

**Innovative Technique to Encrypt Data for Data
Mining Purposes in Cloud Computing**

تقنية مبتكرة لتشفير البيانات لأغراض تنقيب البيانات في الحوسبة السحابية

Prepared by:

Amer H. Khalifa

Supervised by:

Prof. Hebah H. O. Nasereddin

**Master Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of Master in Cloud Computing
Security and Services,**

**Department of Computer Science,
Faculty of Information Technology
Middle East University**

Aug. 2021

Authorization

I, **Amer H. Khalifa**, authorize The Middle East University Graduate Studies to supply a hard copy of my Thesis to libraries, establishments, or individuals upon their request.

Name: Amer H. Khalifa.

Date: 18 / 08 / 2021.

Signature

A handwritten signature in blue ink, consisting of a stylized 'A' followed by a long, sweeping horizontal line that curves upwards at the end.

Thesis Committee Decision

This is to certify that the thesis entitled " **Innovative Technique to Encrypt Data for Data Mining Purposes in Cloud Computing**" was successfully defended and approved on 18/08/2021.

Examination Committee Members:

Prof. Hebah H. O. Nasereddin (Supervisor)

Middle East University

Dr. Bassam Al-Shargabi (Internal Examiner/Chairman)

Middle East University

Dr. Sharefa Murad (Internal Examiner)

Middle East University

Prof. Khalid Abdul-Hafez AlKaabneh (External Examiner)

Al Ahliyya Amman University

Acknowledgment

I send my warm greetings to all the members at the Faculty of Information Technology at the Middle East University, especially the research supervisor, and to everyone who taught me on this long path of knowledge.

The researcher

Amer Hikmat

Dedication

To my mother and father...with salute.

To my wife, my two little moons Nouran & Nadine.

To all my sisters, my friends and to everyone drove me to this achievement.

I dedicate it to you all.

The researcher

Amer Hikmat

Table of Contents

Innovative Technique to Encrypt Data for Data Mining Purposes in Cloud Computing	i
Authorization	ii
Thesis Committee Decision	iii
Acknowledgment.....	iv
Dedication	v
Table of Contents	vi
List of Tables	ix
List of Figures.....	x
List of Abbreviations	xii
English Abstract.....	xiii
Arabic Abstract.....	xv
1. CHAPTER ONE: Introduction	1
1.1 Research Topic	1
1.2 Problem Statement.....	3
1.3 Significance	4
1.4 Objectives.....	5
1.5 Research questions	6
1.6 Delimitations.....	7
2. CHAPTER TWO: Theoretical Background and Literature Review	8
2.1 Big data	8
2.2 Data mining.....	8
2.2.1 Data mining techniques	9
2.3 Security of data in the cloud.....	11
2.4 The Process of KDD	11
2.4.1 Steps of KDD process.....	12
2.4.2 The privacy concerns of PPDM	12
2.4.3 User role-based methodology to be revised all numbers	13
2.5 Cloud Computing.....	14
2.6 Cloud Security Concerns.....	14
2.6.1 Honest but Curious CSP	14

2.7	Types of Cyber Attacks	14
2.8	Cryptographic Techniques	16
2.9	Related work	18
3.	CHAPTER THREE: Methodology & Proposed Model	21
3.1	Data Indexing	21
3.2.	Methodology	22
3.2.1	Encryption Scenario	22
3.2.2	Proposed Technique	25
3.2.2.1	Cloud Environment	25
3.2.2.2	Amplifying/Incrementing Process	27
3.2.2.2.1	Amplifying to 16 characters in each row	29
3.2.2.2.2	Amplifying to 16 and 32 characters in each row	29
3.2.2.2.3	Amplifying to 32 characters in each row	30
3.3	Encryption process	31
3.3.1	Shuffling	31
3.3.2	Bits shifting	33
3.3.3	Bits Substitutions	33
3.3.4	Key Generation	33
3.3.4.1	Key 1 Generation	34
3.3.4.2	Key 2 Generation	34
3.3.5	Encryption Steps (Pseudo Code)	35
3.3.5.1	Flowchart (Encryption)	37
3.3.6	Decryption Process (Pseudo Code)	38
3.3.6.1	Flowcharts (Decryption)	39
3.3.7	Applying Association Rule on Cipher	40
3.3.8	Applying Association Rule on Plain Text	42
3.3.9	Avalanche Effect	43
4.	CHAPTER FOUR: Implementation and Experimental Results	45
4.1	Local and Cloud Systems Specifications	45
4.2	Data Used	46
4.3	Encryption Module	46
4.4	Decryption Module	55
4.5	Association Rule	64

4.6 Avalanche Effect	69
5. CHAPTER FIVE: Conclusion and Future Work	73
5.1 Achievements	73
5.2 Drawbacks	74
5.3 In General	75
5.4 Future Work	76

List of Tables

Number	Content	Page
1	Table 3-1	32
2	Table 3-2	32
3	Table 3-3	32
4	Table 4-1 Encryption vs Decryption	63

List of Figures

Figure 2-1 Privacy Preserving Data Mining Techniques _____	11
Figure 2-2 The process of KDD _____	12
Figure 2-3 A simple illustration of the application scenario with data mining (Xu et al., 2014) _____	13
Figure 2-4 Proposed security algorithm (Chaudhary & Gulati, 2016) _____	16
Figure 3-1 Adopted Scenario _____	23
Figure 3-2 Client/Costumer Scenario for a Costumer _____	24
Figure 3-3 Other side of Client/Costumer Scenario _____	25
Figure 3-4 Proposed User Login Process (Local System) _____	26
Figure 3-5 Proposed User Login Process (Cloud System) _____	27
Figure 3-6 A comma-separated values/CSV File _____	29
Figure 3-7 Flow Chart of Encryption _____	37
Figure 3-8 Flow Chart of Decryption _____	39
Figure 3-9 Concept of C.S.V file in local and cloud systems _____	40
Figure 3-10 Random Transactions Conducted on Cipher _____	41
Figure 3-11 Encrypted Items Repeated in Transactions (Support) _____	41
Figure 3-12 Random Transactions Conducted on Plain _____	42
Figure 3-13 Items Repeated in Transactions (Support) _____	43
Figure 4-1 Local Environment Specifications _____	45
Figure 4-2 Time Needed to Encrypt 500 Items _____	47
Figure 4-3 Time Needed to Encrypt 1000 Items _____	48
Figure 4-4 Time Needed to Encrypt 5000 Items _____	49
Figure 4-5 Time Needed to Encrypt 10000 Items _____	50
Figure 4-6 Time Needed to Encrypt 50000 Items _____	51
Figure 4-7 Time Needed to Encrypt 100,000 Items _____	52
Figure 4-8 Time Needed to Encrypt 500,000 Items _____	53
Figure 4-9 Time Needed to Encrypt 1,000,000 Items _____	54
Figure 4-10 Chart Shows Time Needed to Encrypt Different Count of Items _____	54
Figure 4-11 Time Needed to Decrypt 500 Items _____	55
Figure 4-12 Time Needed to Decrypt 1000 Items _____	56
Figure 4-13 Time Needed to Decrypt 5000 Items _____	57
Figure 4-14 Time Needed to Decrypt 10000 Items _____	58
Figure 4-15 Time Needed to Decrypt 50000 Items _____	59
Figure 4-16 Time Needed to Decrypt 100000 Items _____	60
Figure 4-17 Time Needed to Decrypt 500000 Items _____	61
Figure 4-18 Time Needed to Decrypt 1000000 Items _____	62
Figure 4-19 Decryption Chart _____	62
Figure 4-20 Plain vs Cipher in Size and Time _____	63
Figure 4-21 Applying Affinity Rule in Cloud #1 _____	65
Figure 4-22 Applying Affinity Rule in Cloud #2 _____	66
Figure 4-23 Applying Affinity Rule Locally #1 _____	67
Figure 4-24 Applying Affinity Rule Locally #2 _____	68
Figure 4-25 Avalanche Effect of Without Amplifying _____	69

Figure 4-26 Avalanche Effect with Amplifying from 16 bytes/128 bits → 32 bytes/256 bits	70
Figure 4-27 Avalanche Effect with Amplifying either 16 bytes/128 bits or 32 bytes/256 bits	71
Figure 4-28 Avalanche Effect with Amplifying to 32 bytes/256 bits	72

List of Abbreviations

Abbreviation	Meaning
PPDM	Privacy Preserving Data Mining
DM	Data Mining
WEF	World Economic Forum
CSP	Cloud Service Provider
KDD	Knowledge discovery from data
BFS	Breadth-First Search
GF	Generation Function
Eclat	Equivalence Class Transformation
FP	Frequent Pattern
SVM	Support Vector Machine Algorithms
LFSR	Linear Feedback Shift Register
CPU	Central Processing Unit
HE	Homomorphic Encryption
VC	Verifiable Computation
MPC	Secure Multi-Party Computation
AES	Advanced Encryption System
ETL	Extract, Transform and Load
CSV	Comma-separated values
PaaS	Platform as a Service
AWS	Amazon Web Services
IaaS	infrastructure as a service
IDE	integrated development environment
KB	Kilobytes
mb	megabytes

Innovative Technique to Encrypt Data for Data Mining Purposes in Cloud Computing

Prepared by:

Amer H. Khalifa

Supervised by:

Prof. Hebah H. O. Nasereddin

Abstract

Encryption is inseparable in security world. Indeed, symmetric algorithm is glowing when speaking about lightweight encryption. his thesis is evidence that the application of analytic operations in cryptography is not exclusive to other cryptographic techniques. However, this study does not present itself as being able to perform complete arithmetic operations as in other encryption methods used for this purpose, but by using it, it would be possible to perform analytical operations on encrypted texts as will come later in the whole technique used in this study.

The researcher encrypted the texts row by row using a special simple algorithm consisting of two identical keys, the encryption was done four consecutive times using keys of different lengths each time, and compared the effectiveness of each length by measuring (Avalanche Effect) for each of them, and it was found that the greater the key length is, the more he got a bigger Avalanche Effect, and it's worth noting that he didn't get a suitable (Avalanche Effect) only when using a 256-bit key.

Then he sends the encrypted texts to the cloud and there he uses the "Affinity" correlation rule algorithm on the cipherttexts. And after all of that he compares the results with the Affinity algorithm when it is applied to the

"unencrypted" plaintext on the device not in the cloud. And it turned out that all the results of the affinity algorithm were identical, in other words, each ciphertext or its index that was used in the affinity algorithm in the cloud is the same unciphered text or its index that was used in the same algorithm in the local device.

Keywords: PPDM, Symmetric Encryption, Cloud Computing, Association Rule.

