# Compute Privacy Preserving Data Mining in a Cloud Computing Environment with Homomorphic Encryption

Supinder Kaur*, Rajveer kaur **, Parminder Singh ***

*Assistant Professor (Computer Science and Engineering, Rayat Bahra Institute of Engineering and Nano Technology, Email-id: supinder87kaur87@gmail.com)
**Assistant Professor (School of Engineering, Apeejay Institute of Management and Engineering Technical Campus, Email-id: rajveer216@gmail.com)
*** Assistant Professor (School of Computer Science and Engineering, Lovely Professional University, Email-id: Parminder.29695@lpu.co.in)

## ABSTRACT

**Cloud computing is the term used to describe an information technology infrastructure. Processing, in addition to software and data storage, takes place at a remote data center. Data centers for these fields are usually offered as a service over the internet existence of environments that are both large and complex, often accompanied by noise. Protecting privacy through the use of data mining approaches is equally crucial as well. Valuable sources should not be excluded while extracting frequent closed patterns. The ability to gather transaction-related information from any place is known as capability, which necessitates carrying out particular tasks. The proposed methodology in this paper aims to tackle the presented issue with a specific emphasis. In a distributed environment, the extraction of frequent closed patterns is integrated. We endeavor to maintain the confidentiality of site data, particularly when utilizing cloud technology. A mining task in a cloud environment using homomorphic encryption. Our mechanism requires, as indicated by the results of performance analysis and simulation. Reduced communication and computation costs can effectively attain data preservation. Data confidentiality is assured, data completeness is verified, and optimal transfer speeds are promoted.**

**KEYWORDS: CLOUD COMPUTING, PRIVACY, DATA MINING, FREQUENT CLOSED PATTERNS, HOMOMORPHIC ENCRYPTION**

## I. INTRODUCTION

A new paradigm in computing has gained popularity in the last ten years due to the advent of broadband networks and the standardisation of the Internet: cloud computing. Businesses can benefit from cloud computing in several ways [1], including (i) cost savings and rationalisation, (ii) more end-user flexibility, (iii) use billing, (iv) more efficient use of Internet technology resources, and (v) data centres and high-performance storage settings. It is much more crucial to trust the cloud because of these new benefits. In virtual environments, billions of data are stored or shared. Thousands of storage lines packed in gigabytes define this massive amount of data that has been gathered. But because of this massive volume of data, the privacy issues with data mining tools have gotten worse. In this regard, preserving privacy is a big problem. Imagine, for instance, that two or more sites with private databases wish to combine their databases using a data mining method without disclosing any information that isn't absolutely necessary.

In this context, safeguarding sensitive information is essential while also facilitating its utilization for research or similar purposes. Despite recognizing the potential benefits of pooling their data, none of the involved parties are willing to divulge their databases to others. Consequently, the central challenge lies in how to securely extract data from distributed sources without compromising data confidentiality. This challenge has sparked significant interest among researchers focused on safeguarding the privacy of data sources during the extraction of frequent closed patterns in distributed environments. They propose innovative protection techniques and methodologies to address this issue.

We suggest Cloud-PPDM, a novel method designed to protect privacy while mining, as a solution to this problem. Our method places a high priority on protecting privacy in dispersed settings, such as the cloud, when conducting closed, frequent pattern mining. We do this by presenting a brand-new homomorphic encryption-based data privacy mining technique. This approach uses homomorphic signatures to confirm the integrity of aggregated data and symmetric-key homomorphic encryption to protect data privacy. Additionally, site managers can categorise encrypted and aggregated data according to encryption keys throughout the decryption process. The effectiveness of our suggested strategy is shown by experimental results in terms of security metrics and runtime performance.

The rest of the paper is structured as follows. Section 2 provides an overview of related work concerning privacy-preserving data mining. In Section 3, we introduce various notations that are based on cryptographic principles. Our approach, which focuses on extracting frequent closed patterns in a cloud environment while upholding privacy constraints through the use of tailored homomorphic encryption, is outlined in Section 4. Section 5 presents tests conducted to demonstrate the performance of our approach. Finally, Section 6 summarizes our work and outlines potential areas for future research.

## II. BACKGROUND AND RELATED WORK

Privacy-preserving data mining refers to various methods used to extract valuable insights from data while protecting individuals' privacy. The main challenge is to create effective models that respect privacy. We discuss different approaches to tackle this challenge. Four main categories of Privacy Preserving Data Mining (PPDM) methods are recognized:

1. Anonymization-based PPDM: This method uses techniques like generalization and suppression to generate individual records that are indistinguishable within a group.

