Sādhanā Vol. 42, No. 12, December 2017, pp. 1997–2007
https://doi.org/10.1007/s12046-017-0743-4

Privacy-preserving distributed mining of association rules using Elliptic-curve cryptosystem and Shamir's secret sharing scheme

HARENDRA CHAHAR, B N KESHAVAMURTHY and CHIRAG MODI*

Department of Computer Science and Engineering, National Institute of Technology Goa, Farmagudi,
Goa 403 401, India
e-mail: hchahar616@gmail.com; bnkeshav.fcse@nitgoa.ac.in; cnmodi@nitgoa.ac.in

MS received 13 April 2016; revised 25 January 2017; accepted 19 May 2017; published online 17 November 2017

Abstract. Distributed data mining has played a vital role in numerous application domains. However, it is widely observed that data mining may pose a privacy threat to individual's sensitive information. To address privacy problem in distributed association rule mining (a data mining technique), we propose two protocols, which are securely generating global association rules in horizontally distributed databases. The first protocol uses the notion of Elliptic-curve-based Paillier cryptosystem, which helps in achieving the integrity and authenticity of the messages exchanged among involving sites over the insecure communication channel. It offers privacy of individual site's information against the involving sites and an external adversary. However, the collusion of two sites may affect the privacy of individuals. To address this problem, we incorporate Shamir's secret sharing scheme in the second protocol. It provides privacy by preventing colluding sites and external adversary attack. We analyse both protocols in terms of fulfilling the privacy-preserving distributed association rule mining requirements.

Keywords. Privacy; distributed association rule mining; elliptic-curve-based Paillier cryptosystem; Shamir's secret sharing scheme.

1. Introduction

In modern era, various business organizations are growing rapidly and expanding their business by distributing them across large number of sites, which are geographically located at different regions and storing their per day data at their own locations. Knowledge discovery and hidden pattern discovery from these organizations using a centralized data mining approach is not always better since these approaches require large number of data transfers, which increases the network communication overhead when combining all data at a central location (data warehouse). For this reason, distributed data mining techniques have been widely used. In distributed data mining, these techniques generate local knowledge at each site through local processing on their data. This local knowledge is exchanged to each involving site in order to derive global mining results. In a distributed environment, there are three types of data distributions considered (figure 1).

- Horizontal distribution. In horizontal distribution of data, the set of all attributes will be the same for all sites but number of transactions will be different at each site.
- Vertical distribution. In vertical distribution of data, the set of attributes will be different for all sites but number of transactions will be the same at each site.
- Hybrid distribution. In hybrid distribution of data, data are distributed either first horizontally and then vertically or vice-versa.

For market analysis, business organizations would like to combine local knowledge discovered from all sites in order to derive global knowledge. However, the sites may not admit to release their sensitive knowledge to other sites due to privacy or confidentiality of local knowledge generated at their own locations. The traditional distributed data mining techniques have been investigated in terms of privacy and security and it is observed that these techniques affect the privacy of individual site's information. Thus, there is a need for researching privacy-preserving distributed mining.

For example, suppose a supermarket company has *n* branches at different locations. For market analysis, the owner of company wants to find the relationship among items bought together globally. Through this analysis, the

*For correspondence

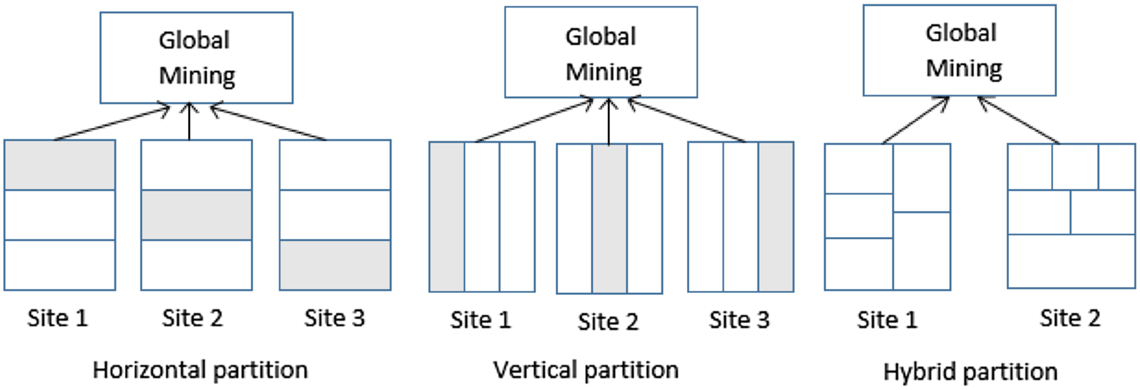

Figure 1. Different types of data distribution.

owner can efficiently provide some offers for increasing the sale, which in turn increases revenue of the company. However, distributed sites are not willing to share their local support count knowledge due to privacy or confidentiality issues.

To address the problem of privacy in distributed data mining, approaches based on data perturbation were proposed in [1–3], wherein original data are modified by adding noise and then they are exchanged to other sites for data mining operation. However, they lack in providing the guarantee about privacy. As an alternative solution, cryptography-based secure multi-party computation approaches as in [4, 5] are considered to fit very well for addressing the privacy problem in distributed data mining. They provide well-defined framework for offering privacy and include methodologies for the privacy guarantee. However, they are slower compared with perturbation-based approaches. They have considerable communication and computation overhead. Therefore, there is a need for designing an efficient protocol that can yield global association rules without compromising the privacy and security of any involving site's information, while preventing external adversaries and colluding sites with reduced communication and computation cost.

In this paper, we propose two protocols for deriving global association rules in horizontally distributed databases securely. The first protocol uses the notion of Elliptic-curve-based Paillier public key cryptosystem [6] to generate secret keys, which helps in achieving the integrity and authenticity of the messages sent among sites over the insecure communication channel. The second protocol uses the notion of Shamir's secret sharing scheme [7] to provide the privacy by preventing sites collusion and external adversary attack.

Rest of the paper is organized as follows. Section 2 discusses the existing proposals for privacy-preserving distributed association rules mining. Section 3 presents the proposed privacy-preserving protocols. Section 4 analyses the proposed protocols. Section 5 concludes our research work with references at the end.

2. Background and related work

2.1 *Association rule mining*

Association rule mining technique discovers the interesting relationship among attributes in large databases. It is widely used for many application areas like intrusion detection system, web usage mining, continuous production and bio-informatics system of real life environment.