تقنية مبتكرة لتشفير البيانات لأغراض تنقيب البيانات في الحوسبة السحابية

إعداد:

عامر حكمت خليفة

إشراف:

الأستاذة الدكتورة هبة ناصرالدين

الملخص

لا وجود لبديل عن التشفير في عالم الامن السيبراني. وبشكل مؤكد فان للخوارزمية المتماثلة أهمية كبيرة عند الحديث عن التشفير خفيف الحمولة. وبما إن تطبيق العمليات التحليلية على نص عادي أمر مفروغ منه، الا ان تطبيق نفس العمليات على نص مشفر وتوقع نفس النتائج عند فك التشفير لا يزال يمثل جدلاً.

هذه الأطروحة دليل على أن تطبيق العمليات التحليلية في التشفير ليس حكراً على تقنيات التشفير الأخرى. ومع ذلك، فان هذه الدراسة لا تقدم نفسها كونها تمكن من إجراء عمليات حسابية كاملة كما في طرق التشفير الأخرى المستخدمة لهذا الغرض، ولكن باستخدامها سيتم التمكن من إجراء عمليات تحليلية على النصوص المشفرة كما سيأتي لاحقاً في التقنية المستخدمة ككل في هذه الدراسة.

قام الباحث بتشفير النصوص صف تلو الآخر باستخدام خوارزمية بسيطة خاصة تتكون من مفتاحين متماثلين، تم التشفير أربع مرات متتالية باستخدام مفاتيح باطوال مختلفة في كل مرة، وقارن فعالية كل طول عن طريق قياس (Avalanche Effect) لكل منها، وتبين انه كلما زاد طول المفتاح كلما حصل على

(Avalanche Effect) أكبر، ومن الجدير بالذكر انه لم يحصل على نسبة (Avalanche Effect) مناسبة

الا عند استخدام مفتاح مكون من 256 بت.

ثم قام بإرسال النصوص المشفرة إلى السحابة واستُخدمت هناك خوارزمية قاعدة الارتباط "تقارب" على

النصوص المشفرة وبعد ذلك كله قام بمقارنة النتائج مع خوارزمية التقارب عند تطبيقها على نص عادي

"غير مشفر" على الجهاز وليس في السحابة. وتبين له ان جميع نتائج خوارزمية التقارب كانت متطابقة،

بمعنى اخر ان كل نص مشفر تم استخدامه او استخدام المؤشر الخاص به في خوارزمية التقارب في

السحابة هو نفس النص غير المشفر الذي تم استخدامه او استخدام المؤشر الخاص به في نفس الخوارزمية

في الجهاز وليس في السحابة.

الكلمات المفتاحية: حماية خصوصية تنقيب البيانات، التشفير المتناظر، الحوسبة السحابية،

قاعدة الارتباط.

CHAPTER ONE

Introduction

1.1 Research Topic

With a huge development of technology and the expansion depending on data, data is representing the main element for development process. Therefore, could see massive amount of data is needed by any organization, these data need to be stored in large amounts of storages in safe and secure place.

Furthermore, the need to extract the useful information from this data has increased, so there are two main issues to treat with these data in a proper manner.

As above mentioned in the two paragraphs, Safety and Extract useful information must be handled from the aspect of this thesis.

To store big data, the needs are directed the researchers to think about complicated infrastructure and multilevel of security. handling these two factors (complicated infrastructure and multilevel of security); could be conducted by Cloud Service. Cloud is the suitable solution to do this, data accession anywhere and anytime without any obstacles and without any worries about backup or security. As long as Cloud is Representing the ideal solution to any entity which its operations generate big data and doesn't want to lose effort and time to handle the multi-level of operations to manage and secure this data, nonetheless; cloud stand still represents a fragile object when talking about security in IT environment (C. Wang et al., 2011). At this point, this thesis sought to apply security level for mining the data in cloud.

Data mining is a process of extracting useful hidden information from large databases. It uses many tools and algorithms for the process of sorting through large quantities of data sets and finding out relevant information, (Chaudhary & Gulati, 2016).

The data mining sub-tools help to predict the futuristic behavior and trends which allow the organizations to obtain a proactive approach of analysis and making decisions for the huge growth at many different aspects. Data mining automates the system to find the relevant information from the databases of the data warehouse of an organization owning that warehouse of data.

Cloud computing could represent a trend in IT world (W. Y. C. Wang et al., 2011), by helping to make processing on the Internet. through the highly optimized virtual servers. These virtual machines offer numerous software, hardware and data resources that can be easily used. Assisting Organizations to connect directly to the cloud and use these services in pay-per-use option. This helps companies to avoid capital expenditure on additional local infrastructure resources and immediately increase or decrease this pool of infrastructure as required (Baek et al., 2015).

Cloud computing are based on the deployment of the cloud computing (Public cloud- Private cloud- Hybrid cloud- Community cloud). they provide services to large companies instead of having their own infrastructure or data centers and without the need of highly cost to managing and maintaining them.

The discussion about the benefits of the cloud may take longer but it is invertible to avoid talking about the advantage of the cloud. Since cloud is offering many innovative features such as encryption strategies to ensure the securing of its storage, access control, secure backup. However, cloud allows users to reach powerful computing capabilities that exceeds their available physical ones.

Still, Cloud is suffering from many security problems (*Farhan Bashir Shaikh and S. Haider, "Security Threats in Cloud Computing," 2011, n.d.*). The main security concerns

are Identification and Authentication: Multiple access to the cloud allow access to the software by more than one user (Manogaran et al., 2016).

1.2 Problem Statement

Assume that there is a need to discover a locked, protected, well-guarded house, searching for a specific commodity, got the authorization to get inside that house legally, and needs to search every room in that house, finding that commodity and then get out of that house, during this mining process, all the house commodities became more vulnerable than ever to any breakthrough and intruders. So, it is required; in any mining methods; to preserve all house contents, the modifications of all house contents and commodities. Therefore, in a way or another, Data mining will breach data privacy; and the data miner shall be obligated to modify the data before applying the mining operations, and these data should; concurrently; be useful after modification. Another major concern of data collector is how to maintain utilization of modified data while retrieve them to its normal formation. That drove us to navigate for two layers of protections, protection of the data itself, and the protection of mining results. The main security concerns with data extracted through DM process is the privacy threats, As DM process will violate the privacy due to unauthorized access to private data, discovery of sensitive information, and use of that private data.

Decision makers; always; are willing to maintain knowledge, and that exactly what DM offers, the data mining algorithms shall be applied to the data obtained from the data storage. As data mining operations are accompanied by many security issues, personal information can be directly observed and data breach happens, privacy of the data owner will also be compromised.

So, the needs for Data modulating are inevitable, the different strategies of encryption may help modulating this data; if and only if; modulating the data will not affect the data mining algorithms when they applied on the modulated/encrypted data, and that what this thesis is all about.

1.3 Significance

It is of axioms; every progress has been taken in the field of facilitating daily life through technology is matched by many complications in terms of the methods that led to the creation of these inventions and how to maintain the high-end quality services from these developments. In digital world, that became not only as axiom of its nature, but it is now one of the main inherent features that define digital technology. Today, according to World Economic Forum (WEF) website, the collective number of digital bytes is about 44 zettabytes. Zetta = 10^{21} or 1000 000 000 000 000 000 000, (*How Much Data Is Generated Each Day? | World Economic Forum, n.d.*).

Living under this wide umbrella of DATA, when mentioning DATA; the first idea arise in our minds is IT, since it has been deeply correlated to IT entities, while it has become the best global commodity ever, hence data overtaken oil as the most valuable commodity (*Regulating the Internet Giants - The World's Most Valuable Resource Is No Longer Oil, but Data | Leaders | The Economist, n.d.*). Therefore, data has not been confined to digital world only, rather it became the most powerful commodity of states rather than companies, hence there are many economies of states are based on DATA.

Thus, many states and major international companies are in a big race to produce the new development and invention of their products to be supplied to markets, and all of that should be conquered within not more than a year. The immersion in that big race must be armed with new and developed technologies.

Data Mining could be considered as a main weapon in that battlefield. It is the flagship of how to harvest the flowed data from heterogeneous sources to their different estuaries, structure them in smart patterns to maintain valuable knowledge about globe in different disciplines through data mining technologies.

Within that atmosphere, any breach or leakage of that valuable commodity will represent a real threat to whole entity. Accordingly, while speaking about data in cloud; losing the reliability between costumer and cloud providers shall break the basic and most important rules of Cloud (Rashid & Chaturvedi, 2019). Therefore, customers should ensure that data is collected, processed, and transferred in a well-guarded manner, as securing sensitive data is an important aspect for running business successfully (Yildirim, 2016).

Many indicators refer to an increase of data breach techniques (Ladekar, 2014), as intruders and hackers are in a daily race to induce new methods and techniques to penetrate secure data, that battlefield propelled this thesis to figure out a new method to change data protection techniques to withstand any penetration attempt.

1.4 Objectives

In this thesis, the author has strived to trace every piece of information about data mining. This collected information has been cited from well-mannered scientific research published in prominent data basis. Hereinafter, the author has went through scanning and analyzing various issues regarding data mining privacy and security, to provide an overview of the data mining. Yet, this thesis could not represent the ideal solution to comprehend all information about data mining privacy and security; nonetheless, it provides a general understanding of the data mining and how to maintain induced technique to cipher data during data mining process. Good to mention this research is not

privacy preserving data mining technique, hence privacy preserving data mining allows all kind of computation to be applied on the data which obligate the researcher to maintain fully homomorphic encryption.

The main objective of this study is to implement and apply security level for the extracted information to secure sensitive data and protect mining results, assuming that is the result of mining process shall be secure from any kind of interception; aiming to apply data mining algorithms effectively without compromising the security of sensitive information contained within that data. To make this comes to reality, practical steps have been taken by adding security level to the information maintained from DM process and make this effective information inaccessible from unauthorized users, as a result the duty could be summarized by adding security level characterized by “encryption” the data in an own novel mechanism. In another word, to keep data secure after applying DM process; must add our own additional security level (other than the Cloud Service Provider/CSP’s security level), represented by encrypt data before transmitting to cloud (*Category 8 // Encryption, 2012*). And the objective is making data apart from unauthorized access.

1.5 Research questions

1. How could encrypted data shall be retrieved after applying the data mining process (Affinity Association Rule
2. what are the random functions used in generating the keys of encryption process?
3. What are the specific suggested procedures of the induced algorithm that would achieve the encryption and masking of extracted data?
4. How to decrypt the data to be readable by authenticated users?

5. How to guarantee these solutions are effective and preserve the minimum accepted level of privacy for needed data?

1.6 Delimitations

Obviously, there are several issues related to security of data, and yet more concentrate is needed to cover these issues, this thesis has considered these issues and will indicate them in the related study section.

One of these issues is the storage size of data and where that data is located. Consequently, must consider Distributed Data and NoSql concepts (Tian, 2017), when security measurements are about to be applied, and there are many new Data Mining techniques may result in optimizing processing performance, but; in the meantime; securing these tools became more sophisticated (Qi & Zong, 2012).

