposes a scheme for distributed association rule mining over encrypted data. Domadiya and Rao [20] is a study for privacy-preserving association rule mining using a custom encryption scheme, with the encryption aiding in privacy enhancement. Shamir's secret sharing scheme is used to protect privacy in [27] and [4] employs the

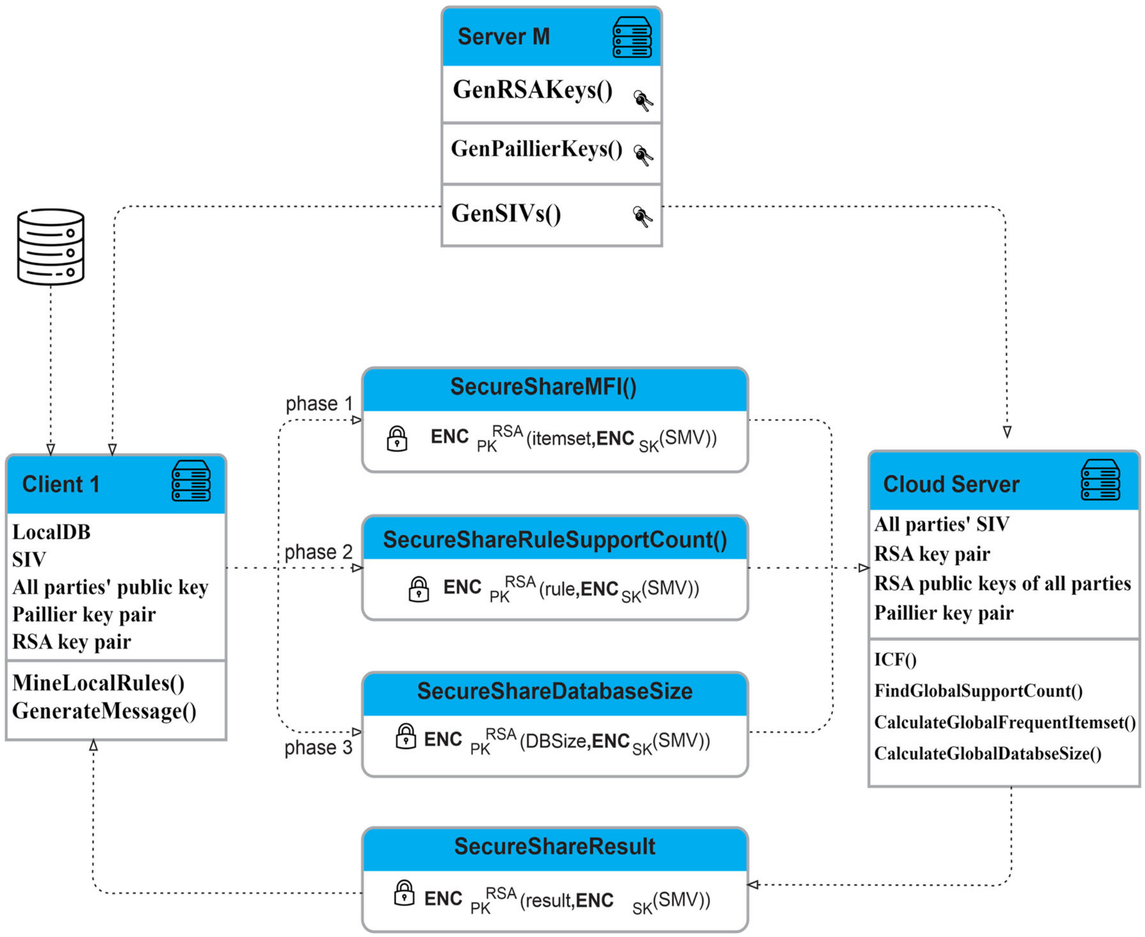


Fig. 7 The association rule mining scheme between one party and the cloud server

Table 3 Characteristics of tested datasets

Dataset	Number of instances	Number of features
Heart disease	303	75
Adults	48842	14
Cardiotocography	2126	23

homomorphic encryption and adds fictitious data before transmission. Rajesh and Selvakumar [28] proposes a privacy-preserving data-mining scheme that incorporates data-mining perturbation.

5.1 Time complexity

The complexity of our scheme depends on the number of candidate itemsets(N) and the number of collaborative parties(n). We used MFI technique in our proposed method. To calculate the global MFI with M subsets, the computational cost at algorithm 2 is $O(Mn^2)$. The computational costs for the Algorithm 3 and the Algorithm 4 are reduced to the calculation of summation M subsets using homomorphic encryption which is $O(Mn^2)$. Our communication scheme using Algorithm 1 has a cost of $O(Nn)$ where N is the number of subsets generated for a message.