Association rule mining is defined in [8]. Assume $A = \{a_1, a_2, a_3,\ldots,a_n\}$ is a set of size n binary-value attributes. $DB = \{t_1, t_2, t_3,\ldots,t_m\}$ is a set of size m transactions. In this, each transaction t is called an itemset if $t \subseteq A$. 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 $P \cap Q = \emptyset$. The support value of an association rule $P \subseteq Q$ can be derived as follows. This rule has support value S if the probability of a transactions in databases DB containing both P and Q is S. The confidence of this rule is C if the probability of a transaction in database DB containing P and then Q is C.

$$Support(P \Rightarrow Q) = \frac{|P \cup Q|}{|DB|}$$

$$Confidence(P \Rightarrow Q) = \frac{|P \cup Q|}{|P|}$$

If support value of an itemset is greater than or equal to user-defined minimum support threshold s then it is called as a frequent itemset. Association rule mining works in two steps, as per apriori algorithm [9].

- In the first step, discover all itemsets from a given database
  that satisfy a user-defined minimum support threshold s. These itemsets are called frequent itemsets.
- In the second step, generate all possible association rules from the frequent itemsets discovered in the first step. The overall cost of mining association rules is dominant in the first step because in this step we need

to scan the database for counting the support value of itemsets.

2.2 *Distributed association rule mining in horizontally distributed databases and privacy problem*

The distributed association rule mining [10] is defined as follows. Assume that a transactional database of size DB is horizontally distributed to n sites $Site_1$, $Site_2$,..., $Site_n$ where $Site_i$ has database size DB_i and $i = 1, 2, \ldots, n$. Now $P.sup$ and $P.sup_i$ are the global and local support count of itemset P, respectively. If user-defined minimum support threshold is s then itemset P is globally frequent with the following condition: $P.sup \geq s * |DB| = s * \sum_{i=1}^{n} |DB_i|$; similarly, itemset P is locally frequent at $site_i$, if $P.sup_i \geq s * |DB_i|$; and the global support count of itemset P will be $P.sup = \sum_{i=1}^{n} P.sup_i$.

If we have the global support count of itemsets P and Q then the global confidence of a rule $P \Rightarrow Q$ is as follows:

$$Confidence(P \Rightarrow Q) = \frac{|P \cup Q|.sup}{|P|.sup}.$$

The privacy problem is to derive global association rules in distributed and unsecured environment in such a way that any involving site should not be able to reveal other site's information even though site collusion takes place. An external adversary should not be able to affect the privacy of any site's information. He/she should not be able to hamper global mining result. In addition, it should have low communication and computational cost.

2.3 *Privacy-preserving distributed association rules mining: existing proposals*

There have been several works to date for privacy-preserving distributed association rule mining. Existing approaches can be classified into data perturbation approaches, which are further divided into addition and multiplication; secure multi-party computation, which is further divided into secure union, secure comparison and secure sum; and cryptography approaches, which are divided into Shamir's secret sharing, oblivious transfer and homomorphic encryption (see figure 2).

Data perturbation techniques [1–3] provide the privacy through modifying the original data values by adding and multiplying noise; later, it is exchanged with other sites. Hence, receiving sites are unable to identify the original data values. The basic idea of secure multi-party computation is that a computation is secure. At the end of computation, no site knows anything except its local value and global result. In secure sum method of secure multi-party computation, the initiator site chooses a random number uniformly and adds this to its local value and sends the sum value to next site; thus, the next site is unable to learn actual local value of initiator site. In the secure union method, commutative encryption techniques, e.g., Pohlig-Hellman encryption, [11] are used, wherein each site encrypts its items and each site also encrypts the other sites items. In addition, decryption can be done in any order. Thus, by permuting the encrypted items, it can prevent tracking the source of an item. In Shamir's secret sharing scheme [7], the local values of items from sites are not exchanged directly with other sites. Only shares are exchanged with other sites; hence, sites are unable to reveal the local value of other site even when they collude since they have only share of local value. In oblivious transfer [12, 13], sender selects one piece from large number of information and sends it to the receiver, but sender does not know the piece that has been selected. In homomorphic encryption [14], addition and multiplication can be performed on encrypted data. These operations provide another encrypted result. However, when decrypted, it matches the result performed on plain text. In this, each site encrypts local information using homomorphic encryption; however, receiving site gets only total of all local values.

Chen *et al* [15] have presented an overview about data mining techniques and different types. The secure two-party computations concept has been introduced by Yao [5]; later, it is generalized to multi-party computation. A secure computation protocol in ID3 classification algorithm for two parties is proposed in [16], where data are horizontally partitioned and complete zero knowledge leakage is achieved. Lindell and Pinkas [17] have discussed various paradigms of secure multi-party computation, which are used in privacy-preserving data mining, while Clifton *et al* [18] have presented four efficient methods, namely, secure set union, secure sum, scalar product and secure size of set intersection. In secure set union, commutative encryption is used. This approach prevents identifying the source of item but increases the computation overhead. In the secure sum method, sites can collude, which reveals the information of other site. These methods can be used for providing privacy preservation for association rule mining in horizontally and vertically partitioned databases. An efficient two-party algorithm for privacy-preserving association rule mining in vertically partitioned databases is designed by Vaidya and Clifton [19]. It does not achieve complete zero knowledge leakage. It is limited to boolean attributes and works only for two-party computation.

Kantarcioglu and Clifton [20] have presented a scheme for privacy-preserving association rules mining in horizontally distributed databases. It has two phases. In the first phase, it uses commutative encryption for hiding the source of itemset during secure union of locally large itemsets. It also uses fake itemsets to hide the actual number supported, which increases the communication overhead. In the second phase, secure sum method is used for calculating global support count. It is not a collusion-