2. Perturbation-based PPDM: Here, statistical information from perturbed data closely matches that from the original data, minimizing differences significantly.

3. Randomization-based PPDM: This technique distorts data through minimal noise introduction to preserve privacy.

4. Cryptography-based PPDM: Cryptographic algorithms are used when multiple parties collaborate to compute results, share non-sensitive mining outcomes, and prevent the disclosure of sensitive information.

This passage discusses privacy-preserving data mining (PPDM) techniques that allow researchers to extract knowledge from datasets without revealing individual data points.
The focus is on cryptographic methods that guarantee strong privacy. Researchers in [10] tackled association rule mining on horizontally partitioned data. Their engagement involved parties encrypting their data multiple times and sharing counts with added random values. This way, a final count could be reached without revealing the underlying data.
Another study [11] addressed association rule mining on vertically partitioned data. Here, the goal was to find how often items appear together without revealing individual transactions. The authors focused on the security of the method used to calculate a specific mathematical operation. A different approach is presented in [28] where a substitution cipher protects data privacy when mining is outsourced to a service provider. However, this method assumes a centralized system where one party receives all the data and performs all the mining tasks. To avoid overloading this central party, the authors propose a scheme where the data holders only send counts in response to queries, and the mining itself happens at a global level.

Several studies explore advanced cryptographic techniques for privacy-preserving association rule mining. A combination of secure multi-party computation and differential privacy was proposed [29] to protect statistical operations. However, challenges arise when applying it to association rule validation due to the need for secure division operations. Another study[30] introduced RobFrugal, a scheme for outsourced mining based on substitution ciphers. Research by [12] investigates multi-party computation using asymmetric cryptography to achieve anonymity for data owners. While communication is secure and site privacy is respected, this method doesn't guarantee the integrity of exchanged data, leaving it vulnerable to manipulation by malicious participants. A method in [32] utilizes a special type of encryption (homomorphic encryption) to perform calculations on encrypted data while ensuring message authenticity. Finally, [33] proposes a method based on Secure Multiparty Computation (SMC) for distributed data mining. This technique offers strong privacy but suffers from increased communication overhead as the number of participants grows.

This section explores the use of public key cryptography (asymmetric ciphers) in privacy-preserving data mining ([34]). Public key systems use separate keys for encryption and decryption, like the widely used RSA method that secures online transactions. While offering strong security, public key cryptography can be slower than alternative methods.

Another approach leverages Elliptic Curve Cryptography (ECC) and the ElGamal cryptosystem ([13]). These techniques aim to minimize encryption operations at each site, facilitating secure communication.

As discussed in [27], cryptographic techniques for PPDM offer a balance between advantages and disadvantages:

- **Advantages:**
  - Strong data protection
  - Verification of sender and recipient identities
  - Anonymity for data owners
  - Fair and unbiased computations
  - Ability to track responsibility for actions
  - Data integrity during storage
- **Disadvantages:**
  - Increased processing time for decryption
  - Potentially complex cryptographic operations

This section highlights the trade-offs of cryptographic techniques in privacy-preserving data mining (PPDM). While ideal for secure collaboration and hiding sensitive data, cryptography can be computationally expensive. Despite this drawback, ensuring data privacy in cloud environments remains crucial.

Our paper proposes a new approach called Cloud-PPDM, which leverages cryptography for privacy-preserving data mining while optimizing execution time. Cloud-PPDM specifically focuses on mining closed item sets within the cloud. This approach is inspired by data mining research that emphasizes the benefits of lossless reduction techniques for cloud-based mining tasks. Extracting closed item sets, which are maximally condensed datasets, requires less memory and processing power compared to full datasets.

Table 1: Advantages and limitations of cryptography-based PPDM

| Technique | Advantage | Limitations |
|---|---|---|
| Canard et al | Anonymity approach to protect the identity of respondents and reduce link attacks. | Inadequate Safeguard against attribute exposure from homogeneous and background knowledge attacks. |
| Approaches proposed in [ 2, 13, 14] | -Maintaining Confidentiality By leveraging Elliptic curve cryptography and the ElGamal cryptosystem. | Limited scalability regarding dataset size and the number of sites. |
| Zhang et al | -Enhance Privacy compared to existing efficient secure multi-party computation methods.<br>-Improved Precision compared to current approaches based on Differential privacy, all while upholding efficiency. | Vulnerability of directly implementing differential privacy in privacy-preserving data mining to collision attacks |
| Vaidya and Clifton | An efficient way to calculate a scalar product while preserving the privacy of individual values. | -Boolean Association rule mining.<br>-Difficulty to compute scalar product while preserving privacy |
| Giannotti et al | -Introducing weighted support into the original item support transactions to mitigate the proliferation of fake transaction tables and minimize storage overhead.<br>-Enhanced Resilience against guessing attacks and man-in-the-middle attacks. | -This access is recommend only for data owners; however individual registration holders should have additional rights and responsibilities to protect certain private information. |
| Approach proposed in | -In a public-key cryptosystem, there is no | Public-key cryptosystems |