All of the implemented schemes were tested for three parties using real-world datasets shared on [33] and their effectiveness was measured by the average amount of time spent by each party. We randomly partitioned the dataset into three parts, then assigned one of the generated parts to each party. Using these schemes, we calculated the average elapsed time for each user to perform association rule mining. As you can see in Table 4, the proposed method has performed better than other methods on all datasets and has spent less time.

5.2 Communication cost

In our proposed method, we have four phases in which various types of data consisting of encryption keys, mining parameters and encrypted mined knowledge are transmitted. We have counted all messages transmitted between our

Table 4 Average elapsed time for each user to perform association rule mining

Proposed scheme	Our method	[3]	[20]	[4]	[27]	[28]
Heart disease	326.26 ms	8824 ms	19946 ms	18675 ms	21235 ms	189934 ms
Adult	810.72 s	1483.2 s	2464.3 s	2238.14 s	2935.71 s	2423.67 s
Cardiotocography	573.49 s	746.69 s	1307.13 s	1075.85 s	1439.89 s	1198.62 s

parties and the cloud server to analyze our communication burden in comparison to other solutions. The results are shown in Table 5. Our method adds a communication burden to network and that is because of our secure communication method which shares clients' secrets with homomorphic encrypted messages.

5.3 Security

Our scheme has a higher security level because it employs our custom communication protocol, which allows users to check the integrity of received data and, as a result, it can withstand security attacks against integrity, and also other attacks such as replay attacks. Table 6 compares the proposed method to other similar schemes based on the level of security they provide. We also employed the Paillier encryption method, which is based on discrete logarithms. As shown in Table 6, none of the schemes in this table satisfy all of the security properties in the table, whereas our proposed method does.

All data in our proposed scheme is encrypted before being shared and will not be decrypted, ensuring the privacy of sensitive data. We also used our integrity checking function to ensure that no unauthorized data is accepted, and thus data integrity is maintained. Table 6 compares our security achievements for privacy-preserving association rule mining to those of other related schemes. Our proposed communication protocol can also be used to perform distributed data mining schemes such as rule mining and classification over encrypted data, ensuring the integrity of the entire transferred data. Furthermore, we conclude that the intensive computing tasks can be outsourced to a cloud server.

5.4 Accuracy

We have tested our scheme with different sizes of datasets to measure our mining parameters. The results of our implementations are plotted in Table 7. Our method does not have any effect on mining accuracy.We also checked our accuracy on different scales that showed the accuracy was the same as the original method on different dataset scales.

6 Conclusion

The main contribution of this paper is the presentation of a new scheme for privacy-preserving association rule mining over cloud servers that employs a new protocol for communication between all parties in order to maintain the privacy of individual values while the result states the same. To avoid disclosing sensitive data to third parties, we

Table 5 Average elapsed time for each user to perform association rule mining

Proposed scheme	Our method	[3]	[20]	[4]	[27]	[28]
Heart disease	10382	4738	4382	7382	6881	4078
Adult	92748	21064	20280	32429	31289	19157
Cardiotocography	63188	11639	10908	13623	31289	9685

Table 6 Security comparison of the proposed scheme with related schemes

Scheme	Integrity	Confidentiality	Availability	Replay attack resistance
[3]	× (in this paper)	✓	✓	× (in this paper)
[20]	×	✓	✓	× (in this paper)
[4]	×	✓	✓	× (in this paper)
[27]	×	✓	✓	× (in this paper)
[28]	×	✓	✓	× (in this paper)
Our scheme	✓	✓	✓	✓ (in this paper)

Table 7 Accuracy of our scheme on different datasets

Dataset	Accuracy	Increasing dataset size			
		75	150	225	300
Heart disease	Precision	82.43	88.39	84.21	82.31
	Coverage	80.27	85.83	81.47	78.33
Dataset	Accuracy	Increasing dataset size			
		20000	30000	40000	48000
Adults	Precision	91.43	93.23	90.14	94.85
	Coverage	86.46	88.35	84.40	90.04
Dataset	Accuracy	Increasing dataset size			
		500	1000	1500	2000
Cardiotocography	Precision	98.53	98.55	97.02	97.18
	Coverage	97.35	97.41	93.97	94.07

used homomorphic encryption, and all computations are performed over encrypted data.

Our method used the benefits of homomorphic encryption and we proposed a scheme to outsource the computational cost of distributed association rule mining to cloud servers. We also reduced the problem of distributed association rule mining over horizontally partitioned datasets into three phases. Although our method uses homomorphic encryption, we proposed a custom communication protocol to achieve a higher level of security.

We showed that data owners can trust our scheme to mine knowledge from their aggregated data and their privacy will be preserved without losing any global rule. Hence our scheme is usable for many data owners who have horizontally partitioned their datasets.