Sustainable security measurements is one of challenges could be faced in any DM environment to mitigate loss and exposure data risk (Baek et al., 2015). However, within Data Mining; it is regularly required the mutation of security controls and upgrade system security to withstand the growth of attacking techniques, to keep data secure and to mitigate loss and the exposure of data (Niranjan et al., 2016). It's good to mention that security measurements must be scrutinized, monitored and under control to check the effectiveness of these measurements (Penetration Testing) (C. Wang et al., 2011).

Nowadays; a new technology is induced by attackers to breach data, resulted that any security solution should be multi-layered and mutable (Niranjan et al., 2016). And as mentioned before, this thesis does not offer a full security for every data mining algorithm and technique, the author has negotiated this issue extensively in future work section.

CHAPTER TWO: Theoretical Background and Literature Review

2.1 Big data

Big data and business analytics have grown massively over the past few years, big data is used to describe massive, complex, and real-time streaming data that require wise management, analytical, and processing techniques to get extracted. The type of data generated from the usage of new technologies such as social media, smart phones, and sensors, are often not clean data, they may contain missing facts, unclear data, messy and noisy data. Nonetheless, Data editing before applying any model is inevitable (Mikalef et al., n.d.) (Gupta & George, 2016).

The huge amount of data requires the usage of powerful computational techniques to discover trends and patterns within and between these extremely large datasets. When data extraction process is applied, the retrieved data is must be meaningful and useful data which eventually used in statistics report. This assists to discover new ideas in real time report (George et al., 2014).

2.2 Data mining

(K. & K., 2017) has considered DM as an essential and important step in knowledge discovery in databases (KDD), is used to discover useful, unknown patterns from large repositories of data. DM consists of various functionalities, techniques and algorithms that are used to extract interesting patterns from the large repository of data (Ashraf, 2014). Data mining is a process of extracting interesting patterns, associations, changes, anomalies and significant structures from large amounts of data which is stored in multiple data sources such as file systems, databases, data warehouses or other information repositories (*Data Mining: Concepts and Techniques - 3rd Edition*, n.d.).

2.2.1 Data mining techniques

Data mining techniques are classified into two categories: descriptive and predictive. (Monshizadeh & Yan, 2014). The descriptive category provides information from data itself (Classification), The predictive category extracts information that is discovered based on previous data (Clustering). However, DM algorithms could be much wider, below set of algorithms (Ashraf, 2014):

- **Classification:** It is used to retrieve important and relevant information about data, and metadata. This method helps to classify data in different classes.
- **Clustering:** It is a data mining technique to identify data that are like each other. Clustering helps to understand the differences and similarities between the data.
- **Regression:** It is the data mining method of identifying and analyzing the relationship between variables. It is used to identify the prospect of a specific variable, given the presence of other variables.
- **Association Rules:** this rule represents one of the main subjects of my thesis. Briefly, it helps to find the relation between two or more Items. It discovers a hidden pattern in the data set. Applying it on plain text is common but applying it on cipher is what this thesis; besides the innovative symmetric cryptographic technique; is about. It has different kinds of algorithms, as shown below:

- **Apriori algorithm**

Adopts a **breadth-first search (BFS)** strategy to count the support of item sets through nominee generation function (GF) (Agrawal & S&ant, n.d.).

- **Equivalence Class Transformation (Eclat) algorithm**

Eclat (Mohammed J. Zaki, 2000) is an algorithm of a depth-first search depends on set intersection. It could be used for sequential and

parallel execution with locality-enhancing properties (M J Zaki et al., 1997; Mohammed J Zaki et al., 1997)

- **Frequent Pattern -Growth Algorithm**

It counts the repetitions of items (attribute-value pairs) in the dataset of transactions and stores these counts in a header table and constructs the FP-tree by placing transactions into a trie (Han et al., n.d.).

- **Affinity analysis (Market Basket Analysis)**

Discovers meaningful relationships of distinct elements in a data set based on their co-occurrence. It can be used to derive considerable knowledge regarding unanticipated trends in practically all processes and systems. It takes use of analyzing attributes that are related, which aids in the discovery of hidden patterns in large data sets (Larose & Larose, 2014).

In this thesis, the author used one of this algorithm techniques on encrypted data to measure the relation of commodities based on costumers' purchase invoices. Regardless of the 'threshold' needed in some similar algorithms.

- There are other association rule mining algorithms like Algorithm for Unordered Search (Jilke, n.d.).

- **Outer detection:** It refers to observation of data items in the dataset which do not match an expected pattern or expected behavior.
- **Sequential Patterns:** It helps to discover or identify similar patterns or trends in transaction data for certain periods.

- **Prediction:** Prediction has used a combination of the other data mining techniques like trends, sequential patterns, clustering, classification.

2.3 Security of data in the cloud

Privacy Preserving Data Mining (PPDM) helps to safeguard sensitive information from an unsolicited or unsanctioned disclosure. Several PPDM approaches have been proposed by (*10View of Security in Data Mining- A Comprehensive Survey*, n.d.) Some of them are listed as shown in Fig.2-1

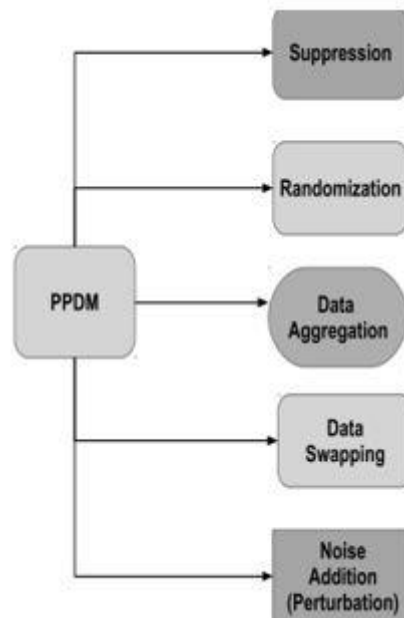


Figure 2-1 Privacy Preserving Data Mining Techniques

(*10View of Security in Data Mining- A Comprehensive Survey*, n.d.)

2.4 The Process of KDD

Based on (Xu et al., 2014), knowledge discovery from data (KDD) is the final destination be reached through the data mining process to obtain useful knowledge from data.

2.4.1 Steps of KDD process

1. Data preprocessing
 - A. Data retrieving
 - B. Data cleaning
 - C. Data integration
 - D. Data transformation: transform data to useful form
2. Data mining: extract patterns (clusters, classifications, association rules...etc)
3. Pattern evaluation and presentation

Figure 2-2 shows the process of KDD.

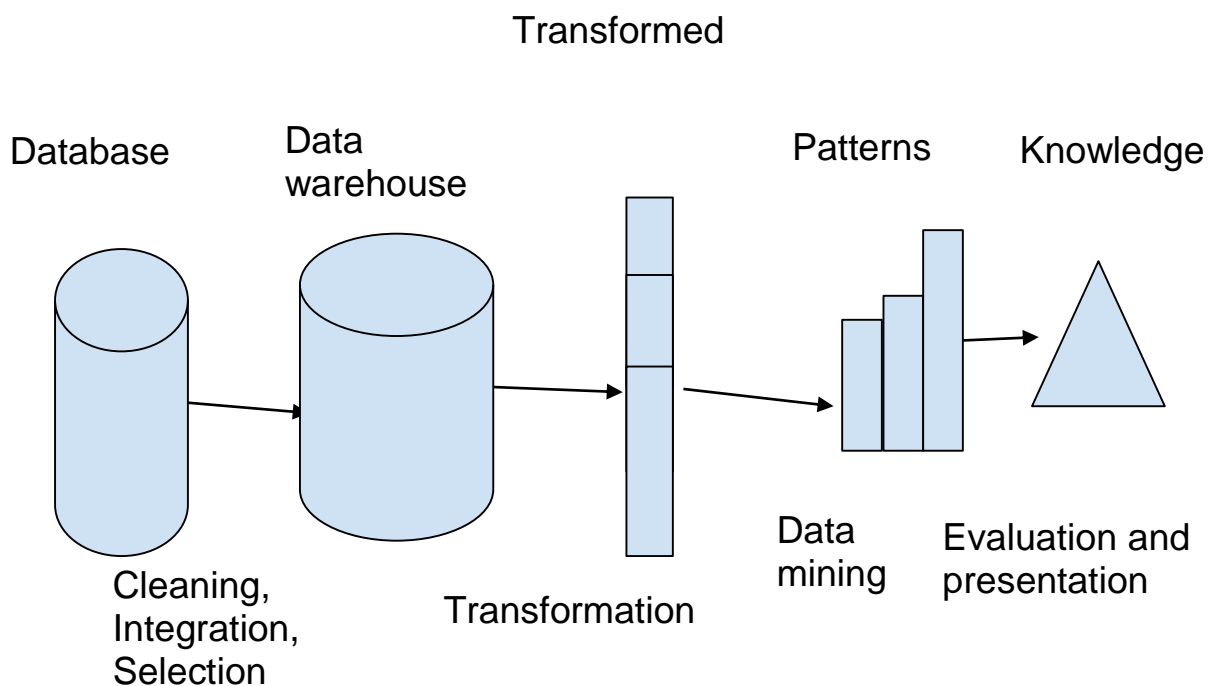


Figure 2-2 The process of KDD

2.4.2 The privacy concerns of PPDM

Discovery process of the data which can be resulted from the data mining process will cause privacy threats from people who are interested to violate access to this information. Data mining and through discovery process of information could cause the

privacy threats from people who awaits to violate access to this information. PPDM Could be applied to handle this issue (Niranjan et al., 2016).

Recently PPDM has gained a huge development. The objective of PPDM is to safeguard sensitive information from unsanctioned disclosure, while preserving utility of the data (*10View of Security in Data Mining- A Comprehensive Survey*, n.d.).

PPDM considers the sensitive raw data, which should not be directly used for mining. And, excludes sensitive mining results which disclosure privacy breach (Niranjan et al., 2016).

2.4.3 User role-based methodology to be revised all numbers

PPDM's proposed models and algorithms are mainly; focusing on how to hide that sensitive information from certain mining operations.

