EFPA: Efficient and Flexible Privacy-Preserving Mining of Association Rule in Cloud

Cheng Huang and Rongxing Lu
School of Electrical and Electronic Engineering, Nanyang Technological University, 50 Nanyang Avenue, Singapore 639798
Email: {huangcheng, rxlu}@ntu.edu.sg

***Abstract*—With the explosive growth of data and the advance of cloud computing, data mining technology has attracted considerable interest recently. However, the flourish of data mining technology still faces many challenges in big data era, and one of the main security issues is to prevent privacy disclosure when running data mining in cloud. In this paper, we propose an efficient and flexible protocol, called EFPA, for privacy-preserving association rule mining in cloud. With the protocol, plenty of participants can provide their data and mine the association rules in cloud together without privacy leakage. Detailed security analysis shows that the proposed EFPA protocol can achieve privacy-preserving mining of association rules in cloud. In addition, performance evaluations via extensive simulations also demonstrate the EFPA's effectiveness in term of low computational costs.**

***Keywords*—Big Data, Cloud, Privacy-preserving, Association Rule Mining**

I. Introduction

In our society, with the unprecedented and explosive increase of data, the data mining technology has become an important processing tool to analyse data and summarise mussy data into useful information and knowledge. In the meantime, the limited computational resources may constrain the abilities of mining with big data. Under this circumstance, cloud computing technology was proposed and has become a perfect platform for data mining with less cost. By means of cloud's advanced computing capabilities and great storage space, data mining can be achieved easily and effectively. Among a large amount of data mining algorithms, association rule mining is one of most common algorithms, which can help uncover relationships in seemingly unrelated data. Market basket analysis [1] is one of the famous applications of association rule mining to help supermarkets to find the combinations of products that frequently co-occur in transactions. For example, people who buy flour and sugar, maybe also tend to buy eggs for planning on baking a cake. With this rule $\{flour, sugar\} \rightarrow \{eggs\}$, supermarkets can put the flour, sugar and eggs together to improve the customer shopping experience and stimulate consumption, which would be very promising.

However, although it is convenient to implement complicated association rule mining and analysis by cloud computing, there exist some privacy issues. One of the main security issues is that cloud need to access data providers' valuable data which may leak sensitive information before data mining. Obviously, the privacy disclosure is not expected by data providers. For instance, lots of hospitals have stored their patients' health information and want to apply association rule mining algorithm in cloud, to compute mining results jointly. In this case, each hospital needs to upload their sensitive and crucial patients' data to cloud. If cloud can directly access these data, the privacy of patients and hospitals would be divulged. Therefore, how to handle this privacy problem of association rule mining with big data has become an important research field recently [2–10].

Motivated by the above-mentioned, in this paper, we aim to address the problem of privacy-preserving association rule mining with collaborative data providers in cloud. Although several privacy-preserving association rule mining protocols in cloud have been studied, most of them are using secure multiparty computation, homomorphic encryption or differential privacy to achieve privacy protection in association rule mining. Different from those previously reported ones, based on the fast scale product technique in [11], we propose a novel privacy-preserving association rule mining protocol, called EFPA, which not only enables plenty of data providers to jointly achieve privacy-preserving association rule mining in cloud, but also achieves the properties of efficiency and flexibility at the same time. Specifically, the main contributions of this paper are three-fold.

- Firstly, we present an efficient and flexible privacy-preserving association rule mining protocol, called EFPA. Unlike most existing works, EFPA can support distributed data providers to collaboratively achieve association rule mining without exposing any privacy of data providers or mining results, i.e, the providers' data and mining results cannot be revealed by cloud.

- Secondly, EFPA can provide flexible approaches for privacy-preserving association rule mining in cloud, and guarantee accurate mining results, i.e, no matter what kinds of association rules, cloud can flexibly achieve data mining with different data providers' encrypted data. Different with our protocol, previously homomorphic encryption based protocols can only complete mining the fixed-in-advance association rule. Once association rules change, each data provider needs to re-encrypt raw data and then send encrypted data to the cloud, which is obviously inflexible.

- Finally, our proposed EFPA is further implemented in Java, and we use real UCI's chess data sets [12] to run extensive experiments to validate the efficiency of EFPA in terms of computational cost.

The remainder of this paper is organized as follows. In Section II, we formalize the system model, security model, and identify our design goal. We recall some preliminaries in Section III. We present the detailed design of our EFPA protocol in Section IV, followed by the security analysis

and performance evaluation in Section V and Section VI, respectively. Section VII reviews some related works and Section VIII draws some conclusions.