| | | |
|---|---|---|
| [34, 35 ] | requirement for exchanging keys, thereby resolving the key distribution challenge.<br>-Private keys are on necessary to be transmitted or disclosed to any party.<br>-Capable of Offering reputable digital signatures. | characterized by lengthy execution times |
| Moez et al | -Anonymity With commutative cryptography<br>-Increase security with Asymmetric cryptography | Lack of data Integrity between Sites vulnerability To transmitting false information in the event of a malicious site |
| Kantarcioglan d Clifton | Incorporating Cryptographic techniques to minimize the information shared while adding a bit more to the mining task | Very successful Disinformation malicious websites |
| Wong et al | Robust security minimal data transformation expense.<br>-Secure Encryption Method leveraging substitution Cipher techniques.<br>-Reduction of resource requirements toa minimum | Directly applying one-ton item mapping isn't feasible since it essentially represents a one-to one item mapping |
| Chang et al | - Safety<br>- Security<br>- Reliability | Communication Complexity grows exponentially with the number of sites. |

## III. IMPLEMENTATIONS

### 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

Homomorphic encryption systems have the potential to perform operations on data that are encrypted but do not have to be decrypted. Such a method helps ensure secure aggregation, thus allowing for direct data aggregation on encrypted data. For example, by applying aggregation functions such as summation or average on encrypted data, one can greatly reduce the burden of work placed upon network nodes. During this process, information is encrypted and sent to the main center. The information acquired is encoded on the last page. After that, the data is decrypted by the original sender in the encoded form. This is a kind of encryption system called homomorphic encryption. It involves additions and multiplications on encrypted data without the keyholder having access to unencrypted data. An example should be given concerning the way multiplicatively homomorphic calculations occur: decryption is simply a multiplication of plain text values corresponding to each other multiplied encrypted text together. When some parties don't have decryption keys and others want to work on a group of ciphertexts, these systems are very useful. Here comes a description of elliptic curve cryptography also known as ECC in conjunction with its signature schemes.

### 3.2 Elliptic Curve Cryptography

Elliptic Curve Cryptography (ECC) is a public key encryption technique based on elliptic curve theory that makes it easier to generate cryptographic keys that are quicker, smaller, and more effective. ECC makes use of the characteristics of elliptic curve equations, in contrast to the traditional method that depends on huge prime number products for key generation.

This approach works in unison with Diffie-Hellman and RSA, two more public key encryption techniques. According to certain research, ECC can provide a security level comparable to a 164-bit key, whereas other systems require a 1024-bit key to provide a comparable level of protection. ECC is becoming more and more popular for mobile apps because of its primary benefit of establishing comparable security levels with less processing power and battery resource consumption.

### 3.3 Signature Scheme

With its foundation in elliptic curve theory, Elliptic Curve Cryptography (ECC) is a public key encryption technique that makes it easier to generate cryptographic keys that are quicker, smaller, and more effective. Using the characteristics of elliptic curve equations, ECC generates keys differently than the traditional method, which depends on huge prime number products. Other public key encryption techniques like RSA and Diffie-Hellman are easily integrated with this technology. In contrast to other systems that require a 1024-bit key for comparable security assurance, some research indicates that ECC can reach a security level equivalent to a 164-bit key. The main reason ECC is becoming more and more popular for mobile apps is that it provides equivalent security levels with less computational overhead and battery usage.

## Cloud-PPDM Approach to Ensure Privacy-Preserving Data Mining

This section dives into the core of our paper. We'll first define the problem we're addressing, followed by a detailed explanation of our Cloud-PPDM approach. Cloud-PPDM consists of two crucial components:

Privacy-Preserving Frequent Pattern Extraction: This initial component utilizes our new Dist-CLOSE algorithm to extract frequent closed patterns while ensuring data privacy.

Security Scheme for Dist-CLOSE: The second component provides a complementary security mechanism that works alongside Dist-CLOSE to further address privacy concerns. The specifics of this component are outlined in the Algorithm.