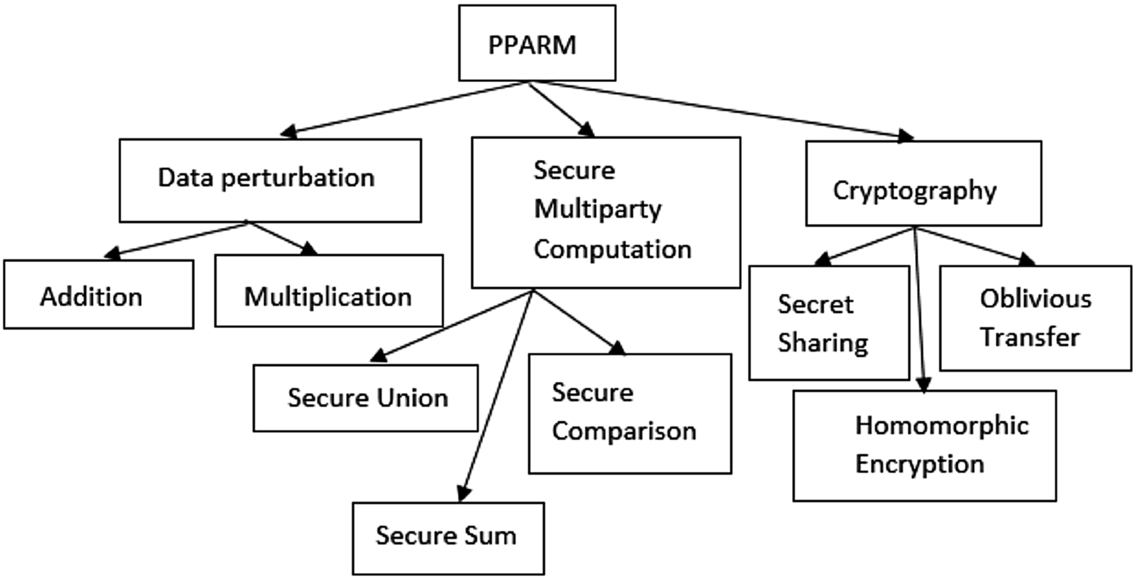

Figure 2. Classification of existing privacy-preserving distributed association rule mining approaches.

resistant protocol. Liu *et al* [21] have designed a privacy-preserving association rules mining algorithm in distributed environment, wherein a global hash table is built to prune candidate itemsets in early iteration of mining operation, which increases the efficiency of algorithm. It uses cryptographic techniques to provide the privacy between sites.

Hussein *et al* [22] have presented a privacy-preserving association rules mining scheme on horizontally distributed databases. It uses an efficient communication path to reduce the communication cost. It has two phases. In the first phase, it uses frequent itemset approach for candidate set generation. In the second phase, RSA encryption is used for secure union of all local support counts of an itemset. Here, an attack from combiner is possible since RSA encryption is used by client side to encrypt the messages, and initiator and combiner can collude to reveal the local support of itemset of clients. Gurevich and Gud [23] have proposed a new framework for privacy-preserving data mining in distributed environment. It considers two separate sites called minor and calculator, which do not have any database part, wherein authors have proposed three algorithms: one for horizontally distributed database; one for vertically distributed database and one for mixed partition database.

Mathews and Manju [24] have presented an extended distributed r*k* (random *k*, where *k* is the number of sites)-secure sum protocol for privacy-preserving and apriori algorithm for mining frequent itemsets. It provides the privacy by preventing site collusion through changing the position of participating sites. Chaturvedi and Gawande [25] have presented two secure multi-party algorithms. The first algorithm computes the secure union of local subsets held by involving sites. The second algorithm tests whether an element held by a site is included in a subset held by another site. Wang *et al* [26] have designed an algorithm for horizontally partitioned databases. It uses the secure set union in first phase for secure union of locally large itemsets, which prevent identifying the source of itemset. It uses Frequent Data Mining (FDM) algorithm [27] to reduce the communication overhead. In the second phase, it uses secure sum method for calculating the global support count securely. Here, sites can collude to reveal the support of other site. Juan and Yanqin [28] have proposed an efficient association rule mining algorithm for distributed databases, which is based on distributed oblivious transfer protocol. It provides prevention from sites collusion but does not provide prevention from external adversary attack.

Lakshmi and Rani [29] have designed a protocol based on double hash function for finding global association rules in horizontally distributed databases securely. This protocol prevents site collusion but it has high communication overhead. Nguyen *et al* [30] have proposed an improved scheme for privacy-preserving association rules mining for horizontally partitioned databases. It has two phases. In the first phase, it uses maximal frequent itemset algorithm [31] to minimize the communication cost. In the second phase, it uses homomorphic-encryption-based public key cryptosystem [32] to compute global support count of itemsets. Combiner and initiator used in this protocol can collude to compromise the privacy of client sites. It lacks in offering authenticity and integrity of messages exchanged among participating sites.

Patel *et al* [33] have proposed an algorithm for securely deriving global association rules in horizontally partitioned databases, which provides security against external adversary for participating sites using Elliptic-Curve integrated encryption scheme. It offers authentication between participating sites using Elliptic-curve-based digital signature algorithm. Here, the communication cost for sending $(2^m - 1)$ messages to n sites is $O(n \times 2^m)$.

Modi *et al* [34] have presented an efficient approach for deriving global association rule in horizontally partitioned databases. It uses the notion of onion routing protocol for dynamic communication path among participating sites. Here, encrypted messages (using Elliptic-curve cryptography) are sent to other sites and thus receiver site cannot see the original contents of sender side and onion routing provides dynamic communication path. Therefore, trusted third party is unable to find communication path; thus, it cannot compromise the privacy of individual site information. In addition, it offers privacy against external adversary attack.

Sari [35] have presented the way of selection of key generating curve based on computational cost for Elliptic-curve cryptography used in privacy-preserving association rule mining.

From the literature survey, we observed that the existing approaches consider privacy issues only in the context of participating sites and communication channel between participating sites is assumed as secured. However, in real life, it is not always true. An external adversary attack may affect the global mining result and privacy by monitoring the communication channel between participating sites. Thus, there is a need for protecting information privacy against unauthorized entities. In addition, existing approaches suffer from high computational and communication cost.

3. Proposed protocols

We propose two protocols for securely generating global association rules in horizontally distributed databases. The first protocol is based on Elliptic-curve cryptosystem, while the second protocol is based on Shamir's secret sharing scheme.

3.1 *Design goals*

The privacy-preserving distributed association rule mining should satisfy the following security and privacy requirements in unsecured communication channel among participating sites:

- any site should not be able to learn anything about the other involving sites,
- by monitoring the communication channel between involving sites, adversaries should not be able to affect the privacy and security of the messages exchanged and global mining result and
- it should have low computational and communication cost

3.2 *Proposed protocols: background*

3.2a *Elliptic-curve-based Paillier public key cryptosystem scheme*: Elliptic-curve cryptography (ECC) uses points on an elliptic curve to generate key bits required for encryption and decryption processes. It meets the security standards with much smaller key size than that of the other systems [36]. The comparison of key size used in ECC, RSA and Diffie-Hellman systems are given in [35, 37]. It saves significant computation time and memory space. Here, we present a probabilistic and homomorphic Paillier public key cryptosystem [6] that is based on Elliptic-curve over ring. It has the following phases.