A user-role based methodology might conduct the review of related studies. Based on KDD process on Fig. 3, they proposed a typical data mining scenario (Fig.2-3) which consists of (Xu et al., 2014):

- Data Provider
- Data Collector
- Data Miner
- Decision Maker

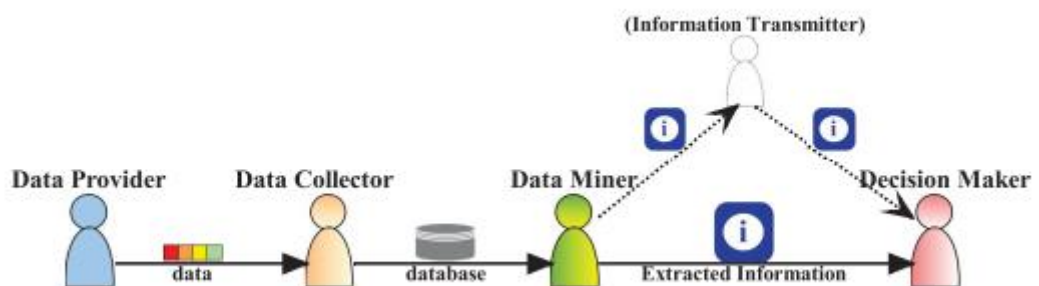


Figure 2-3 A simple illustration of the application scenario with data mining (Xu et al., 2014)

2.5 Cloud Computing

(Karimunnisa & Kompalli, 2019) defined cloud computing as a pool of shared resources, which are used by many public and private sectors, small, medium, and large enterprises, with different services based on the resources required.

Cloud services are offered by CSP (Cloud Service Provider), CSP are the companies that offer various network services, infrastructure, and business applications in the cloud with its big data centers. Amazon, Microsoft, Google, IBM, and Facebook are flagships of CSPs (Youssef, 2016).

2.6 Cloud Security Concerns

Other than what said in chapter one of this thesis about the cloud fragility, still one of the important concerns is to face honest but curious CSP who may disclose the uploaded data to its cloud environment (Samanthula et al., 2019).

2.6.1 Honest but Curious CSP

When a CSP performs operations honestly as requested by the costumer, but at the same time, they will infer and analyze encrypted information based on the uploaded data and search trapdoor (Jilke, n.d.).

2.7 Types of Cyber Attacks

As it is known, cyber-attacks are types of actions that threatening network security and stability. They can be varied from unauthorized access to some information in networks, network resources occupying, software corruption, Botnet, DoS, Malware, Spyware, Adware, Scareware and Ransomware (Monshizadeh & Yan, 2014).

A novel algorithm to protect the data from being threatened could be applied while mining the data, the main objective of this algorithm is to secure data with accuracy-obtaining of DM results. To achieve this objective, three algorithms are merged. Support Vector Machine Algorithms is used to classify the data then Shamir Secret Sharing Algorithm is applied to data. Linear Feedback Shift Register Algorithm is used to produce a sequence of bits, then LFSR generate pseudo random numbers and shift their position after every cycle, so it becomes very difficult to recognize the confidential data and misuse it. Combining all three algorithms result in accuracy of data with security (Chaudhary & Gulati, 2016).

All preprocessing work is done with data to find duplicate items in tables, duplicated data is removed from the dataset to get rid of redundancy (Zhu & Wu, 2004), Support Vector Machine Algorithms (SVM) is used to classify the data. as this algorithm is the only classification algorithm which considers the boundary values and is very efficient in high dimensional spaces. SVM can solve multilabel classification (Monshizadeh & Yan, 2014).

Fig.2-4 illustrates the algorithm proposed by (Chaudhary & Gulati, 2016), the result of their proposal is gaining accuracy when providing security in data mining with optimal CPU time.

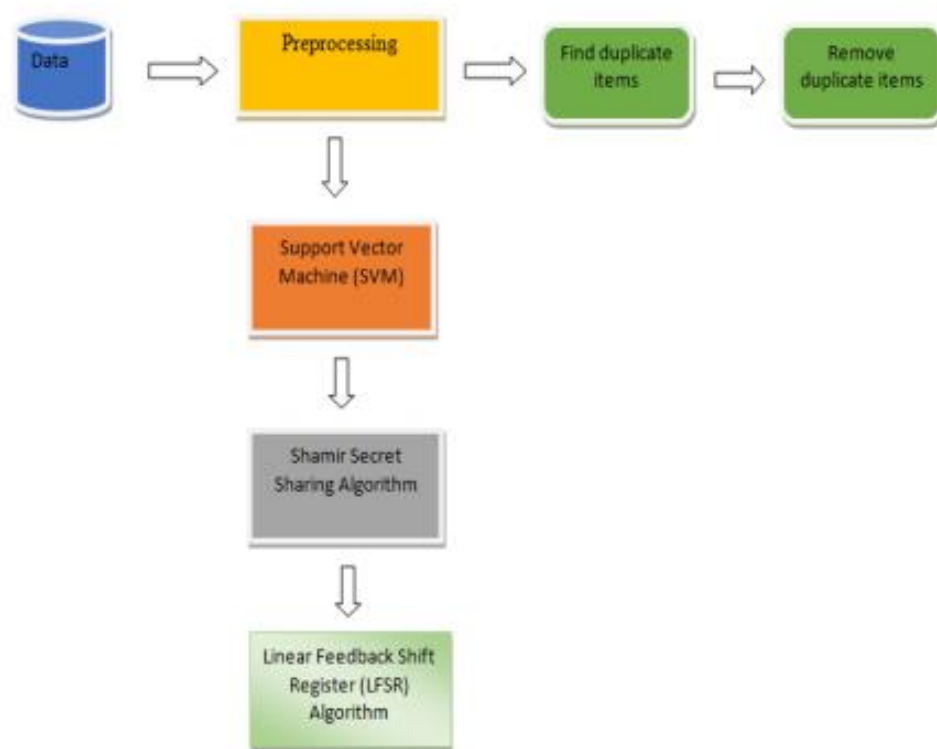


Figure 2-4 Proposed security algorithm (Chaudhary & Gulati, 2016)

2.8 Cryptographic Techniques

Other than cryptographic technique used in this thesis, three main cryptographic techniques could be surveyed that are particularly applicable to achieving secure big-data analytics in the cloud (Chaudhary & Gulati, 2016):

- Homomorphic Encryption (HE),
- Verifiable Computation (VC),
- Secure Multi-Party Computation (SMPC)

They noted that many other cryptographic techniques can be used to maintain secure cloud computing. these techniques include functional encryption, identity-based encryption, and attribute-based encryption. However, (*Encryption Techniques for Big*

Data in a Cloud, n.d.) focus on the techniques they believe are the most promising to securely delegating computational processes to a cloud.

- **Homomorphic encryption** is a type of encryption that allows functions to be computed on encrypted data without decrypting it first. It encrypts a message only. One can obtain an encryption of a function of that message by computing directly on the encryption. let $E_k(m)$ be an encryption of a message m under a key k . An encryption scheme is homomorphic with respect to a function f if there is a corresponding function f_0 such that the $D_k(f_0(E_k(m))) = f(m)$, where D_k is the decryption algorithm under key k (Gentry, 2009).
- **Verifiable Computation:** a computer offloads the computation of some function, to other an expected untrusted client, and still maintain reliable results. In this scheme, the data owner gives data, with a specification of the computation desired, to some entity called the prover. The prover outsources the result of the specified computation, along with some proof that this output is in fact correct (Lai et al., 2014)
- **Multi-Party Computation:** is suited to take advantage of the semi-trusted cloud setting. It leverages the presence of honest parties, without necessarily knowing which parties are honest, to achieve confidentiality and integrity of the data and computation. Multi-party computation offers weaker security guarantees but can be much more efficient.

2.9 Related work

Despite applying association rule on encrypted data using symmetric key algorithm is rare, also dealing with three variables in one subject is not as easy as one thinks (encryption, data mining, cloud computing). still, many well-coined papers have discussed the data mining security techniques and methods in cloud computing. So, it is possible to say there are three main types of related work for this study they are (almost close, close, somehow close). Starting with the almost close studies (Hussein Saeed & Hussain, 2019) used encryption of association rule by using modified dynamic mapping and (AES) algorithm, they divided their encryption algorithm into three phases to secure association rule mining with proper time speed. In their study, they encrypt the association rule itself rather than encrypting the stored data and that will enable to have multi party computations but data still fragile to be exposed by intruder or curious employee. (Vashi et al., 2019) have used symmetric key encryption to preserve the privacy of DM in vertically partitioned data. They dealt with data as preprocessed data so they could categorize the data to and partition them vertically based on (highly sensitive data, normal sensitive, low sensitive). After partitioning the data, they used different symmetric key algorithm for each table rather than using one symmetric key algorithm for all tables and then download them all as one partition to allow third party to have his data mining rules. That was really invented idea with good security level but need much time to be implemented. (Dawood et al., 2019) they used large symmetric key algorithm to encrypt big data. That was worthy algorithm to secure big data with proper time and speed, but they did not tell us how someone could have his own mining operations on the resulted cipher.

Other studies could be listed here to show the different encryption techniques with different methods to have one realize the other used encryption techniques rather than

symmetric algorithms. (Hossain et al., 2019) and (Sugumar & Imam, 2015) used symmetric algorithm to encrypt data in cloud, they did not talk about applying mining methods on data there.

A close related works begins with (Chaudhary & Gulati, 2016) where a novel algorithm has been made through the merging of three algorithms, All preprocessing work is done with data to find duplicate items in tuples. Duplicate data is removed from the dataset to remove redundancy. Support vector machine algorithms is used to classify the data as this is the only classification algorithm which considers the boundary values and is very efficient in high dimensional spaces. SVM can solve multilabel classification.

(Tian, 2017) have described a security intelligent model to achieve best security for big data, these data are divided into two categories, passive data and active data, the data then can be ingested by various tools. For example, using ETL to extract, transform and load data, or use flume to stream log collection, or use Sqoop to transfer data between relational databases and Hadoop, and so on. The platform provided by (Tian, 2017) is event correction, offence prioritization and real time analytics to gain insights of security intrusion. Intelligence analytics gives meaningful security information to protect big data.

Other works like (*Encryption Techniques for Big Data in a Cloud*, n.d.) depicts several symmetric, public key and homomorphic cryptosystems to help practitioners understand encryption schemes for data on cloud storage. Advanced Encryption Standard AES is used in several secure applications for cloud-based data. Fully homomorphic encryption schemes are the future for cloud environments but are far from being practical because of their performance. Homomorphic evaluation of AES has interesting applications as a practical encryption scheme for data on cloud storage. It will be a future work needs to be well-studied and implemented.

In paper of (Yakoubov et al., 2014), They presented a model for big data analytics in the cloud and surveyed several cryptographic techniques that can be used to secure these analytics in a variety of settings. While these techniques are considered a good start for secure cloud computing, further research is needed to turn them into practical solutions that can achieve secure cloud computing in the real world. This needs to design and develop secure multi-party computation techniques tailored specifically for a private semi-trusted cloud setting.

(Samanthula et al., 2019) addressed the problem of outsourcing association rule mining task to a federated cloud environment/distributed in a privacy-preserving manner. Classified the CSP as a semi-honest (or honest-but-curious) agent. they used a Homomorphic encryption based on a public-key encryption using -El- Gamal scheme based on the Diffie–Hellman key exchange and the Pailler. (LASKARI et al., 2003) Is (A-2020 Republished Paper) addressed the scenario of encrypting data locally and then transfer it to the cloud. (Salam et al., 2015) studied privacy preserving keyword search over encrypted cloud data, presented implementation of a privacy preserving data storage and retrieval system in cloud computing and used the symmetric key primitives. The implemented scheme enables a user to store data securely in the cloud by encrypting it before outsourcing and provides user capability to search over the encrypted data without revealing any information about the data or the query.