II. Models and Design Goal

In this section, we formalize our system model, security model, and identify our design goal on privacy-preserving mining of association rule in cloud.

A. System Model

In our system, we focus on privacy-preserving association rule mining in cloud. Specifically, there are four main entities in our system model, namely a trusted authority (TA), a huge number of participants $P = \{P_1, P_2, ..., P_n\}$, a cloud server (CS) and a data center (DC), as shown in Fig.1.

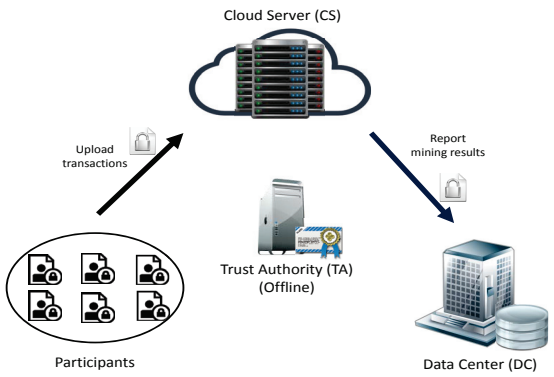


Fig. 1. System model under consideration

Trusted authority (TA): TA is a fully trustable and powerful entity, which is responsible for initializing the whole system and distributing key materials to others in the system.

Participants : Participant is a data provider, and each participant has his/her own data including a set of transactions. Each transaction contains a transaction identity (ID) and a set of items corresponding to attributes. Participants upload their encrypted data to CS for association rule mining.

Cloud server (CS): CS is a core entity. The main function of CS is to perform the privacy-preserving association rule mining with different participants' data, and sends the mining results to DC according to DC's published rules.

Data center (DC): DC's responsibility is to publish the optional association rule and store the results of data mining. i.e. DC publishes the possible and interesting rules to CS, waits for CS to accomplish association rule mining, and obtains the mining results. Typically, DC is a company or other interested party who wants to obtain the association rule of many participants' data.

In our model, TA first initializes the whole system, and participants provide their encrypted data for CS. After receiving participants' data, CS applies the privacy-preserving association rule mining algorithm to compute the mining results according to rules chosen by DC. Finally, CS reports the encrypted mining results to DC. Different from previous models, we consider that the participants possess a myriad of useful data in real world, which is much more than any company's collected data. Participants can take part in association rule mining and provide their data for CS. On the one hand, the participants' data will not be wasted. On the other hand, the mining results can be more accurate. In addition, CS, due to its great computational abilities, can improve the effectiveness of the whole system.

B. Security Model

The security model we consider is *honest-but-curious*. Although participants' data can provide more data for mining and CS can provide great computational power, however, considering that participants' data are sensitive, in general, participants may not expect to disclose their data to CS for association rule mining, i.e., if participants share their data with CS in plain text, CS may faithfully follow the association rule mining protocol, but may be curious and try to disclose participants' information and habits which may leak participants' privacy when the condition is satisfied. In addition, DC does not want CS to obtain the data mining results which may be valuable, and just want to rely on its computational power. Thus, the following security requirements should be satisfied to guarantee the confidentiality of association rule mining results and the privacy of participants in our model.

Privacy: Firstly, CS cannot disclose participants' data even though they obtain a great number of data, i.e. participants can protect their privacy from CS by encrypting their data. Secondly, the participants cannot eavesdrop other participants' data, i.e., one participant cannot disclose other participants' data. Thirdly, DC should keep the mining results secret from CS, i.e. DC can protect their privacy from CS by concealing the mining results. In current model, the collusion attack on privacy is beyond the scope of this work, which is discussed broadly in many privacy-preserving association rule mining protocols [3, 5, 13]. In other words, the participants, CS and DC do not collude with each other in our model.

Authentication and data integrity: The data transmitting in the whole system should be authenticated that they are really generated by the corresponding entities and participants, i.e. if any data is forged, modified and/or replayed by any entity, this malicious behaviour should be detected.

C. Design Goal

Considering the aforementioned system model and security model, our design goal is to propose an efficient and flexible privacy-preserving association rule mining protocol in cloud. Specifically, the following design goals should be satisfied: *i) The security requirements should be guaranteed; ii) The association rule mining should be flexible; and iii) Computation should be efficient.*

III. Preliminaries

In this section, we briefly recall some preliminaries for the construction of EFPA, including the basic algorithm for association rule mining and the bilinear pairings.