Phase 1: key generation process

- Select two odd prime numbers p and q and calculate $N = pq$.
- Select a random elliptic curve $E : y^2z = x^3 + axz^2 + bz^3$ over $\mathbb{Z}/N\mathbb{Z}$ where $gcd(N, 6(4a^3 + 27b^2)) = 1$.
- Calculate $M = LCM(\#E(F_p), \#E(F_q))$.
- Select a point $Q = (x : y : z)$ that has an order dividing M in $E(\mathbb{Z}/N^2\mathbb{Z})$. This point can be found by choosing a random point $Q' = (x' : y' : z')$ and setting $Q = NQ'$.
- Public key is $(N, Q, (a, b))$ and secret key is M.

Phase 2: encryption process

- Assume we want to encrypt a message m; first get correct public key of receiver site where $m \in \mathbb{Z}/N\mathbb{Z}$.
- Select a random integer K that lies in the range $1 \le K < N$ and calculate the point

$$S = KQ + P_m$$

where $P_m = (mN : 1 : 0)$ and send this point to respective receiver.

Phase 3: decryption process

- Receiver computes $MS = K(MQ) + MP_m = P_mM = (mMN : 1 : 0)$.
- Divide the x-coordinate value by N and multiply by the inverse of $M \bmod N$ for the point calculated earlier, which gives the original message m where $m \in \mathbb{Z}/N\mathbb{Z}$.

This cryptosystem has additive homomorphic property. Let m_1, m_2 be two messages. Then encryption: $E_k(m_1 + m_2) = E_k(m_1) * E_k(m_2)$; decryption: $D(E_k(m_1) * E_k(m_2)) = m_1 + m_2$, i.e., the sum of m_1 and m_2 can be computed without revealing m_1 and m_2. The encryption in this cryptosystem is probabilistic, i.e., it will generate different ciphertexts for the same plain text.

3.2b *Shamir's secret sharing scheme*: Shamir's secret sharing scheme [7] works as follows. Divide the secret S among n sites P_1, P_2, P_3,...,P_n, where each site is having share of secret S_1, S_2, S_3,..., S_n, respectively. Consider a polynomial of degree $(k - 1) = n$ whose constant term is equal to the secret value, where a prime modulus $p \in P$,

$p > S$ and $p > n$; choose random $(k-1)$ random positive coefficients of the polynomial. For reconstruction of secret S, we need k or more shares.

Example: Assume secret $S = 1234$, $p = 1613$. We need the shares from three or more sites to reconstruct the secret. Here, $k = 3$; coefficients of polynomial are 166 and 94; thus, the polynomial is $f(x) = 94x^2 + 166x + 1234 (\text{mod } 1613)$. From this polynomial, generate different shares for each site. When $x = 1$, then $f(1) = 1494$, this (1, 1494) point is given to site P_1. Similarly, we generate different shares (points) for each site; $x = 2$, $f(2) = 329$ then (2, 329) is given to site P_2; $x = 3, f(3) = 965$ then (3, 965) is given to site P_3; $x = 4, f(4) = 176$ then (4, 176) is given to site P_4; $x = 5, f(5) = 1188$ then (5, 1188) is given to site P_5.

Reconstruction phase: For reconstructing the secret S, we need the shares from three or more sites since we have selected $k = 3$. Suppose three shares (1, 1494), (2, 329) and (3, 965) are collected. In this phase, Lagrange polynomial is used for reconstructing the secret from given set of points since it is the least degree polynomial, which is unique.

$$L(x) = \sum_{j=0}^{k} y_j l_j(x)$$

$$l_j(x) = \prod_{\substack{0 \le m \le k \\ m \ne j}} \frac{x - x_m}{x_j - x_m}$$

where

$$l_0 = \frac{(x-x_1)}{(x_0-x_1)}\frac{(x-x_2)}{(x_0-x_2)} = \frac{(x-2)}{(1-2)}\frac{(x-3)}{(1-3)} = \frac{(x-2)(x-3)}{2}$$
$$l_1 = \frac{(x-x_0)}{(x_1-x_0)}\frac{(x-x_2)}{(x_1-x_2)} = \frac{(x-1)}{(2-1)}\frac{(x-3)}{(2-3)} = \frac{(x-1)(x-3)}{-1}$$
$$l_2 = \frac{(x-x_0)}{(x_2-x_0)}\frac{(x-x_1)}{(x_2-x_1)} = \frac{(x-1)}{(3-1)}\frac{(x-2)}{(3-2)} = \frac{(x-1)(x-2)}{2}.$$

Then

$$\begin{aligned} L(x) &= y_0 l_0 + y_1 l_1 + y_2 l_2 \\ L(x) &= \frac{1494(x-2)(x-3)}{2} + \frac{329(x-1)(x-3)}{-1} \\ &\quad + \frac{965(x-1)(x-2)}{2} \\ L(x) &= 94x^2 + 166x + 1234. \end{aligned}$$

In this way, we can reconstruct the secret 1234.

3.3 *Proposed protocol 1: proposed protocol based on Elliptic-curve cryptosystem*

We have used the Elliptic-curve-based Paillier public key cryptosystem since it requires shorter key length and provides the same level of security as presented earlier. We sign the message with the help of secret key of Elliptic-curve-based Paillier public key cryptosystem before sending it to other sites. This helps in validating the integrity and authenticity of a message.

The proposed protocol works as follows. Consider a database DB distributed among n sites $site_1$, $site_2$,...,$site_n$ in such a way that $site_i$ containing database DB_i has the same number of attributes as that of other sites but a different number of transactions. Here, all involving sites are considered as semi-honest. As shown in figure 3, consider 4 sites $Site1$, $Site2$, $Site3$ (combiner) and $Site4$ (miner) containing the databases DB_1, DB_2, DB_3 and DB_4 respectively. Here, certificate authority does not have any database part and generates the Elliptic-curve based Paillier public and secret keys for all the involving sites. Here, we see that the homomorphic property of Elliptic-curve-based Paillier cryptosystem helps find the global count of an itemset securely.