## Problem Statement

In this setting, each participant (site) holds a private database of transactions. The goal is to find frequently closed item sets across these distributed databases without revealing any sensitive information. This includes:

- The contents of transactions at other sites
- The specific item sets discovered by other sites
- The exact support values for items at other sites (unless this information can be inferred solely from a participant's data and the final result)

The research community is particularly interested in leveraging homomorphic encryption and secure aggregate signature schemes to build a secure multi-party computation protocol for this task.

## Background

Within this subsection, we introduce fundamental definitions pertinent to closed pattern mining, which form the foundation of our work.

Basic Definition 1: (Extraction Context)

An extraction context is represented as a triplet K = (O, I, R), where:

O denotes a finite set of objects.

I signify a finite set of items.

R denotes a binary (incidence) relation, i.e., $R \subseteq O \times I$. Each pair $(o, i) \in R$ signifies that the object $o \in O$ contains the item $i \in I$.

Definition 2: (Closure Operator)

Let K = (O, I, R) be a data mining context, with O as a set of transactions, I as a set of items, and R as a binary relation between transactions and items. For $O \subseteq O$ and $I \subseteq I$, the closure operator is defined as follows:

- $f(O) = \{i \in I \mid \forall o \in O, (o, i) \in R\}$
- $g(I) = \{o \in O \mid \forall i \in I, (o, i) \in R\}$
- According to the given context:
  All transactions in O share common items, which are associated with set O by f(O).
  The transactions associated with each item in I are associated with set I by g(I).
  These operators are called Galois closure operators: $c \circ f \circ g$ and $c_0 \circ g \circ f$.
  In order to divide the power set of items into disjoint subsets known as equivalence classes, the closure operator c produces an equivalence relation on the set. Closed items are the largest element (in terms of the number of items) in each equivalency class, and they are described as follows:
  Definition #3: (Repeatedly closed) If and only if c(I) = I is an item $I \subseteq I$ taken as closed. Supp(I), the symbol for the support of I, is equal to the number of objects in K that include I. If Supp(I) is more than or equal to a user-specified minimal support level, represented by Minsup, then I am considered frequent. Supp(I) divided by the total number of

objects in K, represented by |O|, is the frequency of I in K.

## Global Architecture

The Cloud-PPDM facilitates the extraction of frequent closed patterns within a cloud environment while upholding privacy constraints through the utilization of our designed homomorphic encryption. In this regard, the Cloud-PPDM adheres to the general principle outlined in algorithms that generate frequent closed item sets, such as the CLOSE algorithm.

The steps of the Cloud-PPDM are outlined as follows:

1. Initialization: The communication protocol is initialized.
2. Distribution of 1-item candidates: The master site, responsible for launching the mining task, distributes the list of 1-item candidates to the different sites.
3. Local algorithm execution: Each site concurrently executes a local algorithm; generating their closures and supporting the communication protocol initiated to transmit the results to the master site.
4. Aggregation of results: The master site receives the set of local closures and local supports of the candidate items. It calculates the global support by summing up local supports and computes the global closure by intersecting local closures.
5. Generation of candidates of higher size: Using the aggregated information, the master site generates candidates of higher size. This process is repeated iteratively whenever higher-sized candidates can be generated.

Algorithm 1 provides detailed specifications of our proposed approach, while Table 2 defines the notations used throughout Algorithm 1.

