Using Homomorphic Encryption to Compute Privacy Preserving Data Mining in a Cloud Computing Environment

Hamza Hammami(✉), Hanen Brahmi, Imen Brahmi, and Sadok Ben Yahia

Faculty of Sciences of Tunis, University of Tunis El Manar, LIPAH-LR11ES14, 2092 Tunis, Tunisia
hamza.hammami@aol.fr, hanen.brahmi@yahoo.fr

Abstract. Cloud computing refers to an information technology infrastructure where data and software are stored and processed in a remote data center, accessible as a service through the Internet. Typical data centers within these fields are large, complex and often noisy. Further-more, privacy preserving data mining is an important challenge. It is required to protect the confidentiality of data sources during the extraction of frequent closed patterns. In fact, no site should be able to learn contents of a transaction at any other site. The work carried out in this paper deals with this problem. In this context, we suggest an approach that combines the extraction of frequent closed patterns in a distributed environment such as the cloud. We aim at maintaining the privacy of the sites during the data mining task in a cloud environment based on homomorphic encryption. The Simulation results and performance analysis show that our mechanism requires less communication and computation overheads. It can effectively preserve data privacy, check data integrity, and ensures high data transmission efficiency.

Keywords: Cloud computing · Privacy · Data mining · Confidentiality · Frequent closed patterns · Homomorphic encryption

1 Introduction

During the last decade, with the standardization of the Internet and the development of broadband networks, the computer world has popularized a new paradigm: *cloud computing*. Indeed, cloud computing brings a lot of benefits for businesses [1] such as: *(i)* rationalization and cost reduction, *(ii)* increased flexibility to the end user, *(iii)* usage billing, *(iv)* more efficient use of internet technology resource, and *(v)* data centers and high-performance storage bases. Thanks to these added-values, the recourse to the cloud is becoming more remarkable. In fact, billions of data are exchanged or stored in virtual spaces. This large volume of collected data is characterized by thousands of recording lines stored in a size of a few gigabytes. However, worsened with this huge volume of data, the privacy issues of data mining techniques have become very painful. In this context, preserving privacy is an important challenge. For example, consider a scenario in which two or more sites owning confidential databases wish to run a data mining algorithm on the union of their databases without revealing any unnecessary

M. Themistocleous and V. Morabito (Eds.): EMCIS 2017, LNBIP 299, pp. 397–413, 2017.
DOI: 10.1007/978-3-319-65930-5_32

information. In this scenario, it is required to protect privileged information. Consequently, it is also necessary to enable its use for research or for other purposes. In particular, although the sites realize that combining their data has some mutual benefit, none of them is willing to reveal their database to any other sites.

Hence, the challenge here is: How can we mine the data across distributed sources securely or without disclosing data to others?

This challenge has actually interested a lot of researchers whose primary purpose is to preserve the privacy of data sources during the extraction of frequent closed patterns from a distributed environment by suggesting new protection techniques and approaches.

To tackle this issue, we introduce a novel approach preserving privacy mining called Cloud-PPDM. In this respect, we take into account a main concern, namely maintaining privacy during closed frequent patterns mining in a distributed environment such as the cloud. To do this, we introduce a novel data privacy mining scheme based on homomorphic encryption. The scheme adopts a symmetric-key homomorphic encryption to protect data privacy and combine it with a homomorphic signature to check the integrity of data aggregation. In addition, during the decryption of aggregated data, the master of these sites is able to classify the encrypted and aggregated data based on encryption keys. Our experimental results reveal that the proposed approach is efficient on both runtime performances and security criteria.

The remainder of the paper is organized as follows. In Sect. 2, we describe the related work on privacy preserving data mining. In Sect. 3, we detail some notations that rely on cryptography. Section 4 describes our approach, which can extract frequent closed patterns in a cloud environment while preserving the constraints of privacy by using designed homomorphic encryption. Section 5 gives some tests to illustrate the performance of our approach. Finally in Sect. 6, we summarize our work and we sketch issues of future work.

2 Related Work

Data mining preserving privacy includes a variety of methods to extract useful knowledge from data, without divulging sensitive information on involved individuals. The challenge is to find effective models that meet these constraints. In the following, we survey some work allowing to deal with this problem. Four main categories of Privacy Preserving Data Mining (PPDM) methods have been identified [10–14]:

- Anonymization-based PPDM [15]: The anonymization technique implements generalization and suppression methods to generate an individual record indistinguishable within a group of records.
- Perturbation-based PPDM [16]: In this way, the statistical information computed from the perturbed data does not differ from the statistical information computed from the original data to a larger extent.
- Randomization-based PPDM [17]: The randomization technique implements data distortion techniques for adding little noise in the actual data.
- Cryptography-based PPDM [14]: Cryptographic algorithms are ideally meant for scenarios where multiple parties collaborate to: *(i)* compute results, *(ii)* share non sensitive mining results, *(iii)* and avoid disclosure of sensitive information.

Table 1 summarizes the main advantages as well as the limitations of the (PPDM) techniques.