Lemma 1 *For an itemset P that belongs to* $(n-1)$ *sites, the global support count can be derived as follows.*

Encryption: $E(P.sup_1 + P.sup_2 + \cdots + P.sup_{n-1}) = E(P.sup_1) * E(P.sup_2) * \cdots * E(P.sup_{n-1})$.

Decryption: $D(E(P.sup_1) * E(P.sup_2) * \cdots * E(P.sup_{n-1})) = P.sup_1 + P.sup_2 + \cdots + P.sup_{n-1}$.

After the decryption process, the result will be equal to the sum of support counts of itemset P at $(n-1)$ sites.

A flow of the proposed protocol is given in figure 4.

Lemma 2 $\bigcup_{i=1}^{n} MFI_{d_i}$ *determines all global frequent itemsets* [38]. *Note*: *MFI is the maximal frequent itemset.*

The proposed communication protocol 1 works in three phases as follows.

Phase 1: maximal frequent itemset generation from all sites

- First, certificate authority generates the public and secret keys for all the involving sites with the help of an Elliptic-curve Paillier public key cryptosystem. Consider (PK_1, SK_1), (PK_2, SK_2), (PK_3, SK_3) and (PK_4, SK_4) to be the generated key-pairs for $Site_1$, $Site_2$, $Site_3$ (combiner) and $Site_4$ (miner), respectively. Distribute the secret keys to respective sites and public keys of each sites to all other sites. Now, each site has its own public-secret key pair and public keys of all other sites.
- Each site computes local maximum frequent itemsets; later each site encrypts the local maximum frequent itemsets using the public key of miner PK_4, and signs the encrypted message with its own secret key. This encrypted and signed information is sent to the combiner.
- Combiner receives all the signed messages from all the sites, and later verifies the integrity and authenticity of signed message through respective public key of sites. It shuffles and combines the received messages with its

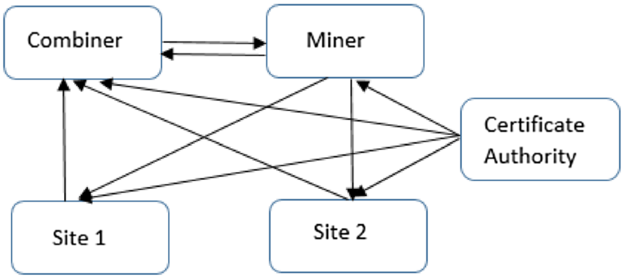

Figure 3. Proposed communication protocol.

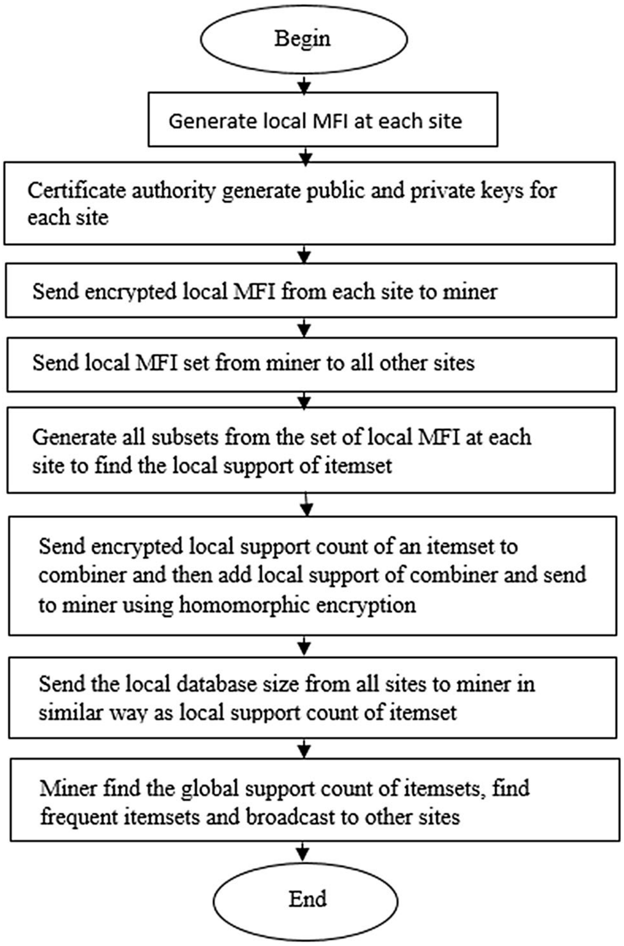
Begin
Generate local MFI at each site
Certificate authority generate public and private keys for each site
Send encrypted local MFI from each site to miner
Send local MFI set from miner to all other sites
Generate all subsets from the set of local MFI at each site to find the local support of itemset
Send encrypted local support count of an itemset to combiner and then add local support of combiner and send to miner using homomorphic encryption
Send the local database size from all sites to miner in similar way as local support count of itemset
Miner find the global support count of itemsets, find frequent itemsets and broadcast to other sites
End

Figure 4. Flow of the proposed protocol.

own encrypted message, signs the combined message through its own secret key SK_3 and later sends it to the miner.

- Miner receives the combined message from combiner and verifies it using public key PK_3 of combiner. Then it decrypts the received message with its own secret key SK_4 and adds its maximum frequent itemset to the decrypted message for computing the global maximal frequent itemset. Then the final global maximal frequent itemsets are shared with all other sites.

Phase 2: calculation of global support of an itemset from all sites

- Each site generates all the nonempty subsets from global maximal frequent itemsets; later each site finds the local support count of a candidate itemset P, encrypts with the miner's pubic key PK_4, signs with its own secret key and later sends it to the combiner. The encryption of the local support count of candidate itemset P at $Site_i$ is denoted by $E(P.sup_i)$.
- For each candidate itemset, combiner computes $E(E(P.sup_1) + E(P.sup_2) + E(P.sup_{combiner})) = E(P.sup_1) * E(P.sup_2) * E(P.sup_{combiner})$. It signs the encrypted message with combiner's secret key SK_3 and later sends this signed message to miner.
- Finally, miner verifies the signed information using the combiner's public key PK_3 and decrypts the received information using its own secret key SK_4. It generates the global support count of each candidate itemset P as follows: $P.sup = P.sup_{miner} + D(E(P.sup_1) * E(P.sup_2) * E(P.sup_{combiner}))$.