Algorithm 1: Dist-CLOSE: Distributed Extraction of Frequent Closed Item-sets 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
        FFC1.generators ←{ 1-itemsets };
    For (k ←1; FFCK .generators ≠ θ; k + +) do
        If master then
            Distribute(FFCk, n);
            Receive(FFCk);
            FFCk ^L ← Gen-Local(F FCk);
            Communication Protocol (FFCk^L)
            If master then
            FFCk^G ← Collect(FFCk^L)
            FF k ← Gen-global(FFCk ^G)
            FFCk+1 ← Gen-Generator FFk ;
    Result; UK FFk
End
```

Table 2: Definition of some notations used throughout Algorithm 1

| Notation | Definition |
|---|---|
| FFCk | Set of frequent closed item<br>candidates of k-size |
| FFk | Set of frequent closed items of K-size |
| FFC^G | Set of global frequent closed itemset candidates k-size |
| FFC^L | Set of local frequent closed itemset candidates k-size |

The Gen-Local procedure accepts a unit of candidate k-groups (FFCk) comprising the k-generator candidates of the k-th iteration as an argument. It computes the local support and closure of each generator. This procedure is executed across all sites.

The Communication protocol procedure receives the set of candidates along with their closures and supports. Subsequently, the communication protocol is executed to transfer the results to the master site while ensuring privacy preservation.

The Gen-Global procedure receives the set of FFCL obtained through the communication protocol execution. It computes the global support by summing up local support and determines the global closure by intersecting the previously received local closures. Subsequently, the master site can execute the Gen-Generator procedure to generate candidates of size k + 1, returning the set of these candidates. This process iterates until the Gen-Generator procedure produces an empty set.

As a final step, the master site executes a procedure to generate a generic base of exact association rules based on the generated candidates and the specified minimum support threshold.

## Communication Protocol

Our aim is to extract frequent closed itemsets and simultaneously ensure their occurrence regularly through our approach goal is to maintain privacy across different sites through a communication protocol involving four procedures.

1. **Setup**: the master creates( $Sv_i$, $Sx_i$ )using the keygen procedure for each site the approach presented in[ 9] involves setting $Sv_i$ equal to $v_i$ and $Sx_i$ equal to $x_i$ resulting in MSpk procedure 2 of keygen is used to generate the MSsk keys it relies on a particular method for this purpose according to the approach suggested in [27] the public key of the master site MSpk is comprised of three components n,g, and k the secret key for the master site MSsk is represented as p, $p_g$ following this step $s_i$ loads the public key of milk for every site i.

2. **Encrypt-Sign**: the procedure known as the encrypt-sign phase is initiated when the site makes a decision to transmit its data securely data it sensed to the next site $s_{i+1}$ finally $s_i$ transmits both encrypted data and ciphertext as a pair I hereby certify that part I ($C_I$,$\partial_I$ ) belongs to site $s_{i+1}$.

3. **Aggregate**: the aggregate phase starts with the launch of the aggregate procedure following the site aggregator $s_n$.

4. **Verify**: all pairs of ciphertext, and signatures ($C_I$,$\partial_I$ ) have been collected previous phase. The verification process takes place at the aggregator site SN allows the master to retrieve and authenticate every sensory data initially the aggregate result is decrypted by the master using their private key furthermore the mapping from the point on the elliptic must be reversed by the master to validate the signature the master calculates a point on the curve by utilizing both the decrypted aggregate result and received signature if the value of k is an integer then the points x-coordinate is equivalent to r(x) and thus sources the data legitimacy is ensured by the master who verifies that all signatures are genuine inclusion of sites in the aggregate algorithm 2 outlines how we incorporate individual sites into the overall total protocol for communication.

For more detailed information, Algorithm 2 provides a comprehensive breakdown of the communication protocol.

1. Setup Phase

KeyGen procedure 1:

For a user, pick random x← Zp, and compute $v=g^x$. The user's public key is v Є G1, and

the secret key is x Є Zp.

KeyGen procedure 2:

p and q are large primes

K, the bit length of prime p

$n=p^2q$, the modulus g Є Z/nZ s.t. p|$ord_{p2}(g)$

$g_p=g \bmod p^2$