Table 1. Advantages and limitations of PPDM techniques

Technique	Advantages	Limitations
Anonymization-based PPDM [15]	Hidden identity or sensitive data about record owners	Heavy loss of information
Perturbation-based PPDM [16]	Different independently preserved attributes	Original data values cannot be regenerated
Randomization-based PPDM [17]	Simple and useful for hiding information about individuals	This method does not deal with multiple attribute databases
Cryptography-based PPDM [14]	Better privacy comparing to randomized approach	Heavy calculations (in terms of computation time and memory consumption)

In the following, we only put the focus on the work based on cryptography. The cryptography-based PPDM technique usually guarantees a very high level of data privacy. In [18], the authors addressed the problem of secure mining of association rules over horizontally partitioned data, using cryptographic techniques to secure the shared information. Their solution was based on the assumption that each party would first encrypt its own patterns utilizing commutative encryption, then the already encrypted patterns of every other party. Later on, an initiating party would transmit its frequency count, plus a random value, to its neighbor. The latter would add its frequency count and pass it to other parties. Finally, a secure comparison would take place between the final and initiating parties to determine whether the final result was greater than the threshold plus the random value.

In addition, the authors in [19] dealt with the problem of association rule mining in vertically partitioned data. In other words, its aim was to determine the item frequency when transactions were split across different sites, without revealing the contents of individual transactions. The security of the protocol for computing the scalar product was analyzed.

Furthermore, the authors in [32] put forward an encryption scheme based on substitution cipher techniques in order to preserve the privacy of the transactional data used for outsourcing association rule mining. However, they considered that the association rules mining would be centralized on a single provider, which had to receive the different pattern frequency count and perform all the association rules mining tasks. In contrast, to avoid such overhead imposed on a single provider, the master miner in this scheme would mine the strong association rules on a global level by sending count queries to the data providers while avoiding to store any part of the data locally.

Moreover, the writers in [33] suggested a privacy-preserving model that merged the secure multiparty computation and differential privacy to preserve the privacy of the statistical operations (*i.e.*, count and aggregate count). However, it was not clear how this approach could be applied to handle association rules mining given that the

division operations had to be performed between parties in a secure way in order to validate the minimum support and confidence.

Otherwise, the authors in [34] proposed to tackle the problem of outsourcing the association rule mining task within a corporate privacy-preserving framework by suggesting an encryption scheme based on substitution ciphers called, *RobFrugal*.

In addition, the authors in [20] focused on the use of encryption techniques to build a secure protocol (multi-party computation) to perform this task. The principle of this approach was to use a communication protocol between sites based on asymmetric cryptography arising protocols through solving the discrete logarithm problem. This protocol would ensure anonymity by commutative cryptography, and therefore would guarantee the preservation of the privacy of data owners. This method ensured secure communication while respecting the privacy of sites. However, it did not ensure the integrity of the exchanged data between the sites. In the case of a malicious site, false information may be generated and subsequently sent to the next site. The latter could not detect any modification, leaching the end, to a miscalculation.

Besides, the writers in [21] put forward an approach to transform original data using an encryption function associated with a signature. The key ensured and verified the authenticity of the message and its integrity. This approach was based on homomorphic encryption whose properties allowed performing various operations on encrypted data without knowing the plaintext data.

Added to that, the authors in [22] suggested a method based on Secure Multiparty Computation (SMC). The SMC was a set of cryptographic techniques that permitted the calculation of any function on a set of data distributed among multiple entities. Each entity had a portion of the data. Common calculation had to be done so that neither party could guess, in any manner, the data of other entities from the results and its own data. The limit of this method stood in the fact that communication complexity would exponentially increase as far as the number of distributed sites rose.

In [23], the authors proposed a method based on public key cryptosystems (asymmetric ciphers). A public-key (asymmetric key) based algorithm used two separate keys: a public key and a private one. The public key was utilized to encrypt the data, and only the private key could decrypt the data. A form of this type of encryption was called RSA [24]. It was widely used for secured websites that carry sensitive data such as username, passwords, and credit card numbers. A disadvantage of using public-key cryptography for encryption is speed. There have been other popular secret-key encryption methods, which are significantly faster than any currently available public-key encryption method.

In addition, the authors in [25] put forward an approach based on the Elliptic Curve Cryptography (ECC) [6] and the ElGamal cryptosystem [26]. These approaches would avoid multiple cipher operations on each site in order to ensure secure communication between different sites.

In this respect, there are various advantages and disadvantages of using cryptography techniques to ensure privacy preservation data mining [31]. These advantages and disadvantages are [31]:

- Advantages:
  - Robust
  - Sender and recipient authentication

- Anonymity
- Fairness
- Accountability
- Integrity in storage

– Disadvantages:
  - Taking a long time to figure out the code
  - Overall cryptography as a long process

Generally cryptographic techniques are ideally meant for such scenarios where multiple parties collaborate to compute results or share non sensitive mining results and thereby avoiding the disclosure of sensitive information. However, the major drawback for using cryptography techniques to ensure privacy during the mining task is the execution time. Owing to its usability and importance, preserving the privacy of data in a cloud computing environment still presents a thriving and compelling issue. In this respect, the main thrust of this paper is to propose a novel approach, called Cloud-PPDM, to ensure privacy preserving data mining. Our approach is based on cryptographic techniques in order to improve the performances in terms of execution time. Moreover, the Cloud-PPDM approach relies on mining closed itemsets within a cloud computing environment. The main idea behind our approach comes from the conclusion drawn from the data mining community that focuses on the lossless reduction of itemset mining over cloud computing data. In fact, the extraction of the latter requires less memory and running time. Table 2 summarizes the surveyed approaches dedicated to the cryptography based PPDM.