A. Basics of Association Rule Mining

Association rule mining [14] is a data mining algorithm that discovers interesting relations among different attributes based on historical data, and the relations are generally presented as an association rule. The basic idea of association rule mining is to calculate the support and confidence to decide whether this rule is strong by comparing with the minimum threshold.

To formalize the basic idea, let $T = \{t_1, t_2, ..., t_i\}$ be the set of all transactions and $A = \{a_1, a_2, ..., a_d\}$ be the set of all attributes. Any subset of A can be represented as A_{sub}. The support count of A_{sub} is defined as the number of transactions in T that contains A_{sub}, i.e, $\delta(A_{sub}) = |\{t|t \in T, A_{sub} \subseteq t\}|$. An association rule is an implication of the form $A_x \rightarrow A_y$, where A_x and A_y are two subsets of A and $A_x \cap A_y = \varnothing$. The strength of this rule can be measured by support and confidence. The support of this rule is defined as follows.

$$Support(A_x \rightarrow A_y) = \frac{\delta(A_x \cup A_y)}{|T|}. \quad (1)$$

Confidence can provide an estimate of the conditional probability of finding attributes of A_y in transactions that contain A_x.

$$Confidence(A_x \rightarrow A_y) = \frac{\delta(A_x \cup A_y)}{\delta(A_x)}. \quad (2)$$

If a minimum support threshold $Support_{min}$ and a minimum confidence threshold $Confidence_{min}$ are given, this rule is strong iff $Support(A_x \rightarrow A_y) \geq Support_{min}$ and $Confidence(A_x \rightarrow A_y) \geq Confidence_{min}$. Here we take

TABLE I. MARKET-BASKET TRANSACTIONS

ID	Bread	Coke	Milk	Beer	Diaper
(1)	1	1	1	0	0
(2)	1	0	0	1	0
(3)	0	1	1	1	1
(4)	1	0	1	1	1
(5)	0	1	1	0	1

the market basket transactions in Table I as an example. The support for the rule $\{Diaper, Milk\} \rightarrow \{Beer\}$ is $\delta(Diaper, Milk, Beer)/5 = 2/5 = 40\%$, and its confidence is $\delta(Diaper, Milk, Beer)/\delta(Diaper, Milk) = 2/3 = 67\%$.

B. Bilinear Pairings

Let $\mathbb{G}$ and $\mathbb{G}_T$ be two cyclic groups of prime order q with the multiplication. Let g be a generator of $\mathbb{G}$ and e be a bilinear map. Let $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ be a bilinear map has the following properties: i) Bilinearity: for all $u, v \in \mathbb{G}$ and $a, b \in \mathbb{Z}_q$, we have $e(u^a, v^b) = e(u, v)^{ab}$; ii) Non-degeneracy: $e(g, g) \neq 1$; and iii) Computability: There is an efficient algorithm to compute bilinear map $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$.

Notice that, in group $\mathbb{G}$, the Computational Diffie-Hellman (CDH) problem is hard, i.e., given g, g^a, g^b for $g \in G$ and unknown $a, b \in \mathbb{Z}_q$, it is intractable to compute g^{ab} in a polynomial time. However, the Decisional Diffie-Hellman (DDH) problem is easy, i.e., given g, g^a, g^b, g^c for $g \in \mathbb{G}$ and unknown $a, b, c \in \mathbb{Z}_q$, it is easy to judge whether $c = ab \bmod q$ by checking $e(g^a, g^b) \stackrel{?}{=} e(g^c, g)$. [15] can provide more detailed description of pairing technique, and complexity assumptions.

Definition 1. *A bilinear pairing generator algorithm* Gen() *can take a security parameter τ as input, and outputs a 5-tuple parameters $(q, g, \mathbb{G}, \mathbb{G}_T, e)$.*

IV. PROPOSED EFPA PROTOCOL

In this section, we propose the EFPA, an efficient and flexible privacy-preserving association rule mining protocol in cloud, which is mainly comprised of four phases: system initialization, encryption of participants' data, privacy-preserving association rule mining in cloud server and decryption at data center. Compared with the state-of-the-art privacy-preserving protocols for association rule mining, EFPA is efficient and flexible to be capable of addressing large data sets for finding optional association rule in cloud.