Public-Key: (n,g,k), Secret Key: p, $g_p$

2. Encrypt-Sign

Encoding: m Є {0, 1, ..., $2^{k-2}$}, a message r Є Z/nZ, a random integer $c_i = g^{m\%rn}$ mod n,

ciphertext

Signature

$\partial_i= x_i \times h_i$ where $h_i = x_i = H(\partial_i)$

3. Aggregate Phase

Aggregated Ciphertext:

$C' = \sum^{n}_{i=1}=1\ c_i$

Aggregated Signature:

$\partial'== \sum^{n}_{i=1} \partial_i$.

Send the aggregated result (C',$\partial'$) to the master

4. Verify Phase

When receiving (C', $\partial'$) from the aggregator site $s_n$, the master can recover and verify each

sensing data via the following steps:

The decryption of C':

$M'=L(c^{p-1} \bmod p^2)L(g^{p-1}{}_p\bmod p^2)^{-1} \bmod p$

Master obtains M' by decrypting C'.

Master obtains m' from M' through the reverse function map():

m'=rmap(M')= m1 + m2 + ... + $m_n$.

Master obtains each sensing data from m'.

The master site verifies each $ə^i$ by checking whether the equation

$e(g,\partial) = \boldsymbol{\Pi}^{k}_{i=1}\, e(v_i, h(m_i))$ holds or not.

## Evaluation

This section dives into the evaluation of our Cloud-PPDM approach. Here's how it's structured:

Security and Performance Assessment: This subsection details the methods we used to evaluate the security measures and assess the performance of Cloud-PPDM.

Experimental Setup: This subsection provides specific details about the communication protocol used in our experiments and the characteristics of the datasets we employed to measure Cloud-PPDM's effectiveness.

Results and Analysis: This subsection presents the experimental outcomes and our analysis of those results.

## Security Analysis

This section demonstrates the effectiveness of our approach in upholding integrity. Data between all sites is ensured to be fresh and confidential. Data is vulnerable to exploitation by malicious adversaries that may breach confidentiality. To address potential risks to sensitive information, we mitigate them through the implementation of encryption measures in our approach. Additionally, To guarantee the integrity of all data transmissions, each message is only sent once. Each message is accompanied by a signature that has been calculated. Using the source's private key ensures that the report cannot be accessed by anyone else. When it is kept at other sites, it becomes forged. Ensuring the security of messages and data by utilizing Elliptic Curves. Confidentiality of data is ensured as each site comes equipped with a specific elliptic curve. A random number is generated throughout the network, while the master public key and parameters are constants. Generated with a fresh key (k) after certain time intervals, guaranteeing the validity of the signatures. Each site selects an additive and secure approach to prevent attacks at the beginning of every round. The private key is selected and used to compute the corresponding public key.

The act of selecting a private key involves,

This task is simple and involves selecting an integer in the elliptic curve's field by the website. Each round of processing requires a new set of public and private keys. Another site can be determined by a malicious site with just two signatures. The private key is responsible for signing a message, but if another message is signed using the same private key, it will be evident. The signature alone is not secure, but we enhance the security by signing the message with an additional level of protection. Afterward, the sentence is coded before transmission to the succeeding level. In case a website approves of it repeatedly. The private key can be determined by another site if messages are encrypted using the same key. The signature scheme is created in a way that allows an easy combination of all signatures. Arithmetic operation increases the workload of a master site significantly. The compact size makes it suitable for PPDM purposes. Data that has been exchanged is enhanced to function with homomorphic encryption and be aggregated.

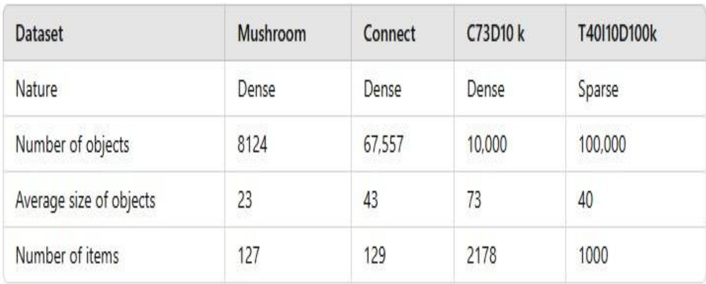
Time is up, it collects and processes the signatures. Once the aggregator acquires the data; they will combine it with both ciphertexts - namely, the digital signature and public keys. As a result of this process, only one set of exchanged data is sent to the master.

The collection includes a single ciphertext that represents the total readings from all locations. Moreover, it gets a signature that corresponds to both the total of data and the sum. All sites' public keys are collected, so that the master can decrypt and authenticate the message. By adding up the signatures and public keys, we can obtain a total value.