Table 2. Advantages and limitations of cryptography-based PPDM

Technique	Advantages	Limitations
Kantarcioglu and Clifton [18]	- Incorporating cryptographic techniques to minimize information shared, while adding little overhead to mining task	- Very successfull false information for malicious sites
Vaidya and Clifton [19]	- Efficient method for computing scalar product while preserving privacy of individual values	- Boolean association rule mining - Difficulty to compute scalar product while preserving privacy
Moez et al. [20]	- Anonymity by commutative cryptography - Increased security by asymmetric cryptography	- No integrity of exchanged data between sites - Easily transmitted false information in case of malicious site
Canard et al. [21]	- Anonymity approach for security of respondents identity and decreasing linking attack	- No sufficient protection against attribute disclosure by homogeneous attack and background knowledge attack
Chang et al. [22]	- Safety - Security - Trust-worthiness	- Exponential rising communication complexity with the number of sites

(*continued*)

Table 2. (*continued*)

Technique	Advantages	Limitations
Approaches proposed in [23, 24]	- For public-key cryptosystems, no need for exchanging keys, thus eliminating key distribution problem - No need for private keys to be transmitted or revealed to anyone - Ability to provide repudiated digital signatures	- High execution time public-key cryptosystems
Approaches proposed in [6, 25, 26]	- Preserving privacy, taking advantage of elliptic curve Cryptography and ElGamal cryptosystem	- Poor scalability in terms of dataset size and number of sites
Wong et al. [32]	- High security with low data transformation cost - Secure encryption scheme taking advantage of substitution cipher - Minimization of demands in resources	- One-to-n item mapping cannot be directly applied since it is effectively a one-to-one item mapping
Zhang et al. [33]	- Stronger privacy than current efficient secure multiparty computation approaches - Better accuracy than current differential privacy approaches while maintaining efficiency	- Weakness of direct use of differential privacy in privacy-preserving data mining against collision attack
Giannotti et al. [34]	- Adding weighted support in original item support transactions to reduce fake transaction table and storage overhead - Robustness against guessing attack and man-in-the-middle attack	- This approach is proposed only for information holders; however individual record owners should additionally have the rights and obligations to ensure their own particular private information

3 Cryptography Techniques

In this section, we provide the definition of some notations that rely on the cryptography and secure communication used in our work.

3.1 Homomorphic Encryption

A homomorphic encryption system provides the ability to perform various treatments on encrypted data without using the decryption operation [2]. Furthermore, homomorphic encryption schemes ensure secure aggregation. In fact, they allow data aggregation to be performed on encrypted data. In homomorphic encryption, certain aggregation functions such as the sum and the average can be applied on the encrypted data, reducing, significantly, the workload of the sites in the network. The data is encrypted and sent to

the master site. The last site applies the aggregation function on the encrypted data. The master site receives the encrypted aggregated result and decrypts it. A homomorphic encryption scheme allows arithmetic operations on ciphertexts. These latter are the result of encryption performed on a plaintext using an algorithm, called a cipher. One example is a multiplicatively homomorphic scheme, where the decryption of the efficient manipulation of two ciphertexts yields the multiplication of the two corresponding plaintexts. Homomorphic encryption schemes are especially useful whenever some parties do not have the decryption key(s), while the other parties need to perform arithmetic operations on a set of ciphertexts. In the following, we present a description of the elliptical curve cryptography (ECC) and the signature scheme.

3.2 Elliptic Curve Cryptography

Elliptical Curve Cryptography (ECC) is a public key encryption technique based on elliptic curve theory that can be used to create faster, smaller and more efficient cryptographic keys [6]. The ECC generates keys through the properties of the elliptic curve equation instead of the traditional generation method as a product of very large prime numbers. This technology can be used in conjunction with public key encryption methods, such as the RSA [6] and the Diffie-Hellman [7]. According to some researchers, the ECC can yield a level of security with a 164-bit key, while other systems require a 1024-bit key to achieve the same security level [8]. Mainly, the ECC helps to establish equivalent security with lower computing power and battery resource usage. Consequently, it is becoming widely used for mobile applications.

3.3 Signature Scheme

A signature is a piece of information ensuring authenticity of messages between two parties without any shared secret information in advance [9]. The sender creates the signature by using their private key, while the receiver verifies a signature by using the sender's public key. An aggregate signature scheme is a method for combining n signatures from n different signers on n various messages into a single signature. Indeed, the latter will convince the verifier that the n signers have signed the n original messages. In the next section, we discuss our proposed approach that takes advantage of these cryptography techniques in order to preserve the privacy of data sources during the extraction of frequent closed patterns from a distributed environment such as cloud computing.

4 Cloud-PPDM Approach to Ensure Privacy Preserving Data Mining

In this section, we describe the problem statement. Then we present our Cloud-PPDM approach that is based on two components:

1. The first component uses our proposed Dist-CLOSE algorithm to extract frequent closed patterns with privacy preserving.

2. The second provides a security scheme associated with Dist-CLOSE, in order to ensure privacy concerns. In Algorithm 2, we show the details of this component.

4.1 Problem Statement

The need to ensure the confidentiality of data sources during the extraction of frequent closed patterns from a distributed environment such as the cloud is a hot research topic of data mining community. Each site in this environment has a private transaction database DB_i. The goal is to extract frequent closed itemsets in a distributed environment. In the meanwhile, no $site_i$ should be able to learn: contents of a transaction at any other $site_n$, what patterns are supported by any other site, or the specific value of support for any items at any other site, unless that information is revealed by the knowledge of one's own data and the final result. Furthermore, we are interested in using homomorphic encryption and aggregate signature scheme toolkits to construct a secure multi-party computation protocol to perform this task.