Based on the closest studies above, some of them encrypts data based on its impotency and applied mining operations on data, others encrypt the mining algorithm without encrypting data. The author encrypts whole data before sending them to cloud and sends mining algorithm to cloud to navigate the whole encrypted data. Also, to speed up the adopted security technique, the author used reference numbers/indexes instead of the encrypted texts in mining algorithms.

CHAPTER THREE: Methodology & Proposed Model

3.1 Data Indexing

It is a process of structuring data to improve the speed of retrieving the data in database and database tables. It is widely used to locate data quickly rather than search every row in database tables. It is a sorted table which generally contains: (K) the key in which the records are sorted, and (L) the physical location of the records belongs to that key in the main table. To visualize the concept of Index, a textbook is a good sample to have, since there are two ways to look for word or phrase in that textbook, either the reader keeps reading the textbook and move from a page to another to reach the word, or he could refer to the index of that textbook to locate the word or the phrase then move to that page directly. Index could improve the database processing system ((*PDF*) *A Study on Indexes and Index Structures*, n.d.).

Searchable encryption could be maintained through two main options, option 1 by creating an index that lists the documents that contain for each word of interest. option 2 by performing a sequential scan aside from an index as an option. When the number of documents is considerable, utilizing an index may be faster than scanning sequentially. The downside of utilizing an index is that it is more difficult to store and update data (Song et al., n.d.).

“Sequential scan may not be efficient enough when the data size is large. For some applications, i.e., large databases, a common technique to speed up the searching is to use a pre-computed index. Here we show how we can answer search queries with the aid of an encrypted index without sacrificing security” (Song et al., n.d.).

So, once need to query about the item, it is not necessary to write the item itself. But instead, could write the index number of the item in both plain and cipher CSV files to query about it and, to apply Affinity Rule hired in this thesis. That limited the author to encrypt each row independently. But even row by row encryption is not the only possible way to have this thesis being conducted in a proper manner. But used this method for the easiness.

3.2. Methodology

This research helps to achieve security of data and applies DM association rule to the encrypted data, and to ensure security of data extracted from mining processes, the methodology depends on symmetric encryption technique the information after getting that information from the security models.

3.2.1 Encryption Scenario

This scenario consists of “Alice” acting as a business or scientific organization and “Bob” shall represent a CSP, who offering data infrastructure. Alice owns a database, and as any database it contains fields and field values that identical to attributes and attribute values (Barsalou, 1992) referred by the data mining rules. The Attribute values, regardless of what they represent, must be encrypted. Since each attribute value has a label like “Clothes” or “Good costumer” or “potato”, this label can be transformed to an integer. As a result, a label/Integer can be transformed to a string of bits using ASCII code table.

We assume that Alice (Costumer) encrypts the database and then send them to cloud. Then, applying association rule to the encrypted data. Eventually, the rule returned to Alice. When Alice decrypting them by the same encryption-decryption key used before sending the data to the cloud, she obtains the true meaning of the extracted rules.

Clearly, the reliability is based on the resulting rules, which should match to the rules result that would be obtained when data mining had been applied on the real data/plain.

Figure.3-1 below demonstrate how the scenario could be illustrated.

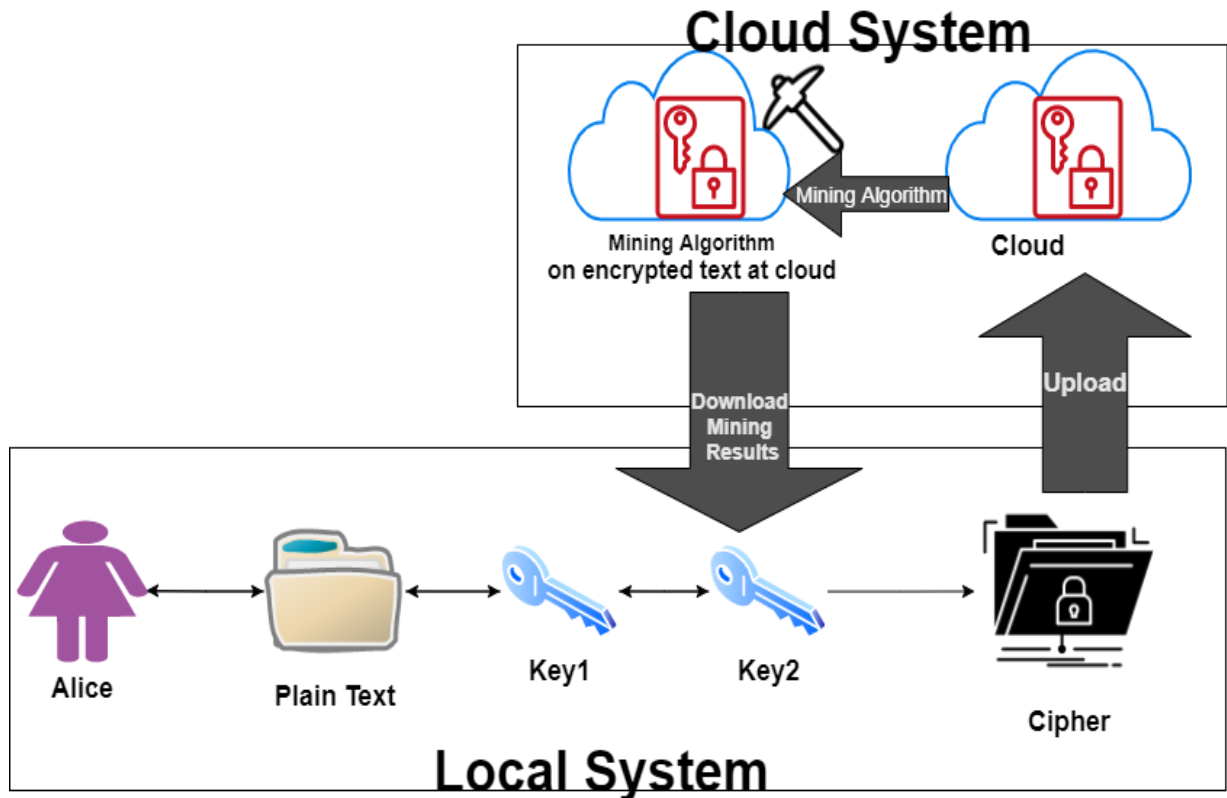


Figure 3-1 Adopted Scenario

This thesis seeks to maintain security of DM, and to ensure security of data extracted from mining processes, the methodology depends on encryption the information after getting that information from the security models. Encrypting and mining data as text; will restrict data, which is extracted from databases from unauthorized access to protect sensitive data. This restriction makes security more reliable and robust by reducing the surface area of the overall security system. In addition.

Fig.3-2 shows another scenario might be adopted using same Alice to Alice encryption, but the data owner is a client referred to the encryptor to save his data

encrypted in cloud. So, if the client has been removed from the figure 3-2, figure 3-1 would pop up.

Fig.3-3 shows other side of Client/User relation in this thesis technique.