A. System Initialization

The single trusted authority (TA) takes the charge of bootstrapping the whole system and generating public and private keys for participants, cloud server (CS) and data center (DC). Specifically, TA executes the bootstrap as follows:

• Given the security parameter τ, TA first runs the bilinear pairing algorithm to generate the parameters $(q, g, \mathbb{G}, \mathbb{G}_T, e)$, chooses two different secure cryptographic hash functions $H_1()$ and $H_2()$, where $H_1 : (0,1)^* \rightarrow \mathbb{Z}_q^*$ and $H_2 : (0,1)^* \rightarrow \mathbb{Z}_q^*$, and a secure symmetric encryption algorithm $E()$, such as AES encryption; In addition, TA selects two random numbers $(a, x) \in \mathbb{Z}_q^*$ as the master key, two random elements $(h_1, h_2) \in \mathbb{G}$, and computes $A = g^a$ and $e(g, g)^x$.

• Based on security parameters k_1, k_2, TA chooses two large primes p and α, such that $|p| = k_1$ and $|\alpha| = k_2$.

• Afterwards, TA chooses a random number $v \in \mathbb{Z}_q^*$ as its private key and obtains h, which is a random generator of $\mathbb{G}$, and performs the following steps to distribute secret keys to participants, CS and DC:

- TA computes C_1, C_2 and C_3, where $C_1 = g^v$, $C_2 = A^v h_1^{-v}$ and $C_3 = h_2^{-v}$, and allots them to participants, when the participant registers himself in TA.
- For each registered participant $P_i \in P = \{P_1, P_2, ..., P_n\}$, TA chooses two random parameters $(t_{i1}, t_{i2}) \in \mathbb{Z}_q^*$ and computes the pre-shared key $sk_i = (g^{x+at_{i1}}, g^{t_{i1}}, g^{t_{i2}}, h_1^{t_{i1}} h_2^{t_{i2}})$; then, TA sends the sk_i to the corresponding participant.
- For each registered participant $P_i \in P = \{P_1, P_2, ..., P_n\}$, TA chooses a random number $s_i \in \mathbb{Z}_q^*$ as P_i's private key, computes $S_i = h^{s_i}$ as P_i's public key, and denotes ID_i as P_i's identity; then, TA sends (s_i, S_i, ID_i) to participant P_i.
- For CS, TA also chooses a random number $s_0 \in \mathbb{Z}_q^*$ as CS's private key, and computes $S_0 = h^{s_0}$ as CS's public key, and denotes ID_c as CS's identity.
- For DC, TA first computes $e(g, g)^{xv}$ and assigns it to DC; then, TA chooses a random number $s_d \in \mathbb{Z}_q^*$ as DC's private key, computes $S_d = h^{s_d}$ as DS's public key, and denotes ID_d as DC's identity.

• $< q, g, \mathbb{G}, \mathbb{G}_T, e, p, \alpha, H_1(), H_2(), E(), ID_c, ID_d, S_0, S_d >$ and each $< S_i, ID_i >$, for $P_i \in P$, are published as the system-wide public information finally.

Note that, when a participant registers himself in TA, all registered participants and DC can compute a shared secret key s, which can be used for efficient and flexible privacy-preserving association rule mining in cloud. For each participant P_i, P_i uses sk_i, C_1, C_2 and C_3 to compute

$$\begin{aligned}
&\frac{e(C_1, g^{x+at_{i1}})}{e(g^{t_{i1}}, C_2) \cdot e(g^{t_{i2}}, C_3) \cdot e(h_1^{t_{i1}} h_2^{t_{i2}}, C_1)} \\
&= \frac{e(g^v, g^x g^{at_{i1}})}{e(g^{t_{i1}}, g^{av} \cdot h_1^{-v}) \cdot e(g^{t_{i2}}, h_2^{-v}) \cdot e(h_1^{t_{i1}} h_2^{t_{i2}}, g^v)} \\
&= \frac{e(g^v, g^x) e(g^v, g^{at_{i1}})}{e(g^{t_{i1}}, g^{av}) e(g^{t_{i1}}, h_1^{-v}) \cdot e(g^{t_{i2}}, h_2^{-v}) \cdot e(h_1^{t_{i1}} h_2^{t_{i2}}, g^v)} \\
&= \frac{e(g^v, g^x)}{e(g^v, h_1^{t_{i1}} h_2^{t_{i2}})^{-1} \cdot e(h_1^{t_{i1}} h_2^{t_{i2}}, g^v)} \\
&= e(g, g)^{vx}. \qquad (3)
\end{aligned}$$

The shared secret key can be calculated as $s = H_1(e(g, g)^{vx} | TimeStamp)$, where $TimeStamp$ is the current timestamp. Similarly, DC can also compute the secret key s.