## Test Environment and Datasets

All simulation processes on the AWS platform are implemented using Java. To showcase the effectiveness of our proposed approach, we will demonstrate it using the EC2 cloud computing platform. Our approach involves the utilization of High-CPU Medium Instances containing 1.7 GB memory and featuring 5 EC2 compute resources. Two virtual cores are specified, with each core having 2.5 EC2 compute units and a local instance storage capacity of 320 GB. We opt for a variety of datasets, both dense and sparse, that can be accommodated by 64-bit platforms' storage capabilities. One of the sparse datasets discovered on the UCI KDD machine learning repository is Mushroom [28]. We connected 29 to Connect, C73D10K to 30, and T40I10D100K to 35 in our experimentation. Please refer to Table 3 for further information. The attributes of the dataset are being delineated.

Table 3: 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 |

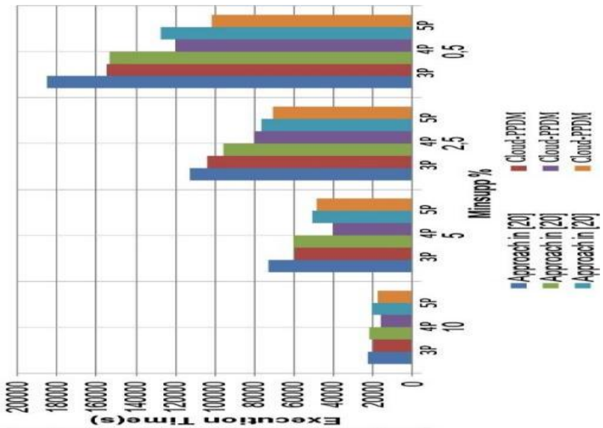
## Result and analysis

To evaluate the efficiency of our methodology, we carried out a comparative study on the processing time between Cloud-PPDM and the method described in [12], maintaining consistency across a range of datasets with differing characteristics. Our analysis began with the methodology outlined in [12] to measure the time taken for processing both dense and sparse datasets while adjusting the number of sites from three to five

Fig.1 Execution time of Cloud-PPDM vs approach proposed in [12]

In Figures 1 and 2, the vertical axis displays the execution time of Cloud-PPDM compared to the method proposed in [12], shown for both the Mushroom and Connect datasets. The horizontal axis illustrates changes in execution time based on the number of sites (P) for various minsup values, where P represents the number of sites. Examining Figure 1, for example, in the Mushroom dataset with a minimum of 60% and three sites, Cloud-PPDM required 2,218 s compared to 3,494 s with the alternative approach. Similarly, in the Connect dataset with a minsup of 90% and four sites, Cloud-PPDM completed in 324,216 s while the other method took 453,415 s. Importantly, the total processing time rises linearly as mins up decreases, primarily due to increased computation time for generating frequent closed itemsets, which outweighs the time spent on communication and distribution management

Execution time for C73D10K dataset Execution time for T40I10D100K dataset



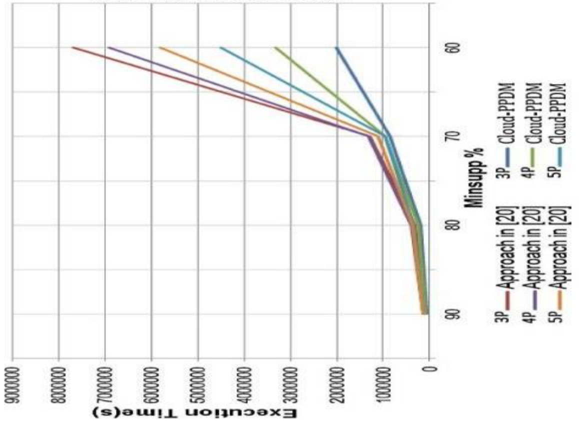


Fig 2: Execution time of cloud PPDM vs approach proposed in [12] respectively on C73D10K and T40I10D100K datasets.

Figure 2 presents the comparison of execution times between Cloud-PPDM and the method proposed in [12] using the C73D10K and T40I10D100K datasets. It's clear that our approach consistently demonstrates shorter execution times across both datasets. For example, with three sites for the C73D10K dataset and a minimum of 80%, Cloud-PPDM required 197.614 s compared to 327,143 s with the alternative method. Similarly, with five sites for the T40I10D100K dataset and a minsup of 0.5%, Cloud-PPDM took 100.068 s versus 127.583 s with the other approach. Moreover, in the case of the T40I10D100K dataset, we observe execution time improvement as the number of sites increases. The communication cost of Cloud-PPDM is contingent upon the number of sites, following a complexity of O(n), where n represents the number of sites. In summary, our experimental analysis (Figures 1 and 2) underscores the superior efficiency of Cloud-PPDM in extracting frequent closed itemsets within a distributed setup while upholding data privacy compared to the method proposed in [12].

## Conclusion

In this paper, we present a novel secure method that complements the Dist-CLOSE algorithm by leveraging homomorphic encryption. This innovative approach facilitates secure and anonymous mining tasks while preserving the confidentiality of data sources during the extraction of frequent closed patterns in distributed environments such as cloud computing. Our method ensures individual security through efficient communication schemes. Extensive experimentation conducted on standard datasets validates the effectiveness and security of our proposed scheme, demonstrating enhancements in both runtime performance and security analysis. Our future research endeavors aim to enhance this approach by enhancing the autonomy of exchanged data between sites. We intend to empower the data itself with protective measures during the exchange, eliminating the necessity for verification calculations by the master site to ensure safety. This advancement holds promise for further enhancing the efficiency and security of distributed mining operations.