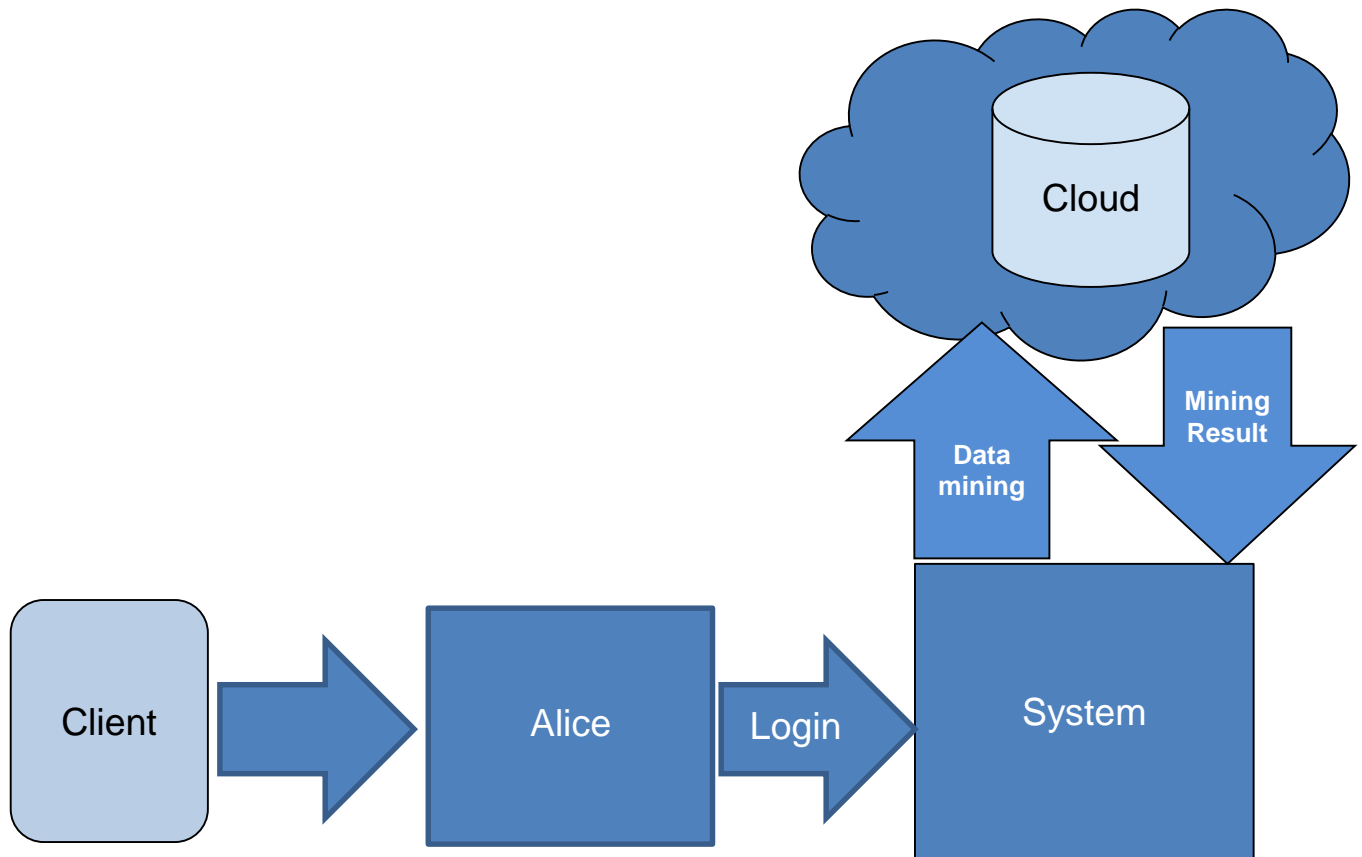


Figure 3-2 Client/Customer Scenario for a Customer

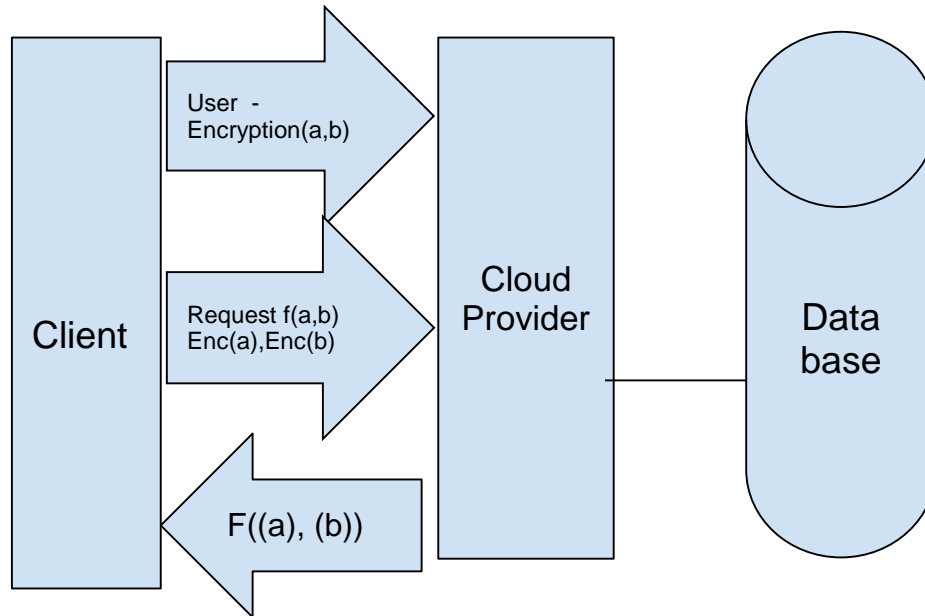


Figure 3-3 Other side of Client/Consumer Scenario

3.2.2 Proposed Technique

We propose the following:

- Step1: Data Indexing (Offline).
- Step2: Data encryption (Offline).
- Step3: Sends Data to CSP
- Step4: Data Indexing
- Step5: Data mining
- Step6: Retrieve Encrypted Mining Results
- Step7: Receive Data from Cloud.
- Step8: Decrypt information (Offline)

3.2.2.1 Cloud Environment

In this thesis the author used Replit as his cloud server. Replit uses both Google Cloud and Heroku. Hence Replit hosted by a platform as a service (PaaS) called Heroku (which uses amazon web service/AWS under the hood).

The code compilation/evaluation/hosting infrastructure are on Google Cloud. Google cloud is similar to AWS as it's an IaaS (infrastructure as a service). (*What Kind of Servers Do You Use? What Are the Specs :P - Replit, n.d.*)

Figure 3-4 shows user login process in local system while figure 3-5 shows user login process in cloud system.

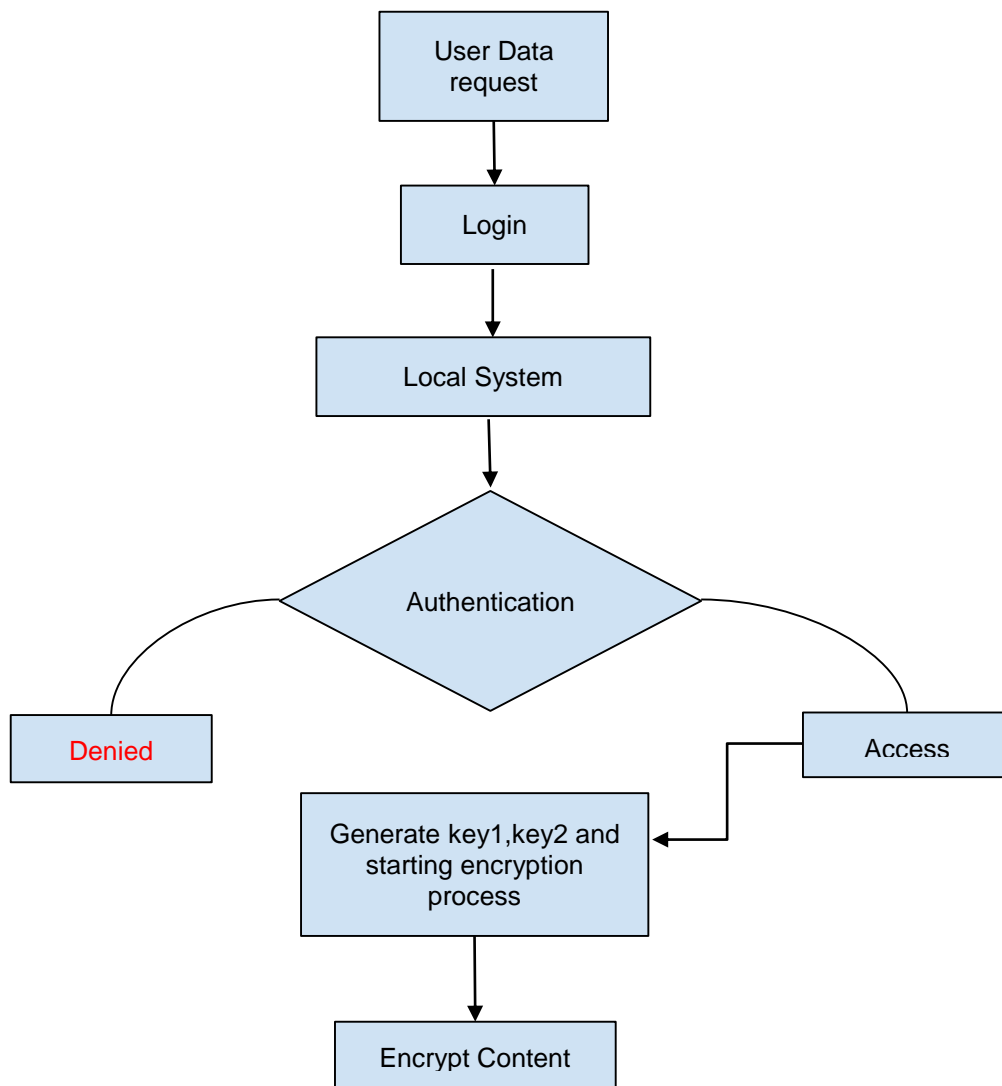


Figure 3-4 Proposed User Login Process (Local System)

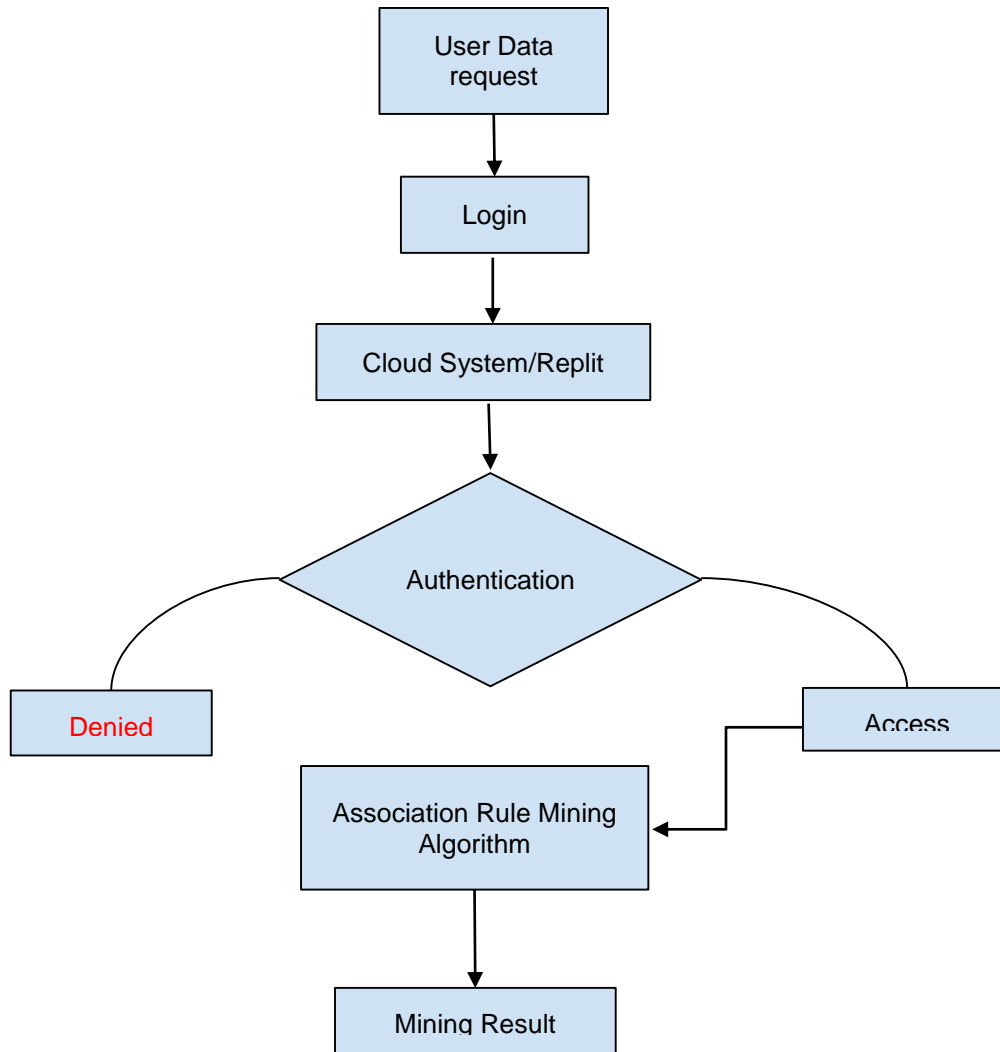


Figure 3-5 Proposed User Login Process (Cloud System)

3.2.2.2 Amplifying/Incrementing Process

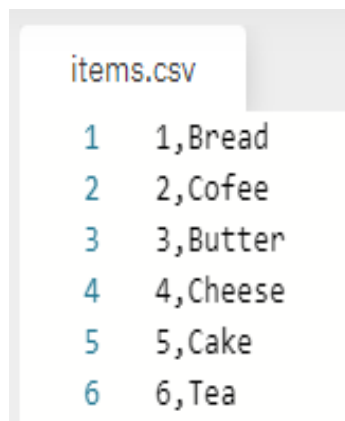
The original text used in this thesis is emulating real items of a store to add some reliability for the thesis sail. Encrypting the original text that is downloaded from (GitHub) and conducting data mining techniques on the generated cipher depending on the index of the item and the inability to use block cipher technique prevented me to use large key size for encryption. Especially there are some items consist of only 3 characters like (Ice). And to make the thesis keeps up with real transactions of a store when there is only one item in a transaction.