B. Encryption of Participants' Data

Assuming that there are n registered participants $P = \{P_1, P_2, ..., P_n\}$ in the whole system, participant P_i has m_i transactions and l_i attributes for each transaction. For all registered participants, the total number of transactions are N and there are most L attributes, i.e., $\sum_{i=1}^{n} m_i = N$ and $l_i \in [1, L]$. In all l_i attributes, $attr_k$ denotes the kth attribute of a transaction. By the following procedure, participant P_i can securely send encrypted transactions to cloud server for mining association rules, and avoid revealing his/her privacy.

TABLE II. P_i'S ENCRYPTED TRANSACTIONS (R_i)

ID	$attr_1$	$attr_2$	...	$attr_{l_i}$
(1)	c_{i11}	c_{i12}	...	c_{i1l_i}
(2)	c_{i21}	c_{i22}	...	c_{i2l_i}
...	...	...	...	...
(m_i)	c_{im_i1}	c_{im_i2}	...	$c_{im_il_i}$

Step 1: for each transaction $j \in [1, m_i]$, P_i generates a random number r_{ijk} whose length is k_3 for each attribute $k \in [1, l_i]$; then, P_i traverses all l_i attributes and uses the shared secret key s to encrypt the attributes. If $attr_k$ is equal to 1, P_i computes $c_{ijk} = s(\alpha + r_{ijk}) \mod p$. Otherwise, if $attr_k$ is equal to 0, P_i computes $c_{ijk} = s \cdot r_{ijk} \mod p$. Finally, P_i encrypts the transaction as $C_j = \{c_{ij1}, c_{ij2}, ..., c_{ijl_i}\}$.

Step 2: P_i repeats the operations in Step 1 for all m_i transactions, and the encrypted transactions are $R_i = \{C_1, C_2, ..., C_{m_i}\}$, which are shown in Table II.

Step 3: P_i computes the session key for secure communication as $k_{ic} = H_2(S_0^{s_i} | ID_c | ID_i | TimeStamp)$.

Step 4: P_i performs AES encryption using k_{ic} to encrypt R_i as $T_i = E_{k_{ic}}(R_i | TimeStamp)$ and sends it to CS.

C. Privacy-preserving Association Rule Mining in Cloud Server

Before implementing privacy-preserving association rule mining of participants' data, CS asks DC for the association rules firstly, such as $A_x = \{attr_{x_1}, attr_{x_2}, ..., attr_{x_{k'}}\}$, $A_y = \{attr_{y_1}, attr_{y_2}, ...attr_{y_{k''}}\}$ and $A_x \rightarrow A_y$ (e.g. $\{attr_2, attr_3, attr_4\} \rightarrow \{attr_5\}$), where $k', k'' \in [1, L]$ and $A_x \cap A_y = \varnothing$. Based on the rule, CS can compute the support counts (SC) of attributes $A_{xy} = A_x \cup A_y$ and A_x, without exposing participants' privacy. After receiving participants' encrypted data $T_i = \{T_1, T_2, ..., T_n\}$, CS performs the following steps to realize privacy-preserving association rule mining.

Step 1: For each participant's encrypted data T_i, CS computes the session key as $k_{ci} = H_2(S_i^{s_0} | ID_c | ID_i | TimeStamp)$ to decrypt corresponding T_i, and then CS gets $R = \{R_1, R_2, ..., R_n\}$ and $TimeStamp$. Meanwhile, CS counts the number of all participants' transactions as N.

Step 2: CS computes SC'_{ai} and SC'_{bi}, which are the encrypted P_i's support counts of A_{xy} and A_x, as follows.

$$\begin{aligned}
SC'_{ai} = &\sum_{j=1}^{m_i} c_{ijx_1} \cdot c_{ijx_2} \cdot ... \cdot c_{ijx_{k'}} \\
&\cdot c_{ijy_1} \cdot c_{ijy_2} \cdot ... \cdot c_{ijy_{k''}} \qquad (4)
\end{aligned}$$

$$SC'_{bi} = \sum_{j=1}^{m_i} c_{ijx_1} \cdot c_{ijx_2} \cdot ... \cdot c_{ijx_{k'}} \qquad (5)$$

Step 3: CS aggregates all SC'_{ai} and SC'_{bi} to obtain SC'_a and SC'_b which are the all participants' aggregated encrypted support counts of A_{xy} and A_x, as follows.