4.2 Background

Along this sub-section, we introduce basic definitions for closed pattern mining, on which our work relies.

Basic Definition 1. (Extraction context) An extraction context is a triplet $\mathcal{K} = (\mathcal{O}, \mathcal{I}, \mathcal{R})$, where $\mathcal{O}$ represents a finite set of objects, $\mathcal{I}$ is a finite set of items and, $\mathcal{R}$ is a binary (incidence) relation (*i.e.,* $\mathcal{R} \subseteq \mathcal{O} \times \mathcal{I}$). Each couple *(o, i)* $\in \mathcal{R}$ expresses that the object $o \in \mathcal{O}$ contains the item $i \in \mathcal{I}$.

Definition 2. (Closure operator) Let $\mathcal{K} = (\mathcal{O}, \mathcal{I}, \mathcal{R})$ be a data mining context, $\mathcal{O}$ a set of transactions, $\mathcal{I}$ a set of items, and $\mathcal{R}$ a binary relation between transactions and items. For O $\subseteq \mathcal{O}$ and I $\subseteq \mathcal{I}$, we define:

$$f(O) = \{i \in I \mid \forall\, o \in O,\ (o, i) \in \mathcal{K}\}$$

$$g(I) = \{\, o \in O \mid \forall\, i \in I,\ (o,\ i) \in \mathcal{K}\}.$$

f(O) associates with *O* the items common to all transactions $o \in O$, and *g*(*I*) associates with *I* the transactions related to all items $i \in I$. The operators $\gamma = \mathrm{f} \circ \mathrm{g}$ and $\gamma' = \mathrm{g} \circ \mathrm{f}$ are the Galois closure operators.

The closure operator γ induces an equivalence relation on the power set of items portioning it into disjoint subsets called equivalence classes [3]. The largest element (*w.r.t.* the number of items) in each equivalence class is called a closed itemset and is defined as follows:

Definition 3. (Closed frequent itemset) An itemset I $\subseteq \mathcal{I}$ is said to be closed if and only if $\gamma(I) = \mathrm{I}$ [4]. The support of I, denoted by *Supp*(I), is equal to the number of objects in $\mathcal{K}$ that contain I. I is said to be frequent if *Supp*(I) is greater than or equal to a user-specified minimum support threshold, denoted Minsup. The frequency of I in $\mathcal{K}$ is equal to $\frac{Supp(\mathcal{I})}{|\mathcal{O}|}$.

4.3 Global Architecture

The Cloud-PPDM allows extracting the frequent closed patterns in a cloud environment while preserving the constraints of privacy by using homomorphic encryption that we have designed. In this respect, the Cloud-PPDM follows the general principle presented in the algorithms that generate the frequent closed itemset such as the CLOSE algorithm [5]. Generally, the steps of the Cloud-PPDM are detailed as follows: Firstly, the initialization process of the communication protocol is invoked. Secondly, the master site, *i.e.* the site which launches the mining task, distributes the list of 1-itemset candidates. Therefore, the different sites run, concurrently, a local algorithm described in Fig. 1, which generates their *closure* and *support*. At this step, the communication protocol is lunched in order to communicate the results to the master site. Now, the master site has at a hand the set of local *closures* as well as local *supports* of the candidate items. The master site can now generate the global *support* by making the sum of local *supports*. The global *closure* is computed by making the intersection of local *closures*. In this way, the master site can generate the candidates of higher size. Then at this level, the master site repeats the above steps whenever it can generate candidates of higher size. Algorithm 1 shows the details of the our proposed approach. In Table 3, we present the definition of some notations used throughout Algorithm 1.

Algorithm 1: Dist-CLOSE: Distributed Extraction of Frequent Closed Itemsets with Privacy Preserving

Input: n: Number of sites; K: Extraction context;
Minsupp: Minimal threshold of support;
master: Boolean *flag* : Set to true if the current site is the master one, otherwise it is set to false;
Begin
  Initialize(n);
  If master **then**
    $FFC_1.generators \longleftarrow \{$ 1-*itemsets* $\}$;
  For $(k \longleftarrow 1; FFC_K.generators \neq \emptyset; k++)$ **do**
    If master **then**
    Distribute(FFC_k, n);
    Receive(FFC_k);
    ${FFC_k}^L \longleftarrow$ *Gen-Local*(FFC_k);
    Communication Protocol (${FFC_k}^L$)
    If master **then**
    ${FFC_k}^G \longleftarrow$ *Collect*(${FFC_k}^L$)
    $FF_k \longleftarrow$ *Gen-global*(${FFC_k}^G$)
    $FFC_{k+1} \longleftarrow$ *Gen-Generator* FF_k;
  Result:$\cup_K FF_k$
End

The *Gen-Local* procedure receives a Frequent Closed Candidates (FFC_k) unit of candidate k-groups containing the *k-generator* candidates of the iteration k in argument. It computes the local *support* and *closure* of each *generator*. This procedure is run on all sites. The *Communication protocol* procedure receives the set of candidates with their *closures* and *supports*. Thus, the *communication protocol* is executed in order to transfer the results to the master site while ensuring privacy preserving. The *Gen-Global* procedure receives the set of FFC_k^L obtained by executing the protocol of communication, and generates the global *support* by making a sum of local *supports*