We implemented our method, and the results show that we can achieve a high level of privacy at a low computational cost. There are several avenues for future work. Parallel computation techniques should be used in future work to reduce server-side computational costs even further. This paper focuses on association rule mining, but other types of data mining, such as classification, clustering, and sequence detection, face similar privacy concerns also.

Funding This work was supported by Shahid Rajaee Teacher Training University.

Declarations

Conflict of interest The authors declare that they have no conflicts of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

1. Sakpal, M. (2019). A glimpse of secure multiparty computation for privacy preserving data mining. *Advanced Science, Engineering and Medicine, 11*(1–2), 163–166.
2. Deka, G.C. (2014). Handbook of research on securing cloud-based databases with biometric applications. IGI Global.
3. Hammami, H., Brahmi, H., Brahmi, I., Yahia, S.B.(2017). Using homomorphic encryption to compute privacy preserving data mining in a cloud computing environment. In *European, Mediterranean, and middle eastern conference on information systems* (pp. 397–413). Springer.
4. Li, L., Lu, R., Choo, K. K. R., Datta, A., & Shao, J. (2016). Privacy-preserving-outsourced association rule mining on vertically partitioned databases. *IEEE Transactions on Information Forensics and Security, 11*(8), 1847–1861.
5. Kantarcioglu, M., & Clifton, C. (2004). Privacy-preserving distributed mining of association rules on horizontally partitioned data. *IEEE Transactions on Knowledge and Data Engineering, 16*(9), 1026–1037.
6. Agrawal, R., Imieliński, T., Swami, A. (1993). Mining association rules between sets of items in large databases. In *Proceedings of the 1993 ACM SIGMOD international conference on management of data* (pp. 207–216).
7. Agrawal, R., Srikant, R., et al. (1994). Fast algorithms for mining association rules. In *Proc. 20th int. conf. very large data bases, VLDB* (Vol. 1215, pp. 487–499).
8. Matatov, N., Rokach, L., & Maimon, O. (2010). Privacy-preserving data mining: A feature set partitioning approach. *Information Sciences, 180*(14), 2696–2720.
9. Prakash, M., & Singaravel, G. (2015). An approach for prevention of privacy breach and information leakage in sensitive data mining. *Computers & Electrical Engineering, 45,* 134–140.
10. Kalyani, G., Rao, M. C. S., & Janakiramaiah, B. (2018). Privacy-preserving classification rule mining for balancing data utility and knowledge privacy using adapted binary firefly algorithm. *Arabian Journal for Science and Engineering, 43*(8), 3903–3925.
11. Wu, X., Qi, L., Gao, J., Ji, G., & Xu, X. (2022). An ensemble of random decision trees with local differential privacy in edge computing. *Neurocomputing, 485,* 181–195.
12. Shen, L., Wu, X., Wu, D., Xu, X., Qi, L.(2020). A survey on randomized mechanisms for statistical learning under local differential privacy. In *2020 IEEE 22nd international conference on high performance computing and communications; IEEE 18th international conference on smart city; IEEE 6th international conference on data science and systems (HPCC/SmartCity/DSS). IEEE* (pp. 1195–1202).
13. Kao, Y. H., Lee, W. B., Hsu, T. Y., Lin, C. Y., Tsai, H. F., & Chen, T. S. (2015). Data perturbation method based on contrast mapping for reversible privacy-preserving data mining. *Journal of Medical and Biological Engineering, 35*(6), 789–794.
14. Chen, T. S., Lee, W. B., Chen, J., Kao, Y. H., & Hou, P. W. (2013). Reversible privacy preserving data mining: A combination of difference expansion and privacy preserving. *The Journal of Supercomputing, 66*(2), 907–917.
15. Du, M., Wang, K., Xia, Z., & Zhang, Y. (2018). Differential privacy preserving of training model in wireless big data with edge computing. *IEEE Transactions on Big Data, 6*(2), 283–295.
16. Wu, X., Khosravi, M. R., Qi, L., Ji, G., Dou, W., & Xu, X. (2020). Locally private frequency estimation of physical symptoms for infectious disease analysis in internet of medical things. *Computer Communications, 162,* 139–151.
17. Zhan, J., Matwin, S., Chang, L. (2005). Privacy-preserving decision tree classification over vertically partitioned data.

Multiagent Data Warehousing (MADW) and Multiagent Data Mining (MADM).