$$\begin{aligned}
SC'_a = \sum_{i=1}^{n} SC'_{ai} = &\sum_{i=1}^{n} \sum_{j=1}^{m_i} c_{ijx_1} \cdot c_{ijx_2} \cdot ... \cdot c_{ijx_{k'}} \\
&\cdot c_{ijy_1} \cdot c_{ijy_2} \cdot ... \cdot c_{ijy_{k''}} \qquad (6)
\end{aligned}$$

$$SC'_b = \sum_{i=1}^{n} SC'_{bi} = \sum_{i=1}^{n} \sum_{j=1}^{m_i} c_{ijx_1} \cdot c_{ijx_2} \cdot ... \cdot c_{ijx_{k'}} \qquad (7)$$

Step 4: CS computes the session key as $k_{cd} = H_2(S_d^{s_i} | ID_c | ID_d | TimeStamp)$, and use AES encryption to encrypt SC'_a, SC'_b, N as $SC''_a = E_{k_{cd}}(SC'_a | TimeStamp)$, $SC''_b = E_{k_{cd}}(SC'_b | TimeStamp)$ and $E_{k_{cd}}(N | TimeStamp)$; finally, CS reports them to DC.

D. Decryption at Data Center

Upon receiving the results SC''_a and SC''_b, DC performs the following steps to compute the supports and confidence of the corresponding association rule.

Step 1: DC computes the session key as $k_{dc} = H_2(S_c^{s_0} | ID_c | ID_d | TimeStamp)$, and decrypts SC''_a, SC''_b and $E_{k_{cd}}(N)$ to gain SC'_a, SC'_b,N and $TimeStamp$.

Step 2: To decrypt SC'_a and SC'_b, DC first decrypts $SC^*_a = s^{-(k'+k'')} SC'_a \bmod p$ and $SC^*_b = s^{-k'} SC'_b \bmod p$. Whereas, there exists some constraints, when DC tries to decrypt and compute support counts of the rule. We needs $k_1 \gg$

$k_2 \cdot (k'+k'')$ and $k_2 \gg k_3$; then, DC has support counts $SC_a = \frac{SC_a^* - SC_a^* \ (\mathrm{mod}\ \alpha^{k'+k''})}{\alpha^{k'+k''}}$ and $SC_b = \frac{SC_b^* - SC_b^* \ (\mathrm{mod}\ \alpha^{k'})}{\alpha^{k'}}$, so the supports of this rule are $SP = SC_a/N$; finally, confidence can be calculated as $CF = \frac{SC_a}{SC_b}$.

Step 3: According to the minimum support threshold SP_{min} and minimum confidence threshold CF_{min}, DC can judge whether this rule is strong by comparing SP_a with SP_{min} and CF with CF_{min}.

V. Security Analysis

In this section, we analyse the security properties of the proposed EFPA protocol. In specific, our analyses focus on how the proposed EFPA achieves all the security requirements defined earlier.

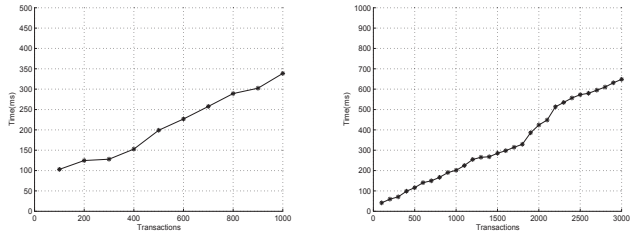
• *Participants' data is privacy-preserving.* For the privacy preservation of participants' data, since each participant P_i's transaction j is $C_j = \{c_{ij1}, c_{ij2}, ..., c_{ijl}\}$, and C_j is one-time masked with random $c_{ijk} = s(\alpha + r_{ijk}) \mod p$ or $s \cdot r_{ijk} \mod p$. r_{ijk} is the random number to ensure that every c_{ijk} is privacy-preserving. Meanwhile, each participant P_i can calculate their own session key k_{ic} with CS, which is different from other participants. When participant P_i sends data to CS, $k_{ic} = H_2(S_0^{s_i}|ID_c|ID_i|TimeStamp)$ can be used as the key to encrypt the communication packages using AES encryption algorithm. Therefore, none of the participants can decrypt the data excepts CS or his/her own. Therefore, the participants' data is privacy-preserving.

• *The data mining results are privacy-preserving.* Note that, only the DC and participants can calculate the shared secret key $s = H_1(e(g,g)^{vx}|TimeStamp)$. If DC and participants do not collude with CS, according to our privacy-preserving association mining algorithm in CS, CS cannot calculate the support counts by $SC_a^* = s^{-(k'+k'')} SC_a' \bmod p$ and $SC_b^* = s^{-k'} SC_b' \bmod p$, then the confidence cannot be computed by CS. Under this circumstance, the data mining results cannot be obtained by CS. In addition, CS's encrypted data mining results are encrypted by session key $k_{cd} = H_2(S_d^{s_i}|ID_c|ID_d|TimeStamp)$, which only can be computed by DC and CS, so other attackers cannot eavesdrop the data and the data mining results are privacy-preserving.