Encrypting such items row by row independently using key length equal to the original text shall represent a weakness in the encryption technique and make it vulnerable to brute-force attack (Kumar & Sharma, n.d.).

As example, “Milk” with only four characters is an item could be written as: “Almarai milk full fat 250ml” with twenty-seven characters/27 bytes, adding some of the barcode numbers of the item “Milk” would rise the item to reach 32 characters/32 bytes easily. Under these circumstances, the data editor shall be responsible to write the data carefully according the encryptor’s need. So, the increments are not a real at many aspects but to increase the data set. So, for the easiness than adding characters manually, the author has used Amplifying technique to add these characters/bytes automatically.

The author has used characters/symbols from 224 to 243 in ASCII range to be added as a separator between the original texts and the added characters (Amplifying process). Beside this, this symbol shall differentiate between the original text and the incremented text. That helped me to recognize the original texts when decrypting. This range shall limit the resulting cipher but even with this constrained addition the encryption algorithm results are encouraging as shown in chapter 4.

Figure 3-6 shows some items not long enough to get encrypted independently, hence that shall shorten the key. So, the author would add some imaginary texts as addition to the original item as reasoned above. Note that each row contains one item only and not to contain more than one item and the item length should be written carefully not to exceed 32 characters as maximum as explained next.



```
items.csv
1 1,Bread
2 2,Cofee
3 3,Butter
4 4,Cheese
5 5,Cake
6 6,Tea
```

Figure 3-6 A comma-separated values/CSV File

So, if one imagined there are more than one item in a row like bread and cake in one row numbered 1, as example:

1 Bread Cake

The encryption method would treat them as a one item (Block), and that would lead to inability of executing mining process.

3.2.2.2.1 Amplifying to 16 characters in each row

In this scenario, the author has increased each row has less than 16 characters to be at least 16 characters, and not to change any row has more than or equal 16 characters.

Based on these increments, the key length shall be semi dynamic. It will be either 16 bytes or more than 16 bytes. And the resulting cipher will vary from 16 to 32 bytes.

3.2.2.2.2 Amplifying to 16 and 32 characters in each row

In this scenario, the author has increased each row has less than 16 characters to be at least 16 characters, and each row has more than or equal 16 characters to be 32 characters.

Based on these increments, the key length shall be binary dynamic. It will be either 16 bytes or 32 bytes. And the resulting cipher will be either 16 or 32 bytes.

3.2.2.2.3 Amplifying to 32 characters in each row

In this scenario, the author has increased each row to be at least 32 characters, and each row has less, more than or equal 16 characters to be 32 characters.

Based on these increments, the key length shall be constant/fixed. It will be 32 bytes always. And the resulting cipher will be 32 bytes.

The Pseudo Code for Increments/Amplifying data is as follows:

```

amplify(int size){
    if plain.length >= size:
        return;
    initialize amplified_plain to the plain text
    declare char temp[size]
    declare char symbols[20]

    for i = 1 --> 20:
        symbols[i] = i + 224

    initialize separator to random number between [1, 20] inclusive
    temp[0] = symbols[separator]

    for i = 1 --> size - amplified_plain.length:
        temp[i] = (char) random decimal number between [32, 127] inclusive

    amplified_plain = amplified_plain + temp[1:size]
}

```

The Pseudo Code for decrease words to its normal number of characters/Disamplify data is as follows:

```

disamplify (){
    for i = 1 --> decrypted_word.length:
        if (int) decrypted[i] not in [32, 127]
            decrypted = decrypted[1 : i]
}

```

3.3 Encryption process

Generate Secret two keys for each row/item → Encryption → Decryption. The process of encryption consists of the following:

3.3.1 Shuffling

It is a process of redistribute each character of the plaintext, in my CSV file there are some words like **Butter** and **Apple**. When same encryption key used in the whole characters of each word without shuffling, the encryption of the two words (as example) would be:

Butter → *\$@@^!

Apple → ₣®®£≠

This shall represent a fragility in any encryption uses the same key/byte for the word/block. Despite used different key/byte for each character in same word in this thesis, the author applied shuffling to the plain text to add some randomization to the cipher text.

After using the shuffling, the above example shall be as follows:

Butter → **trBute** → @!*\$@^

Apple → **lpepA** → £®≠® ₣

The precise shuffling process could be explained using the word “Bread” as it shall represent the plain text, shuffles its character, as follows:

- writes the plaintext as an array below. Tables 3-1, 3-2 and 3-3 show the steps of shuffling:

B	r	e	a	d
0	1	2	3	4

Table 3-1

I shall move upon each character using the function:

**index(character)= swap (integer(index(character),(index(last character-
index(character)/2)**

index(B)=swap(integer(index(B),index(B)-index(d)/2) →

index(B)=swap(integer(0,4-0/2)) = swap(0,2) = (2,0)=(e,B)

e	r	B	a	d
----------	----------	----------	----------	----------

Table 3-2

The final result after applying the function on all characters will be:

d	B	r	e	a
----------	----------	----------	----------	----------

Table 3-3

The Pseudo Code for Shuffling is as follows:

```
string shuffle(str){
  initialize last to the last index in str

  for i = 1 --> last:
    swap between s[i] and s[(last - i)/2]

  return str
}
```


The Pseudo Code for Reorganize/Unshuffling data is as follows:

```
string unshuffle(str){
  initialize last to the last index in str

  for i = last --> 1:
    swap between s[i] and s[(last - i)/2]

  return str
}
```

3.3.2 Bits shifting

The author has rotated the bits of each byte/character one bit to the left side. This will change the locations of each bit in the text one bit to the left.

3.3.3 Bits Substitutions

After shifting the bits, substituted each bit in every byte. So, each one shall be replaced with zero and in return each zero shall be replaced with one.

3.3.4 Key Generation

The generated key shall be the same for each similar word, not to be changed in each similar word while encryption. That will; off course; reduce the security level of the algorithm but no way if needs to apply association rule and need to query while text is encrypted.

Because of the analytical operations included in this thesis, key generation function and its length represented a challenge in this thesis, since cannot use the keys used in block cipher, because each encrypted item must be in separated row with different indexing number as shown in figure 3-6.

The length of the key shall be dynamic as explained above in 3.2.2.2 (Amplifying/incrementing Process).

The algorithm contains two dependent keys to have the encryption achieved. Also, both keys are used in the decryption process. the author has used two keys to have more sophisticated cipher. It is as if encrypts the plain text twice.

3.3.4.1 Key 1 Generation

The first key shall be created through the use of pseudo random function. The seed used for this function is the first character of the username. the reason to choose this seed is that it must be known in both direction (encryption/decryption). The first key represents a part of generating the second key. Choosing the first character rather than the whole username characters shall assist to avoid falling in problem of repeated characters in username. As shown below:

- Username: Amer → “A” shall be the seed for the random function. Username (Amer) represents an appropriate username even if selected the whole characters of the username as a seed.
- Username: mmmmm → “m” shall be the seed for the random function. Username (mmmmm) represents an inappropriate username if selected the whole characters of the username as a seed.

3.3.4.2 Key 2 Generation

The second key shall be created through the use of pseudo random function. The seed used for this function is a combination of the first character of the password and the xor result of the first key. The reason to choose this combination is that it must be known in

both direction (encryption/decryption). Choosing the first character of the password and the xor result of the first key as a seed rather than the whole password characters with the xor result of the first key shall assist to apply more sophistication on the second key and to avoid falling in problem of repeated characters in password. As shown below:

- Password: \$mr08713 → “\$” shall be a part of the seed for the random function. Password (\$mr08713) represents an appropriate password even if selected the whole characters of the password as a part of the seed.
- Password: m1m1m → “m” shall be the seed for the random function. Password (m1m1m) represents an inappropriate password if selected the whole characters of the password as a part of the seed.

3.3.5 Encryption Steps (Pseudo Code)

read username and **password**
initialize *plain* to the plain text

shuffle the amplified plain by calling `shuffle(amplified_plain)`

initialize *seed1* to the first character of username
initialize *seed2* to the first character of password
initialize *upper* to `shuffled_amplified_plain.length`

declare char *keys1[upper]*
declare char *random_number*
for *i* = 1 --> *upper*:
 random_number = number generated randomly based on *seed1*
 keys1[i] = *random_number*
initialize *xor_keys1* to *keys1[0]*
for *i* = 1 --> *upper*:
 xor_keys1 = *xor_keys1* XOR *keys1[i]*

seed2 = *seed2* XOR *xor_keys1*

initialize *temp1* and *temp2* to *shuffled_amplified_plain*
initialize *encrypted* to the same length of *shuffled_amplified_plain*

for $i = 1 \rightarrow$ upper:

temp1[i] = circular rotate left of temp1[i] by 1

temp1[i] = (flip 0's and 1's) substitute of temp[i]

temp2[i] = temp1[i] ^ key1[i]

initialize *key2* to random number generated based on *seed2*

encrypted[i] = temp2[i] ^ key2

change *encrypted* to Hexadecimal representation

3.3.5.1 Flowchart (Encryption)

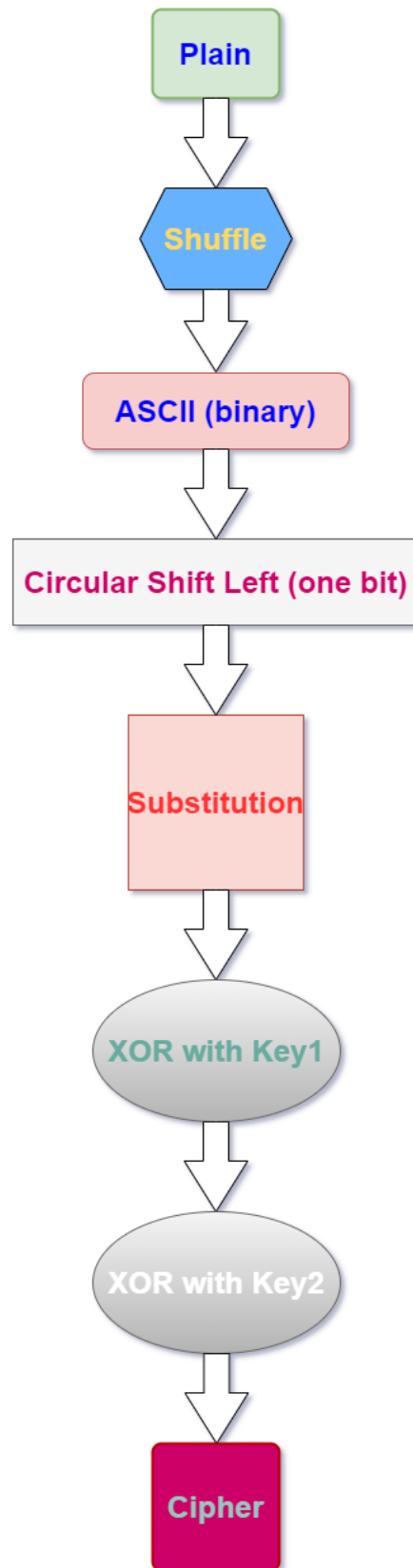


Figure 3-7 Flow Chart of Encryption

3.3.6 Decryption Process (Pseudo Code)

```

read username and password
initialize cipher to the cipher text
convert cipher from Hexadecimal to ASCII characters

initialize seed1 to the first character of username
initialize seed2 to the first character of password
initialize upper to cipher.length

declare char keys1[upper]
declare char random_number
for i = 1 --> upper:
    random_number = number generated randomly based on seed1
    keys1[i] = random_number

initialize xor_keys1 to keys1[0]
for i = 1 --> upper:
    xor_keys1 = xor_keys1 XOR keys1[i]

seed2 = seed2 XOR xor_keys1

initialize decrypted to the same length of cipher
initialize temp1 and temp2 to cipher
for i = 1 --> upper:
    initialize key2 to random number generated based on seed2
    temp1[i] = cipher[i] XOR key2;
    temp2[i] = temp1[i] XOR keys1[i]
    temp2[i] = (flip 0's and 1's) substitute of temp2[i]
    temp2[i] = circular rotate right of temp2[i] by 1
    decrypted[i] = temp2[i]

unshuffle decrypted word by calling unshuffle(cipher)
retrieve the original length of the word by calling disamplify()