Table 3. Deftnition of some notations used throughout Algorithm 1

Notation	Deftnition
FF_K	Set of frequent closed itemset of *k*-size
FFC_K	Set of frequent closed itemset candidates of *k*-size
FFC_K^L	Set of local frequent closed itemset candidates *k*-size
FFC_K^G	Set of global frequent closed itemset candidates *k*-size

and the global *closure* by making the intersection between the local *closures* received previously. Then the master site can run the infrequent itemsets given *minsupp*. At this step, the master site executes the *Gen-Generator* procedure to generate the candidates of size $k + 1$ and it returns the set of this candidates. This process will be repeated until the *Gen-Generator* procedure generates an empty set. As a final step, the master site executes a procedure so as to generate a generic base of exact association rules.

4.4 Communication Protocol

The goal of our approach is to extract frequent closed itemsets while ensuring the maintenance of privacy between the various sites. The communication protocol consists of four procedures: (1) Setup, (2) Encrypt-Sign, (3) Aggregate, and (4) Verify. The **Setup** procedure is to prepare and install necessary secrets for the master and each site. When a site s_i decides to send sensed data to its site s_{i+1}, it performs the **Encrypt-Sign** and sends the result to the site. Once the site s_n receives all results from its sites, it activates the **Aggregate** to the received data, and then sends the final results (aggregated ciphertext and signature) to the master. The last procedure is **Verify**. First, the master site extracts the individual sensed data by decrypting the aggregated ciphertext. Afterwards, the master verifies the authenticity and integrity of the decrypted data based on the corresponding aggregated signature. The details of our approach are detailed as follows:

1. **Setup phase:** For each site s_i, the master generates (Sv_i, Sx_i) by KeyGen procedure 1 based on the approach proposed in [9], where ($Sv_i = v_i$) and $Sx_i = x_i$ (MSpk, MSsk). These keys are generated by KeyGen procedure 2. The latter is based on the approach proposed in [27], where the Master Site public key (MSpk) = (n, g, k) and the Master Site secret key (MSsk) = (p, p_g). After that, the (MSpk) is loaded to s_i for all sites i.
2. **Encrypt-sign phase:** This procedure is triggered when a site s_i decides to send its sensed data to the site s_{i+1}. At the end, the site s_i sends the pair ciphertext and the signature (c_i, ∂_i) to site s_{i+1}.
3. **Aggregate phase:** The Aggregate procedure is launched after the site Aggregator s_n has gathered all ciphertext signature pairs.
4. **Verify phase:** When receiving all the ciphertexts and signatures (C', ∂') from the aggregator site s_n, the master can recover and verify each sensing data via the following steps: First, the master decrypts the aggregate result using its private key. Additionally, the master needs to reverse the mapping from the point on the elliptic curve to the aggregate result. To verify the signature, the master computes a point

on the curve using the received signature, the decrypted aggregate result, and the integer k. If the calculated x-coordinate of the point is the same as *r(x)*, then the signature is verified. The master makes sure that all data are generated by legitimate sites and included in the aggregate. In Algorithm 2, we show the details of the communication protocol.

Algorithm 2: Communication Protocol: privately collect messages from parties

1. **Setup Phase**
   KeyGen procedure 1 :
   For a user, pick random $\mathbf{x} \leftarrow Z_p$, and compute $\mathbf{v}= g^x$. The user's public key is $\mathbf{v} \in \mathbf{G1}$, and secret key is
   $x \in Z_p$
   KeyGen procedure 2 :
   p and q are a large primes
   K, the bit length of prime p
   $n=p^2q$, the modulus $g \in Z/nZ$ s.t. $p|ord p_2(g)$
   $g_p= g \bmod p^2$
   Public-Key:(n,g,k), Secret Key: p, g_p
2. **Encrypt-Sign**
   Encoding : $m \in \{0, 1, ..., 2^{k-2}\}$, a message $r \in Z/nZ$, a random integer $c_i = g^{m \div rn} \bmod n$, a ciphertext
   Signature : $\partial_i= x_i \times h_i$ where $h_i = x_i = H(\partial_i)$
3. **Aggregate Phase**
   Aggregated Ciphertext:
   $C^{'} = \sum_{i=1}^{n} c_i$
   Aggregated Signature:
   $\partial^{'} = \sum_{i=1}^{n} \partial_i$.
   Send the aggregated result $(C^{'}, \partial^{'})$ to the master
4. **Verify Phase**
   When receiving $(C^{'}, \partial^{'})$ from the aggregator site s_n, master can recover and verify each sensing data via the following steps:
   Decryption of $C^{'}$:
   $M^{'}=L(c^{p-1} \bmod p^2)L(g_p^{p-1} mod p^2)^{-1} \bmod p$
   Master obtains $M^{'}$ by decrypting $C^{'}$.
   Master obtains $m^{'}$ from $M^{'}$ through the reverse function rmap():
   $m^{'}= rmap(M^{'})= m_1 + m_2 + ... + m_n$.
   Master obtains each sensing data from $m^{'}$.
   Master site verifies each ∂^i via checking whether the equation
   $e(g,\partial)= \prod_{i=1}^{k} e(v_i, h(m_i))$ holds or not.