Phase 3: calculation of global database size and association rules from globally supported itemsets

- As in phase 2, the global database size is derived by collecting local database size from all the sites. $|DB| = \sum_{i=1}^{n} |DB_i|$ and $|DB_i|$ is database size at $Site_i$.
- After deriving the global support count of all itemsets and global size of database, the miner derives the global association rules using apriori algorithm [9]. These rules are then shared with all the involving sites.

The proposed protocol can securely derive the global association rules since all the information is exchanged after performing encryption and signing. It ensures integrity and authenticity of the sent information. However, it may fail, if miner and combiner collude. To prevent this, we have incorporated Shamir's secret sharing scheme in protocol 2.

3.4 *Proposed protocol 2: finding global support value using Shamir's secret sharing scheme*

To prevent the collusion between combiner and miner, we use Shamir's secret sharing scheme [7]. Like protocol 1, the certificate authority generates public and secret key pair for each site. The certificate authority then distributes the public keys of each site to all other sites and distributes the secret key to respective site except the miner site. It generates a polynomial, in which constant term will be the secret key of miner site. Then it generates different shares of the secret key of miner and distributes them to respective sites. Now each site has one share of secret key of miner site. In protocol 1, if miner and combiner become malicious then they can collude with each other to reveal the local support value of other participating sites. This is prevented using Shamir's secret sharing scheme since miner cannot decrypt the message until it has shares from all sites. For reconstructing the key, miner site needs shares from all sites, then it can decrypt the message. Thus, this approach prevents collusion of miner and combiner.

The proposed protocol 2 works as follows: Consider S_1, S_2,...,S_n sites, which have A_1, A_2,...,A_n local support values of an itemset P, respectively; DB_1, DB_2, DB_3 and DB_4 are the database at each site, respectively. The minimum threshold value is S. In the first step, each site finds the total size of database, i.e., $|DB| = |DB_1| + |DB_2| + |DB_3| + |DB_4|$, using the procedure same as that for calculation of global value of itemset P as follows.

In the second step, each site finds the global value of itemset P. For computing the value $A = A_1 + A_2 + A_3 + \cdots + A_n$ of itemset P securely, each site generates a polynomial of degree k $(k \leq n-1)$. The sites also agree on distinct random values vector $X = (x_1, x_2, \ldots, x_n)$. Each site S_i chooses a random polynomial $p_i(x)$ of degree k, where $p_i(0) = A_i$ and $k = n-1$. Now each site computes the shares of local value for other sites, including itself. Suppose site S_i computes the shares, including itself, as $share(A_i, S_t) = p_i(x)$, where $t = 1, 2, 3, \ldots, n$. Each site sends these shares to respective sites as $share(A_i, S_t)$ to site S_t. Now each site gets the shares $p_1(x), p_2(x), \ldots, p_n(x)$ from the other sites and adds all the received shares to compute $T(x) = p_1(x) + p_2(x) + \cdots + p_n(x)$. This result is sent to other sites.

In the last step, each site generates the linear equations from the polynomial $b_n x^n + b_{n-1}x^{n-1} + b_{n-2}x^{n-2} + \cdots + b_1 x + b_0 = T(x)$ where $X = (x_1, x_2, \ldots, x_n)$ and b_0 is the sum of all local values. Now each site computes the sum of all local values. At the end, each site gets the sum of all local values and the global size of database. To derive frequent itemset, each site checks the condition $A \times |DB| \geq S$. If this condition is satisfied then the itemset is considered as globally frequent. An example illustrating the proposed protocol is given here.

Example Consider four sites S_1, S_2, S_3 and S_4, where each site has local value of an itemset P as $A_1 = 2$, $A_2 = 4$, $A_3 = 6$ and $A_4 = 8$, respectively. Now each site wants to compute $A = A_1 + A_2 + A_3 + A_4$ without revealing their local values to each other. The proposed protocol works as follows.

Step 1: generation of polynomial and shares at each site

- Each site generates a polynomial of degree $k = 3$ where constant term of the polynomial is the local value of an itemset P at that site. All sites also agree on the distinct random values vector $X = (3, 5, 7, 8)$.
- Polynomial generated at sites S_1, S_2, S_3 and S_4 are $p_1(x) = x^3 - 2x^2 + 3x + 2$, $p_2(x) = x^3 + x^2 - 6x + 4$, $p_3(x) = x^3 - 4x^2 - 3x + 6$ and $p_4(x) = 2x^3 - x^2 - x + 8$, respectively.
- Each site generates shares from its own polynomial for all sites, including itself.

Step 2: summation of shares at each site and exchanging it with other sites

- Each site adds the shares which they have received from other sites and exchanges the summation with all other sites (see table 1).

Step 3: generation of linear equations at each site and calculation of global value of itemset

- In this step, each site has the summed shares values. Now each site generates linear equations from the polynomial $b_3x^3 + b_2x^2 + b_1x + b_0 = T(x)$ where $X = (3, 5, 7, 8)$.
- After solving the linear equations, each site gets $b_0 = 20$, which is the sum of all the local values of an itemset P (see table 1).
- Similarly, local database sizes from all sites are shared with other sites.
- Finally, each site gets global size of database and global support value of an itemset. Then, each site can derive global association rules using the apriori algorithm [9].

Thus, each site computes the global value of an itemset without revealing the local values to other sites.

4. Analysis of the proposed protocols

4.1 *Analysis of protocol 1*

In protocol 1, all the sites are semi-honest. All the messages are encrypted with the help of Elliptic-curve-based Paillier public key cryptosystem. Due to probabilistic encryption property of this cryptosystem, it generates different ciphertexts for a same plain text. For this reason, it is very difficult for the combiner to distinguish that a same message is encrypted multiple times. Thus, an attack from combiner is prevented.

In the proposed protocol 1, the encrypted message is signed with the help of secret keys of message-originating sites. Thus, it prevents external adversary attack. Using homomorphic property, the miner derives global support count of itemsets correctly without any modification to data. Thus, accuracy of the final result is achieved.

Suppose there are n sites participating in mining operation. The cost for sending the local maximal frequent itemset from $(n-1)$ sites to miner is $(n-1)$. The cost for sending the set of local maximum frequent itemset to other sites from miner will be $(n-1)$. If the set of local

Table 1. Shares at different sites from different polynomials.