• *The authentication and data integrity have been achieved.* Each entity has his/her public key and privacy key which is assigned by TA. Using the public keys and privacy keys, the session keys can be computed for secure communication. To guarantee authentication and data integrity, the encrypted communication data is $E_k(data|TimeStamp)$ and each entity can verify the decrypted data and compare $TimeStamp$ with current time for authentication and data integrity.

VI. Performance Evaluation

In this section, we evaluate the performance of the proposed EFPA protocol using a custom simulator built in Java. The simulator can support simulations of participants, CS and DS. The performance metrics used in the evaluation are 1) the transactions' encryption time, which indicates the computational costs at participant's side, 2) the time of privacy-preserving association rule mining, which is defined as the computational consumption in CS, and 3) the decryption time in DC, which is used for examining the effectiveness in DC. Specifically, we choose the chess data set [12], including 3196 transactions and 75 attributes. The detailed parameters are set as follows: $\tau = 512$, $k_1 = 2048$, $k_2 = 200$ and $k_3 = 100$, and our experiment environment is a Laptop with 3.1 GHz processor, 8GB RAM, and Window 7 platform.



(a) Participant (b) CS

Fig. 2. Computational costs of participants and CS varying with the number of transactions

Computational costs at participant's side As a participant, firstly, he/she should register himself/herself in TA, and computes the shared secret keys and the session keys. Afterwards, each participant need to encrypt their data by N multiplications in $\mathbb{Z}_p^*$, if there are N transactions for one participant. In our simulator, we set transactions' number from 100 to 1000 for one participant in our simulator, and the average encryption time for different transactions are shown in Fig. 2(a). From the figure, the participant's encryption time is less than 400 ms when the number of transactions is 1000. Thus, the computation at participant's side is efficient.

Computational costs in CS CS first needs to register itself in TA, and computes the session keys. After that, CS receives the encrypted transactions from different participants and applies privacy-preserving association rule mining algorithm. In our simulator, to mine the association rule $\{attr_2, attr_4\} \rightarrow \{attr_6\}$, we assume that there are total transactions from 100 to 3000 generated by 100 participants, i.e., we split chess data sets randomly and assign them to 100 participants, and each participant has different transactions. The computational cost in DS is shown in Fig 2(b), and the results show that, with total 3000 transactions from different participants, the efficiency can also be guaranteed. CS only needs almost 700ms to mine one association rule from 3000 transaction which can be accepted.

Computational costs in DC Except for computing the shared secret key and the session key, DC needs to decrypt the encrypted mining results from CS. In our simulator, when receciving encyrpted mining results from CS, we find that the average decryption time is 16.3 ms for decryption 1000 times.

VII. Related Work

In recent years, there are several privacy-preserving protocols proposed for association rule mining, we briefly review some of them [2–10] in this section.

To achieve privacy-preserving association rule mining, several approaches are used for privacy-preserving association rule mining, such as secure scalar product protocol, secure multiparty computation, bloomfilter, homomorphic encryption and differential privacy. For example, based on secure scalar

product protocol, Vaidya et al. [16] propose a classic secure two-party privacy preserving association rule mining methods to address vertically partitioned data using a secure scalar product protocol. Their method is efficient but just supports two-party privacy-preserving association rule mining and the privacy of one party may be disclosed by brute force method. Based on secure multiparty computation, Kantarcioglu et al. [13] propose privacy preserving association rule mining method for horizonlly partition data. Although this method is high security with multiparty, but when considering two parties, the method has some difficulties for realising privacy preservation. Based on bloomfilter, Qiu et al. [6] and Kantarcioglu et al. [7] propose some efficient privacy preserving association rule mining protocols. However, the mining results are not accurate due to the characteristics of bloomfilter. Besides, homomorphic encryption and differential privacy has been used by Giannotti et al. [2] and Yi et al. [3] for privacy preserving association rule mining in outsourced transaction or cloud. Different from the above protocols, based on a fast scale product technique in [11], our proposed EFPA protocol supports flexible privacy-preserving association rule mining with collaborative participants in cloud, thus it can provide accurate and efficient data mining results with big data. In addition, even though the CS is honest-but-curious, the CS still cannot obtain each participants' data, and only the DC can decrypt the final mining results.