5 Evaluation

In this section, we experiment the effectiveness and scalability of our proposed approach. In Subsect. 5.1, we present the security analysis and performance evaluation of our communication protocol. Subsection 5.2 describes the experimental environment and the characteristics of the datasets used to evaluate the performance in this work. Finally, in Sect. 5.3, we describe the experimental results and give analysis.

5.1 Security Analysis

In this section, we illustrate the performances of our approach in terms of integrity, freshness and confidentiality of the exchanged data between all sites. The exchanged data can be exploited by a malicious adversary to violate the confidentiality of the sensitive data. In our approach, we palliate these threats via the encryption phase. Also, to ensure the integrity of all exchanged data, each data message is sent only once from the original source. A signature is attached to each message. The signature is computed using the private key that is only known to the source such that the report cannot be forged when it is kept at other sites.

We use the Elliptic Curve to provide message and aggregate integrity in addition to data confidentiality. Each site is pre-loaded with the appropriate elliptic curve parameters, the master public key and a network wide random integer. The integer is used to generate a new key (k) at set intervals. This ensures that the signatures are additive and secure against attacks. At the start of each round, each site chooses a private key and computes the appropriate public key. Choosing a private key is straightforward and requires the site to pick an integer in the field of the elliptic curve. A new public and private key pair is necessary during each round of processing, because it will only take two signatures for a malicious site to determine another site private key. Clearly, if another message is signed with the same private key, then that signature will not be secure. We add another level of security by signing the message and then encrypting it before sending it to the next level. If a site signs the same message with the same key, then another site can determine the private key. The signature scheme is designed such that all signatures can be combined via a simple arithmetic operation. This makes the amount of work required from a master site very small and thus well suited for Privacy Preserving Data Mining (PPDM). The exchanged data are optimized to work with homomorphic encryption and aggregated signatures. The aggregator site waits for a certain amount of time, and when the aggregator has received data, they will add the ciphertexts, which are the digital signature and the public keys. At the end, the master receives only one exchanged data, which consists of one ciphertext corresponding to the sum of the readings of all sites. Besides, it receives one signature corresponding to the sum of data and the sum of the public keys of all sites. Then the master can decrypt the message and verify its integrity using the sum of signatures and the sum of public keys.

5.2 Test Environment and Datasets

In this paper, all simulation work is done in Java. Our simulation is run on the Amazon EC2 cloud computing platform. To show the performance of our proposed approach, we use High-CPU Medium Instances which have 1.7 GB of memory, 5 EC2 compute units (2 virtual cores with 2.5 EC2 compute units each), 320 GB of local instance storage, and 64-bit platforms. In addition, we select various types of datasets, dense and sparse, from the UCI KDD machine learning repository such as: Mushroom [28], Connect [29], C73D10K [30], and T40I10D100K [35] in our experimentation. Table 4 describes the dataset characteristics.

Table 4. UCI dataset characteristics: nature, number of objects, average size of objects, and number of items

Dataset	Mushroom	Connect	C73D10 K	T40I10D100K
Nature	Dense	Dense	Dense	Sparse
Number of objects	8124	67 557	10 000	100 000
Average size of objects	23	43	73	40
Number of items	127	129	2178	1000

5.3 Results and Analysis

To determine the efficiency of our approach, we measure the processing time consumption of the Cloud-PPDM with regard to the approach proposed in [20] fitting in the same trend and in the same data characteristics. We start with the approach proposed in [20] in order to determine the consumed time, for dense and sparse datasets with a various number of sites equal to three, four, and five.

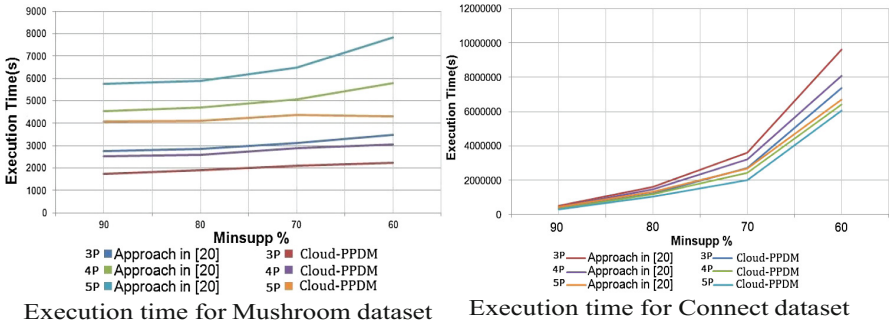

Fig. 1. Execution time of Cloud-PPDM vs approach proposed in [20]

In Figs. 1 and 2, the vertical axis represents the execution time of our Cloud-PPDM approach vs the approach proposed in [20], respectively on the Mushroom and Connect datasets, and the horizontal axis is exploited to present the variations in the execution time according to the number of sites P for various minsup. We note that P represents the number of sites. According to Fig. 1, we can analyze these results as follows: for example, for the Mushroom dataset with a minsup equal to 60% and with a number of sites equal to three sites, the Cloud-PPDM approach requires an execution time equal to 2,218 s to generate the result, while the other approach requires 3,494 s. Furthermore, for the Connect dataset with a minsup equal to 90% and with a number of sites equal to four sites, the execution time passed by the Cloud-PPDM approach to arrive at the result is 324,216 s, whereas the vs. approach passes 453,415 s. We can interpret also through Fig. 1, that the total processing runtime keeps increasing linearly as the number of minsup decreases. This is mainly due to the fact that the calculation time to generate the frequent closed itemsets will increase, when the value of the minsup decreases. In this case, the communication and distribution management time becomes