Shares/polynomials	Site 1 shares	Site 2 shares	Site 3 shares	Site 4 shares
Site 1 polynomial $p_1(x)$	$p_1(3) = 20$	$p_1(5) = 92$	$p_1(7) = 268$	$p_1(8) = 410$
Site 2 polynomial $p_2(x)$	$p_2(3) = 22$	$p_2(5) = 124$	$p_2(7) = 354$	$p_2(8) = 532$
Site 3 polynomial $p_3(x)$	$p_3(3) = -12$	$p_3(5) = 16$	$p_3(7) = 132$	$p_3(8) = 238$
Site 4 polynomial $p_4(x)$	$p_4(3) = 50$	$p_4(5) = 228$	$p_4(7) = 638$	$p_4(8) = 960$
Summation of shares at each site	$T(3) = 80$	$T(5) = 460$	$T(7) = 1392$	$T(8) = 2140$
Exchanging of summation shares with all other sites	$T(3)$, $T(5)$, $T(7)$, $T(8)$			
Linear equations at each site with vector $X = (3, 5, 7, 8)$	$27b_3 + 9b_2 + 3b_1 + b_0 = T(3) = 80$			
	$125b_3 + 25b_2 + 5b_1 + b_0 = T(5) = 460$			
	$343b_3 + 49b_2 + 7b_1 + b_0 = T(7) = 1392$			
	$512b_3 + 64b_2 + 8b_1 + b_0 = T(8) = 2140$			
After solving linear equations, each site gets	$b_3 = 5, b_2 = -6, b_1 = -7$ and $b_0 = 20$			

maximum frequent itemset has m items, the subset of all the maximum frequent itemset will be $(2^m - 1)$. For sending the local support count of all $(2^m - 1)$ itemset by $(n - 1)$ sites to miner, the communication cost will be $(2^m - 1)(n - 1)$. The cost of sending the local database size from $(n - 1)$ sites to miner will be $(n - 1)$. For sending the global association rules to all other sites from miner, the cost will be $(n - 1)$. Thus, total communication cost is $4(n - 1) + (2^m - 1)(n - 1)$. Therefore, total communication cost of protocol 1 is $O(n \times 2^m)$.

4.2 *Analysis of protocol 2*

In the proposed protocol 2, privacy of sites is maintained since each site sends the share of local support value. It prevents collusion with other sites that attempt to reveal the local value of a site. Here, we assume that sites send actual shares to other sites. Therefore, each site gets the same global support value for an itemset. It prevents external adversary attack since even an adversary can see all shares values of all sites and intermediate results generated by each site and results exchanged with other

Table 2. Comparative analysis of the proposed protocols with existing protocols.

Author (year)	Cryptography scheme	Approach	Prevention of external adversary attack	Prevention of site collusion	Communication cost
Wang *et al* [26]	Commutative encryption	Secure set union for hiding the source and secure sum for calculating global support count	No	No	$O(n \times 2^m)$
Juan and Yanqin [28]	Oblivious transfer protocol	Distributed oblivious transfer protocol for global mining	No	Yes	$O(m(u + rv + rw))$
Modi *et al* [34]	Elliptic-curve cryptography	Onion routing for dynamic communication path	Yes	No	$O(n \times 2^m)$
Kantarcioglu and Clifton [20]	Commutative encryption	Secure sum for calculating global support count	No	No	$O(n \times 2^m)$
Veloso *et al* [38]	Not used	Maximal frequent itemset to reduce communication cost and data randomization for privacy	No	No	$O(n \times 2^m)$
Hussein *et al* [22]	RSA cryptosystem	Efficient communication path to reduce communication cost and apriori-Tid is used for mining operation	No	Yes	$O(n \times 2^m)$
Proposed protocol 1	Homomorphic encryption	Elliptic-curve-based Paillier public-key cryptosystem for integrity and authenticity of messages sent over insecure communicational channel	Yes	No	$O(n \times 2^m)$
Proposed protocol 2	Shamir's secret sharing	Shamir's secret sharing scheme to prevent collusion attack	Yes	Yes	$O(n^2 \times 2^m)$

sites, but the adversary does not have distinct random values vector X for calculating the coefficient of sum polynomial. Thus, an adversary is unable to get the sum of all local values.

Suppose there are n sites and maximum size of an itemset is m. The cost for sharing share of $(2^m - 1)$ itemset to $(n-1)$ sites from n sites will be $(n(n-1)(2^m - 1)$. The cost for sending intermediate results to $(n-1)$ sites from n sites will be $(n(n-1)(2^m - 1))$. Thus, total cost will be $2n(n-1)(2^m - 1)$. Therefore, the communication cost of protocol 2 is $\mathrm{O}(n^2 \times 2^m)$.

4.3 *Comparative analysis*

Comparative analysis of the proposed protocols with existing approaches is given in table 2.

5. Conclusions

In this paper, we have proposed two protocols for privacy-preserving distributed association rule mining in unsecured environment. In the first protocol, a digital signature based on Elliptic-curve-based Paillier public key cryptosystem helps in achieving the privacy of individual site's information against involving sites and an external adversary. However, the collusion between two sites may reveal the other sites' information. The second protocol addresses this limitation by applying Shamir's secret sharing, which helps in preventing the collusion. These protocols can be used to securely derive global association rules in horizontally distributed databases, while fulfilling the privacy needs of distributed association rule mining. The analysis of the proposed protocols is very encouraging.

References

[1] Agrawal R and Srikant R 2000 Privacy-preserving data mining. In: *Proceedings of the ACM SIGMOD International Conference on Management of Data. ACM SIGMOD Rec.* 29(2): 439–450

[2] Evfimievski A, Srikant R, Agrawal R and Gehrke J 2004 Privacy preserving mining of association rules. *Inf. Syst.* 29(4): 343–364

[3] Rizvi S J and Haritsa J R 2002 Maintaining data privacy in association rule mining. In: *Proceedings of the 28th International Conference on Very Large Data Bases*, pp. 682–693

[4] Goldreich O 1998 *Secure multi-party computation*. Draft V1.4. Rehovot, Israel: Department of Computer Science and Applied Mathematics, Weizmann Institute of Science. http://www.wisdom.weizmann.ac.il/~oded/PSX/prot.pdf

[5] Yao A 1986 How to generate and exchange secrets. In: *Proceedings of the 27th Annual Symposium on Foundations of Computer Science*, pp. 162–167

[6] Galbraith S D 2002 Elliptic curve Paillier schemes. *J. Cryptol.* 15(2): 129–138