VIII. Conclusion

In this paper, we have proposed an efficient and flexible protocol for privacy-preserving association rule mining in cloud, named EFPA. With distributed participants and their data, the cloud can achieve privacy-preserving association rule mining without privacy leakage. Detailed security analysis shows that the proposed EFPA is privacy-preserving, i.e., no one can read each participant's data, and only the DC can decrypt the final mining results. In addition, through extensive performance evaluation, we have also demonstrated that the proposed EFPA is efficient in terms of computational costs. In future work, we may take differential privacy technique into consideration and combine it with EFPA to achieve a more secure protocol for association rule mining.

Availability

The Java implementation of the proposed EFPA can be downloaded at http://www.ntu.edu.sg/home/rxlu/project/index.htm.

Acknowledgment

This work was supported by Nanyang Technological University under Grants NTU-SUG (M4081196) and MOE Tier 1 (M4011177).

References

[1] P. Giudici and S. Figini, "Applied data mining for business and industry," *Applied Data Mining for Business and Industry, Second Edition*, pp. i–viii, 2009.

[2] F. Giannotti, L. V. S. Lakshmanan, A. Monreale, D. Pedreschi, and W. H. Wang, "Privacy-preserving mining of association rules from outsourced transaction databases," in *Twentieth Italian Symposium on Advanced Database Systems, SEBD 2012, Venice, Italy, June 24-27, 2012, Proceedings*, 2012, pp. 233–242.

[3] X. Yi, F. Rao, E. Bertino, and A. Bouguettaya, "Privacy-preserving association rule mining in cloud computing," in *Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security, ASIA CCS '15, Singapore, April 14-17, 2015*, 2015, pp. 439–450.

[4] O. A. Wahab, M. O. Hachami, A. Zaffari, M. Vivas, and G. G. Dagher, "*DARM*: a privacy-preserving approach for distributed association rules mining on horizontally-partitioned data," in *18th International Database Engineering & Applications Symposium, IDEAS 2014, Porto, Portugal, July 7-9, 2014*, 2014, pp. 1–8.

[5] Y. Duan, J. F. Canny, and J. Z. Zhan, "Efficient privacy-preserving association rule mining: P4P style," in *Proceedings of the IEEE Symposium on Computational Intelligence and Data Mining, CIDM 2007, part of the IEEE Symposium Series on Computational Intelligence 2007, Honolulu, Hawaii, USA, 1-5 April 2007*, 2007, pp. 654–660.

[6] L. Qiu, Y. Li, and X. Wu, "Preserving privacy in association rule mining with bloom filters," *J. Intell. Inf. Syst.*, vol. 29, no. 3, pp. 253–278, 2007.

[7] M. Kantarcioglu, R. Nix, and J. Vaidya, "An efficient approximate protocol for privacy-preserving association rule mining," in *Advances in Knowledge Discovery and Data Mining, 13th Pacific-Asia Conference, PAKDD 2009, Bangkok, Thailand, April 27-30, 2009, Proceedings*, 2009, pp. 515–524.

[8] A. V. Evfimievski, R. Srikant, R. Agrawal, and J. Gehrke, "Privacy preserving mining of association rules," *Inf. Syst.*, vol. 29, no. 4, pp. 343–364, 2004.

[9] T. Tassa, "Secure mining of association rules inhorizontally distributed databases," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 4, pp. 970–983, 2014.

[10] Y. Saygin, V. S. Verykios, and A. K. Elmagarmid, "Privacy preserving association rule mining," in *12th International Workshop on Research Issues in Data Engineering: Engineering E-Commerce/E-Business Systems, RIDE'02, San Jose, California, USA, February 24-25, 2002*, 2002, pp. 151–158.

[11] R. Lu, H. Zhu, X. Liu, J. K. Liu, and J. Shao, "Toward efficient and privacy-preserving computing in big data era," *IEEE Network*, vol. 28, no. 4, pp. 46–50, 2014.

[12] M. Lichman, "UCI machine learning repository," 2013. [Online]. Available: http://archive.ics.uci.edu/ml

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

[14] R. Agrawal, T. Imielinski, and A. N. Swami, "Mining association rules between sets of items in large databases," in *ACM SIGMOD'93.*, 1993, pp. 207–216.

[15] D. Boneh and M. K. Franklin, "Identity-based encryption from the weil pairing," *SIAM J. Comput.*, vol. 32, no. 3, pp. 586–615, 2003.

[16] J. Vaidya and C. Clifton, "Privacy preserving association rule mining in vertically partitioned data," in *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, July 23-26, 2002, Edmonton, Alberta, Canada*, 2002, pp. 639–644.