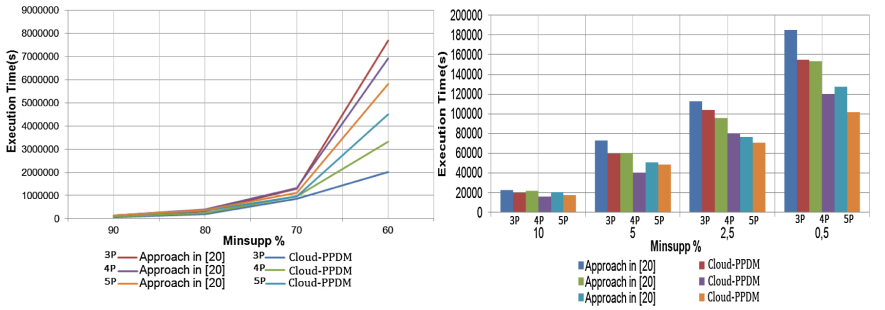

Fig. 2. Execution time of Cloud-PPDM vs approach proposed in [20] respectively on C73D10K and T40I10D100K datasets

negligible with respect to this calculation time. Subsequently, this will remarkably increase the execution time of the algorithm.

In Fig. 2, we show the execution time of the Cloud-PPDM vs the approach proposed in [20], respectively on the C73D10K and T40I10D100K datasets. This figure clearly demonstrates that our approach has the shortest execution time compared to the adversarial approach for each of C73D10K and T40I10D100K datasets. For example, in the case of three sites for the C73D10K dataset with a minsup equal to 80%, our approach requires an execution time equal to 197.614 s, whereas the other approach requires 327, 143 s. Moreover, in the case of five sites for the T40I10D100K dataset with a minsup equal to 0.5%, our approach needs an execution time equal to 100.068 s, while the other approach requires 127.583 s. Otherwise, in the case of the dataset T40I10D100K we can observe a reconciliation between the curves. In this case the execution time varies according to the number of sites. We can notice that if we increase the number of sites for the same threshold, the execution time will decrease.

The total communication cost of the Cloud-PPDM depends on the number of sites. The cost of each run based on the number of items n is as follows: s_1 sends the sensed data to its site s_{i+1}. This latter sends also the sensed data to the next site s_{i+2}. Once the site s_n receives all results from its sites, it will send the final results to the master. Then the communication cost of the Cloud-PPDM is O(n), where n represents the number of sites. Generally, the cost of maintaining privacy depends primarily on the number of sites, the size of the exchanged messages, the number of calls to the communication protocol, and the number of candidates in each iteration.

In this sub-section, we have presented an experimental study (Figs. 1 and 2) on the Cloud-PPDM approache and the one proposed in [20] for the extraction of frequent closed itemsets in a distributed environment while preserving the privacy of data owners. We have performed different tests on the datasets of different types and sizes, to evaluate the performance of our approach with respect to the approach proposed in [20]. According to these experiments, we conclude that the Cloud-PPDM approach mining has the shortest time to ensure privacy mining, compared to the approach proposed in [20].

6 Conclusion

Through this paper, we have introduced a new secure scheme associated with the Dist-CLOSE algorithm which takes advantage of homomorphic encryption. This scheme offers the advantage of carrying out the mining task while guaranteeing security and anonymity. In addition, this scheme protects the confidentiality of data sources during the extraction of frequent closed patterns from a distributed environment such as cloud computing, without revealing information that compromises the privacy of individual sources. In summary, we show that it is possible to achieve good individual security with a communication scheme.

Through extensive experiments carried out on benchmark datasets, we show the effectiveness of our proposed scheme on both runtime performances and security analysis.

Future work will include improving prospects of this approach by strengthening the autonomy of the exchanged data between sites. We plan to give the data the ability to protect itself during the exchange. Hence, the verification calculation by the master site is no longer required in order to ensure safety.

References

1. Muller, S.D., Holm, S.R., Sondergaard, J.: Benefits of cloud computing: literature review in a maturity model perspective. Commun. Assoc. Inf. Syst. **37** (2015). Article no. 42
2. Hayward, R., Chiang, C.C.: Parallelizing fully homomorphic encryption for a cloud environment. J. Appl. Res. Technol. **13**(2), 245–252 (2015). ISSN 1665-6423
3. Bastide, Y., Taouil, R., Pasquier, N., Stumme, G., Lakhal, L.: Mining frequent patterns with counting inference In: KDD Conference, pp. 66–75 (2000)
4. Zitouni, M., Akbarinia, R., Ben Yahia, S., Masseglia, F.: A prime number based approach for closed frequent itemset mining in big data. In: 26th International Conference on Database and Expert Systems Applications, DEXA 2015 Valencia, Spain (2015)
5. Ben Yahia, S., Mephu Nguifo, E.: Approches d'extraction de règles d'association basées sur la correspondance de Galois. Ingénierie des systèmes d'information **9**(3–4), 23–55 (2004)
6. Kumarn, D.S, Suneetha, C.H., Chandrasekhar, A.: Encryption of data using elliptic curve. Int. J. Distrib. Parallel Syst. (IJDPS) **3**(1) (2012)
7. Gajbhiye, S., Karmakar, S., Sharma, M.: Diffie Hellman key agreement with elliptic curve discrete logarithm problem. Int. J. Comput. Appl. **129**(12) (2015). (0975 8887)
8. Moumita, R., Nabamita, D., Jyoti, K.A.: Point generation and base point selection in ECC: an overview. Int. J. Adv. Res. Comput. Commun. Eng. **3**(5) (2014)
9. Boneh, D., Gentry, C., Lynn, B., Shacham, H.: Aggregate and verifiably encrypted signatures from bilinear maps. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 416–432. Springer, Heidelberg (2003). doi:10.1007/3-540-39200-9_26
10. Vassilios, S.V., Elisa, B., Igor, N.F., Loredana, P.P., Yucel, S., Yannis, T.: State of the art in privacy preserving data mining. SIGMOD Rec. **33**, 50–57 (2004)
11. Wang, P.: Survey on privacy preserving data mining. Int. J. Digit. Content Technol. Appl. **4** (9) (2010)