## References

1 Ben Yahia, S., Mephu Nguifo.(2004), E.: Approches d'extraction de règles d'association basées sur la correspondance de Galois. Ingénierie des systèmes d'information 9(3–4), 23–55

2 Kumarn, D.S, Suneetha, C.H. (2012), Chandrasekhar, A.: Encryption of data using elliptic curve. Int. J. Distrib. Parallel Syst. (IJDPS) 3(1)

3 Gajbhiye, S., Karmakar, S. (2015)., Sharma, M.: Diffie Hellman key agreement with elliptic curve discrete logarithm problem. Int. J. Comput. Appl. 129(12) (0975 8887) 4 Moumita, R., Nabamita, D., Jyoti, K.A. (2014).: Point generation and base point selection in ECC: an overview. Int. J. Adv. Res. Comput. Commun. Eng. 3(5)

5 Boneh, D., Gentry, C., Lynn, B., Shacham, H. (2003).: Aggregate and verifiably encrypted signatures from bilinear maps. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 416–432. Springer, Heidelberg doi:10.1007/3-540-39200-9_26 6 Vassilios, S.V., Elisa, B., Igor, N.F., Loredana, P.P., Yucel, S., Yannis, T. (2004): State of the art in privacy preserving data mining. SIGMOD Rec. 33, 50–57

7 Hussien, A., Hamza, N., Hefny, H. (2013): Attacks on anonymization-based privacy-preserving: a survey for data mining and data publishing. J. Inf. Secure. 4(2), 101–112

8 Li, Y., Chen, M., Li, Q., Zhang, W. (2012): Enabling multilevel trust in privacy preserving data mining. IEEE Trans. Knowl. Data Eng. 24(9), 1598–16129 Li, X., Yan, Z., Zhang, P. (2014): A review on privacy-preserving data mining. In: IEEE International Conference on Computer and Information Technology (CIT), pp. 769–774

10 Kantarcioglu, M., Clifton, C. (2002): 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 11 Vaidya, J., Clifton, C. (2002): 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 12 Moez, W., Poncelet, P., Ben Yahia, S. (2009): 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 13 Patel, S.J., Punjani, D., Jinwala, D.C. (2015): An efficient approach for privacy-preserving distributed clustering in the semi-honest model using elliptic curve cryptography. Int. J. Netw. Secure. 17(3), 328–339

14 Jitarwal, Y., Mangal, P.K., Suman, S.K. (2015): Enhancement of elgamal digital signature based on RSA & symmetric key. Int. J. Adv. Res. Comput. Sci. Softw. Eng. 5(5)

15 Okamoto, T., Uchiyama, S. (1998): 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

16 Muller, S.D., Holm, S.R., Sondergaard, J. (2015).: Benefits of cloud computing: literature review in a maturity model perspective. Commun. Assoc. Inf. Syst. 37 Article no. 42

17 Hayward, R., Chiang, C.C. (2015).: Parallelizing fully homomorphic encryption for a cloud environment. J. Appl. Res. Technol. 13(2), 245–252 ISSN 1665-6423

18 Bastide, Y., Taouil, R., Pasquier, N., Stumme, G., Lakhal, L. (2000): Mining frequent patterns with counting inference In KDD Conference, pp. 66–75

19 Zitouni, M., Akbarinia, R., Ben Yahia, S., Masseglia, F. (2015): 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

20 Wang, P. (2010): Survey on privacy preserving data mining. Int. J. Digit. Content Technol. Appl. 4 (9) Using Homomorphic Encryption to Compute Privacy-Preserving Data Mining 411

21 Thakur, D., Gupta, H. (2013).: An exemplary study of privacy-preserving association rule mining techniques. Int. J. Adv. Res. Comput. Sci. Softw. Eng. 3(11) P.C.S.T., BHOPAL C.S Dept., India

22 Nithya, C.V., Jeyasree, A. (2013).: Privacy-preserving using direct and indirect discrimination rule method. Int. J. Adv. Res. Comput. Sci. Softw. Eng. 3(12) Vivekanandha College of Technology for Women Namakkal India

23 Lipmaa, H. (2007): Cryptographic techniques in privacy preserving data mining, University College London, Estonian Tutorial 24 Rathore, B.S., Singh, A., Singh, D. (2015).: A survey of cryptographic and non-cryptographic techniques for privacy preservation. Int. J. Comput. Appl. 130(13) (09758887)25 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) 412 H. Hammami et al.

26 Zhang, N., Li, M., Lou, W. (2011): Distributed data mining with differential privacy. In: Proceedings of the IEEE International Conference on Communications (ICC), pp. 1–5

27 Giannotti, F., Lakshmanan, L., Monreale, A., Pedreschi, D., Wang, H. (2013): Privacy-preserving mining of association rules from outsourced transaction databases. IEEE Syst. J. 7(3), 385– 395

28 Canard, S., Desmoulins, N., Devigne, J., Le Hello, D. (2012): Anonymisation des donnèes. Document de travail de l'objet de recherche: trust identity and privacy

29 Chang, X.-Y., Deng, D.-L., Yuan, X.-X., Hou, P.-Y., Huang, Y.-Y., Duan, L.-M. (2015): Experimental realization of secure multi-party computation in an entanglement access network