[7] Shamir A 1979 How to share a secret. *Commun. ACM* 22(11): 612–613

[8] Agrawal R, Imieliński T and Swami A 1993 Mining association rules between sets of items in large databases. In: *Proceedings of the ACM SIGMOD Conference on Management of Data.ACM SIGMOD Rec.* 22(2): 207–216

[9] Agrawal R and Srikant R 1994 Fast algorithms for mining association rules. In: *Proceedings of the 20th International conference on very large data bases.VLDB J.* 1215: 487–499

[10] Cheung D W, Ng V T, Fu A W and Fu Y 1996 Efficient mining of association rules in distributed databases. *IEEE Trans. Knowl. Data Eng.* 8(6): 911–922

[11] Pohlig S C and Hellman M E 1978 An improved algorithm for computing logarithms over GF(p) and its cryptographic significance. *IEEE Trans. Inf. Theory* 24(1): 106–110

[12] Kilian J 1988 Founding crytpography on oblivious transfer. In: *Proceedings of the 20th Annual ACM symposium on Theory of Computing*, pp. 20–31

[13] Lipmaa H 2005 An oblivious transfer protocol with log-squared communication. In: *Information Security*, pp. 314–328

[14] Gentry C 2009 *A fully homomorphic encryption scheme*. PhD Dissertation, Stanford University. http://crypto.stanford.edu/craig

[15] Chen M S, Han J and Yu P S 1996 Data mining: an overview from a database perspective. *IEEE Trans. Knowl. Data Eng.* 8(6): 866–883

[16] Lindell Y and Pinkas B 2000 Privacy preserving data mining. *Advances in Cryptology CRYPTO, 20th Annual International Cryptology Conference*, pp. 36–54

[17] Lindell Y and Pinkas B 2009 Secure multiparty computation for privacy-preserving data mining. *J. Privacy Confidentiality* 1(1): 1–5

[18] Clifton C, Kantarcioglu M, Vaidya J, Lin X and Zhu M Y 2002 Tools for privacy preserving distributed data mining. *ACM SIGKDD Explor. Newsl.* 4(2): 28–34

[19] Vaidya J and Clifton C 2002 Privacy preserving association rule mining in vertically partitioned data. In: *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 639–644

[20] Kantarcioglu M and Clifton C 2004 Privacy-preserving distributed mining of association rules on horizontally partitioned data. *IEEE Trans. Knowl. Data Eng.* 9: 1026–1037

[21] Liu J, Piao X and Huang S 2006 A privacy-preserving mining algorithm of association rules in distributed databases. In: *Proceedings of IMSCCS*, vol. 2, pp. 746–750

[22] Hussein M, El-Sisi A and Ismail N 2008 Fast cryptographic privacy preserving association rules mining on distributed homogenous data base. In: *Proceedings of the 12th International Conference on Knowledge-Based Intelligent Information and Engineering Systems*, pp. 607–616

[23] Gurevich A and Gud E 2006 Privacy preserving data mining algorithms without the use of secure computation or perturbation. In: *Proceedings of the 10th IEEE International Database Engineering and Applications Symposium*, pp. 121–128

[24] Mathews M T and Manju E V 2014 Extended distributed rk-secure sum protocol in apriori algorithm for privacy preserving. *Int. J. Res. Eng. Adv. Technol.* 2(1): 1–5

[25] Chaturvedi G K and Gawande R M 2014 Privacy preserving association rules mining in horizontally distributed databases using FDM and K&C algorithm. *Int. J. Eng. Dev. Res.* 3(3): 263–266

[26] Wang H J, Hu C A and Liu J S 2010 Distributed mining of association rules based on privacy-preserved method. In: *Proceedings of the International Symposium on Information Science and Engineering*, pp. 494–497

[27] Cheung D W, Han J, Ng V T, Fu A W and Fu Y 1996 A fast distributed algorithm for mining association rules. In: *Proceedings of the 4th International Conference on Parallel and Distributed Information Systems*, pp. 31–42

[28] Juan X and Yanqin Z 2010 Application of distributed oblivious transfer protocol in association rule mining. In: *Proceedings of the 2nd IEEE International Conference on Computer Engineering and Applications*, vol. 2, pp. 204–207

[29] Lakshmi N M and Rani K S 2012 Privacy preserving association rule mining without trusted party for horizontally partitioned databases. *Int. J. Data Mining Knowl. Manage. Process.* 2(2): 17–30

[30] Nguyen X C, Le H B and Cao T A 2012 An enhanced scheme for privacy-preserving association rules mining on horizontally distributed databases. In: *Proceedings of the IEEE International Conference on Computing and Communication Technologies, Research, Innovation, and Vision for the Future*, pp. 1–4

[31] Wang H, Li Q, Ma C and Li K 2003 A maximal frequent itemset algorithm. In: *Rough sets, fuzzy sets, data mining, and granular computing*, pp. 484–490

[32] Paillier P 1999 Public-key cryptosystems based on composite degree residuosity classes. In: *Proceedings of Advances in cryptology-EUROCRYPT99*, pp. 223–238

[33] Patel A C, Rao U P and Patel D R 2012 Privacy preserving association rules in unsecured distributed environment using cryptography. In: *Proceedings of the 3rd International Conference on Computing Communication & Networking Technologies*, pp. 1–5

[34] Modi C N, Patil A R and Doshi N 2015 An efficient approach for privacy preserving distributed mining of association rules in unsecured environment. In: *Proceedings of the International Conference on Advances in Computing, Communications and Informatics*, pp. 753–758

[35] Sari R F 2015 Selecting key generating elliptic curves for Privacy Preserving Association Rule Mining (PPARM). In: *Proceedings of the IEEE Asia Pacific Conference on Wireless and Mobile*, pp. 72–77

[36] Anoop M S 2007 Elliptic curve cryptography. In: *An implementation guide*. http://www.infosecwriters.com/text_resources/pdf/Elliptic_Curve_AnnopMS.pdf

[37] NIST 1999 *Recommended elliptic curves for federal government use*. http://csrc.nist.gov/groups/ST/toolkit/documents/dss/NISTReCur.pdf

[38] Veloso A, Meira Jr W, Parthasarathy S and de Carvalho M 2003 Efficient, accurate and privacy-preserving data mining for frequent itemsets in distributed databases. In: *Proceedings of the Brazilian Symposium on Databases (SBBD)*, pp. 281–292