12. Thakur, D., Gupta, H.: An exemplary study of privacy preserving association rule mining techniques. Int. J. Adv. Res. Comput. Sci. Softw. Eng. **3**(11) (2013). P.C.S.T., BHOPAL C.S Dept., India
13. Nithya, C.V., Jeyasree, A.: Privacy preserving using direct and indirect discrimination rule method. Int. J. Adv. Res. Comput. Sci. Softw. Eng. **3**(12) (2013). Vivekanandha College of Technology for Women Namakkal India
14. Lipmaa, H.: Cryptographic techniques in privacy preserving data mining, University College London, Estonian Tutorial (2007)
15. Hussien, A., Hamza, N., Hefny, H.: Attacks on anonymization-based privacy-preserving: a survey for data mining and data publishing. J. Inf. Secur. **4**(2), 101–112 (2013)
16. Li, Y., Chen, M., Li, Q., Zhang, W.: Enabling multilevel trust in privacy preserving data mining. IEEE Trans. Knowl. Data Eng. **24**(9), 1598–1612 (2012)
17. Li, X., Yan, Z., Zhang, P.: A review on privacy-preserving data mining. In: IEEE International Conference on Computer and Information Technology (CIT), pp. 769–774 (2014)
18. Kantarcioglu, M., Clifton, C.: Privacy preserving distributed mining of association rules on horizontally partitioned data. In: ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, pp. 24–31 (2002)
19. Vaidya, J., Clifton, C.: Privacy preserving association rule mining in vertically partitioned data. In: 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 639–644. ACM Press (2002)
20. Moez, W., Poncelet, P., Ben Yahia, S.: A novel approach for privacy mining of generic basic association rules. In: ACM First International Workshop on Privacy and Anonymity for Very Large Datasets, Join with CIKM 2009, France, pp. 45–52 (2009)
21. Canard, S., Desmoulins, N., Devigne, J., Le Hello, D.: Anonymisation des donnèes. Document de travail de l'objet de recherche: trust identity and privacy (2012)
22. Chang, X.-Y., Deng, D.-L., Yuan, X.-X., Hou, P.-Y., Huang, Y.-Y., Duan, L.-M.: Experimental realization of secure multi-party computation in an entanglement access network (2015)
23. Natarajan, R., Sugumar, R., Mahendran, M., Anbazhagan, K.: Design a cryptographic approach for privacy preserving data mining. Int. J. Innov. Res. Sci. Eng. Technol. **1**(1) (2012)
24. Saxena, S., Kapoor, B.: State of the art parallel approaches for RSA public key based cryptosystem. Int. J. Comput. Sci. Appl. (IJCSA) 5(1) (2015)
25. Patel, S.J., Punjani, D., Jinwala, D.C.: An efficient approach for privacy preserving distributed clustering in semi-honest model using elliptic curve cryptography. Int. J. Netw. Secur. **17**(3), 328–339 (2015)
26. Jitarwal, Y., Mangal, P.K., Suman, S.K.: Enhancement of elgamal digital signature based on RSA & symmetric key. Int. J. Adv. Res. Comput. Sci. Softw. Eng. **5**(5) (2015)
27. Okamoto, T., Uchiyama, S.: A new public key cryptosystem as secure as factoring. In: Proceedings of the Annals International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT 1998), pp. 308–318 (1998)
28. ftp://ics.uci.edu/emerz/mldb.tar.Z
29. http://archive.ics.uci.edu/ml
30. ftp://fpt2.cc.ukans.edu/pub/ippbr/census/pumps
31. Rathore, B.S., Singh, A., Singh, D.: A survey of cryptographic and non-cryptographic techniques for privacy preservation. Int. J. Comput. Appl. **130**(13) (2015). (09758887)
32. Wong, W.K., Cheung, D.W., Hung, E., Kao, B., Mamoulis, N.: Security in outsourcing of association rule mining. In: Proceedings of the 33rd International Conference on Very Large Data Bases (VLDB), pp. 111–122 (2007)

33. Zhang, N., Li, M., Lou, W.: Distributed data mining with differential privacy. In: Proceedings of the IEEE International Conference on Communications (ICC), pp. 1–5 (2011)
34. Giannotti, F., Lakshmanan, L., Monreale, A., Pedreschi, D., Wang, H.: Privacy-preserving mining of association rules from outsourced transaction databases. IEEE Syst. J. **7**(3), 385–395 (2013)
35. ftp://fpt2.cc.ukans.edu/pub/ippbr/census/pumps/pumbs90ks.zip