18. Li, Y., Jiang, Z.L., Wang, X., Yiu, S.M., Fang, J.(2017). Outsourced privacy-preserving random decision tree algorithm under multiple parties for sensor-cloud integration. In *International conference on information security practice and experience* (pp. 525–538). Springer.
19. Li, Y., Jiang, Z. L., Yao, L., Wang, X., Yiu, S., & Huang, Z. (2019). Outsourced privacy-preserving C4.5 decision tree algorithm over horizontally and vertically partitioned dataset among multiple parties. *Cluster Computing, 22*(1), 1581–1593.
20. Domadiya, N., & Rao, U. P. (2018). Privacy-preserving association rule mining for horizontally partitioned healthcare data: a case study on the heart diseases. *Sādhanā, 43*(8), 127.
21. Bahrami, M., Singhal, M. (2015). A light-weight permutation based method for data privacy in mobile cloud computing. In *2015 3rd IEEE international conference on mobile cloud computing, services, and engineering. IEEE* (pp. 189–198).
22. Vaidya, J., Clifton, C. (2003). Privacy-preserving k-means clustering over vertically partitioned data. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 206–215).
23. Tan, Y., Wu, W., Liu, J., Wang, H., & Xian, M. (2020). Lightweight edge-based KNN privacy-preserving classification scheme in cloud computing circumstance. *Concurrency and Computation: Practice and Experience, 32*(19), e5804.
24. Li, Y., Jiang, Z.L., Wang, X., Yiu, S.M. (2017). Privacy-preserving ID3 data mining over encrypted data in outsourced environments with multiple keys. In *2017 IEEE international conference on computational science and engineering (CSE) and IEEE international conference on embedded and ubiquitous computing (EUC). IEEE* (Vol. 1, pp. 548–555).
25. Li, Y., Jiang, Z.L., Wang, X., Yiu, S.M., Zhang, P. (2017). Outsourcing privacy preserving ID3 decision tree algorithm over encrypted data-sets for two-parties. In *2017 IEEE Trustcom/BigDataSE/ICESS. IEEE* (pp. 1070–1075).
26. Huang, C., Lu, R., & Choo, K. K. R. (2017). Secure and flexible cloud-assisted association rule mining over horizontally partitioned databases. *Journal of Computer and System Sciences, 89,* 51–63.
27. Domadiyaa, N.H., Rao, U.P.(2018). Privacy preserving approach for association rule mining in horizontally partitioned data using MFI and shamir's secret sharing. In *2018 IEEE 13th international conference on industrial and information systems (ICIIS). IEEE* (pp. 217–222).
28. Rajesh, N., & Selvakumar, A. A. L. (2019). Association rules and deep learning for cryptographic algorithm in privacy preserving data mining. *Cluster Computing, 22*(1), 119–131.
29. Cheung, D. W., Ng, V. T., Fu, A. W., & Fu, Y. (1996). Efficient mining of association rules in distributed databases. *IEEE Transactions on Knowledge and Data Engineering, 8*(6), 911–922.
30. Paillier, P. (1999). Public-key cryptosystems based on composite degree residuosity classes. In *International conference on the theory and applications of cryptographic techniques* (pp. 223–238). Springer.
31. Whitman, M.E., Mattord, H.J. (2021). Principles of information security. Cengage learning.
32. Conti, M., Dragoni, N., & Lesyk, V. (2016). A survey of man in the middle attacks. *IEEE Communications Surveys & Tutorials, 18*(3), 2027–2051.
33. Hungarian Institute of Cardiology. Budapest: Andras Janosi M.D., U.H.Z.S.W.S.M.U.H.B.S.M.P.M.V.M.C.L.B., Cleveland Clinic Foundation: Robert Detrano M.D., P.: UCI Repository of Machine LearningDatabases, http://www.ics.uci.edu/∼mlearn/MLRepository.html

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Sahand Zehtabchi received his BS in Computer Engineering from Ayatollah Boroujerdi University in 2016 and received his MSc. student in Computer Engineering at Shahid Rajaee Teacher Training University in 2020. His research interests includes machine learning and data mining.

Negin Daneshpour is an associate professor in the Computer Engineering faculty of Shahid Rajaee Teacher Training University, Tehran, Iran. She received her BS degree in computer engineering hardware from the department of electronics and computer engineering at Shahid Beheshti University, Iran, where she graduated summa cum laude in 1999. She received an MS degree and Ph.D. in computer engineering-software from the Department of Computer Engineering and Information Technology at the Amirkabir University of Technology, Iran, in 2002 and 2010, respectively. Her research interests focus on data analysis and management, data mining, and data preprocessing.

Masoumeh Safkhani received the Ph.D. degree in electrical engineering from the Iran University of Science and Technology, in 2012, with a focus on security analysis of RFID protocols. She is currently an Associate Professor with the Computer Engineering Department, Shahid Rajaee Teacher Training University, Tehran, Iran. She is the author/coauthor of over 70 technical articles in information security and cryptology in major international journals and conferences. Her current research interests include security analysis of lightweight and ultra-lightweight protocols, targeting constrained environments, such as RFID, the IoT, VANET, and WSN.