```

3.3.6.1 Flowcharts (Decryption)

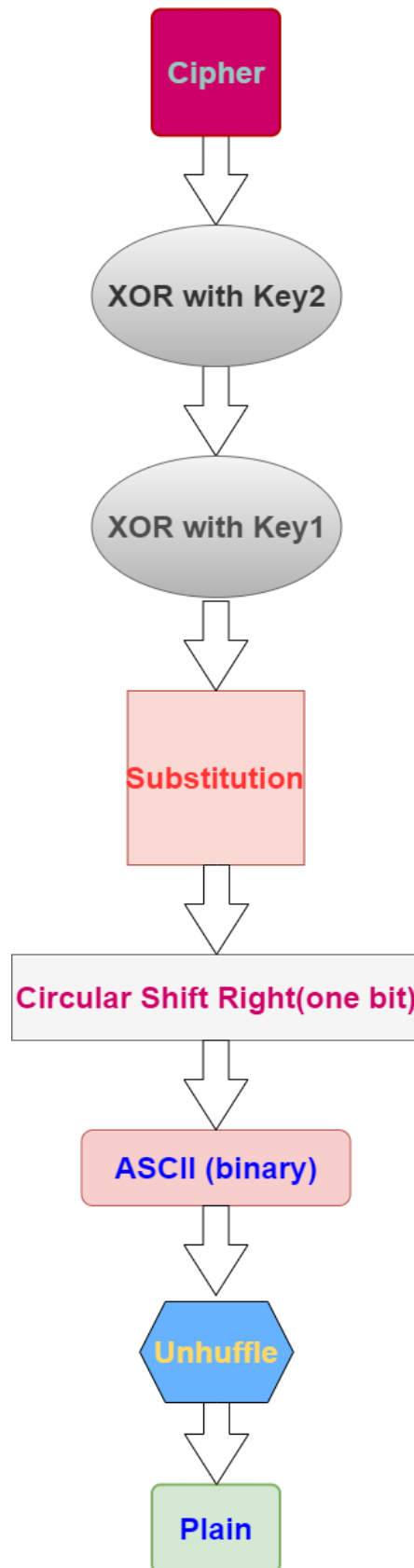


Figure 3-8 Flow Chart of Decryption

3.3.7 Applying Association Rule on Cipher

As explained in chapter two/2.2.1 (Data mining techniques) of this thesis, there are many kinds of association rules. The author has used Apriori algorithm/Affinity analysis (Market Basket Analysis) to be applied on the cipher text.

Based on the location of each item encryption in the encrypted table in cloud, can have analytical operations based on the index used for each row of the item. In another word, mirrored the plain text file in local system as cipher text in cloud system.

A row-by-row encryption would assist to have precise results when applying association rule on both plain and cipher. Both would give same results. Mirroring the C.S.V file of plain text in local system to a C.S.V file of ciphertext in cloud system with same index numbers for each item is the key to conduct any operations of this thesis. Figure 3-9 shows the mirroring concept. The operations on encrypted text would give me (after decrypting) the same results when apply the same operations on plaintext.

Local System		Mirroring	Cloud System	
Item Number	Plain		Item Number	Cipher
1	Bread	→	1	G+ L _{ff} é
2	Coffee	→	2	E①FFÇÇ
3	Cheese	→	3	E ^J ÇÇ)Ç
4	Butter	→	4	ÚÛ④④s^
5	Cake	→	5	Eë⑩Ç
6	Tea	→	6	ôç

Figure 3-9 Concept of C.S.V file in local and cloud systems

Additionally, writing the cipher text of the item while conducting these operations is considered slower than using the index/row number of the same item instead. So, instead of writing “Bread” or “G+ L_{ff}é”, the author would write “1” to query about the item.

Below in Figures 3-10 and 3-11, are examples of random transactions in a market, each transaction shall be executed directly on cipher at cloud environment unlike the encryption of the items which conducted at local environment. Likewise, the affinity rule shall be implemented on cipher at cloud, compute the support and confidence for specific items using affinity rule.

Trans.ID	Items Sold		
1548	G+ L _{ffé}	E①FFÇÇ	E ^J ÇÇ)Ç
1356	G+ L _{ffé}	E ^J ÇÇ)Ç	
1269	G+ L _{ffé}	ÚÚ④ ^{oo} x _u	
1574	E①FFÇÇ	Eä⑩Ç	ôç-

Figure 3-10 Random Transactions Conducted on Cipher

Frequent Itemset	Support	Percentage
G+ L _{ffé}	3/4	75%
E①FFÇÇ	2/4	50%
E ^J ÇÇ)Ç	2/4	50%
G+ L _{ffé} E ^J ÇÇ)Ç	2/4	50%

Figure 3-11 Encrypted Items Repeated in Transactions (Support)

For Rule " G+ L_{ffé} " To " E^J ÇÇ)Ç "

$$\text{support} = \text{support} ("G+ L_{ffé} " \wedge " E^J \text{ÇÇ)Ç} ") = 50\%$$

$$\text{Confidence} = \text{support} ("G+ L_{ffé} " \wedge " E^J \text{ÇÇ)Ç} ") / \text{support} ("G+ L_{ffé} ") = 50\% \div 75\% = 67\%$$

3.3.8 Applying Association Rule on Plain Text

Below in Figures 3-12 and 3-13, are author's examples of random transactions in a market, each transaction shall be executed on plain. Likewise, the affinity rule shall be implemented on plain, compute the support and confidence for specific items using affinity rule.

Trans.ID	Items Sold		
1548	Bread	Coffee	Cheese
1356	Bread	Cheese	
1269	Bread	Butter	
1574	Coffee	Cake	Tea

Figure 3-12 Random Transactions Conducted on Plain

Frequent Itemset	Support	Percentage
Bread	3/4	75%
Coffee	2/4	50%
Cheese	2/4	50%
Bread, Cheese	2/4	50%

Figure 3-13 Items Repeated in Transactions (Support)

For Rule "Bread" To "Cheese"

$$\text{support} = \text{support}(\text{"Bread" } \wedge \text{"Cheese"}) = 50\%$$

$$\text{Confidence} = \text{support}(\text{"Bread" } \wedge \text{"Cheese"}) / \text{support}(\text{"Bread"}) = 50\% \div 75\% = 67\%$$

Both examples have given me same results to compute support and confidence. Note that the author has encrypted only the text not the numbers or percentages used in the rule.

3.3.9 Avalanche Effect

Once an input is morphed slightly (a single bit flipping), the output must be changed remarkably (50% of the output bits flip). In which a small change in either the key or the plaintext should lead a harsh change in the cipher. The below equation is a must to calculate avalanche effect (Aljawarneh et al., 2017):

$$\frac{\text{No of flipping bits in cipher text}}{\text{No of bits in cipher text}} \times 100\% \geq 50\%$$

While it is a must when using Block Cipher (Cryptography and Computer Privacy, n.d.), and despite the algorithm used in this thesis cannot be attributed to Block Cipher, but the author has calculated the Avalanche Effect of the algorithm based on three different length modes as explained in 3.2.2.2 (Amplifying/Incrementing Process). Results to be shown in next chapter

CHAPTER FOUR: Implementation and Experimental Results

4.1 Local and Cloud Systems Specifications

This chapter demonstrates the implementation of the proposed model and the experimental results of the proposed algorithm. The proposed algorithm is coded using C++ programming language using downloaded C++ environment to encrypt data locally then send them to cloud and apply the Affinity Rule on cipher in cloud.

The local device used in encryption has below specifications:

View basic information about your computer

Windows edition

Windows 10 Pro

© Microsoft Corporation. All rights reserved.

System

Processor:	Intel(R) Core(TM) i7-1065G7 CPU @ 1.30GHz 1.50 GHz
Installed memory (RAM):	8.00 GB (7.77 GB usable)
System type:	64-bit Operating System, x64-based processor
Pen and Touch:	No Pen or Touch Input is available for this Display

Figure 4-1 Local Environment Specifications

Replit shall represents the cloud environment, it is an online-integrated development environment (IDE). Replit allows users to write code and build apps using a browser, Replit has various collaborative features such as a capability for real-time programming, code-hosting platform, this helps to deal with C.S.V files as a cloud data.

4.2 Data Used

the author has downloaded the data from GitHub, the link is:

<https://github.com/stedy/Machine-Learning-with-R-datasets/blob/master/groceries.csv>

Basically, downloaded data contains transactions (not items listed properly as wanted in this thesis), so the author has rewritten the file to be in a proper manner to fit with the adopted encryption method. The final result of the C.S.V is 503 words with a file sized 6.72 kb before amplifying.

4.3 Encryption Module

As explained, the encryption process performed locally to generate the ciphertext that will be stored in the cloud. The encryption module would secure the items in cloud storage.

The author has encrypted different size of data after enlarge the same revised file, using 256 double key algorithm, and the timing for each size was as follows:

- 500 items/rows of data \longrightarrow time was 115 milliseconds, plain text file size was 17 kb, as shown in figure 4-2 below.

```
Microsoft Visual Studio Debug Console

username: Amer
password: $mr08713
-----
Encrypting in progress...
please wait...
items count = 500

#####
#                                     #
# Time Taken: 115 milliseconds        #
#                                     #
#####

Finished.....
```

Figure 4-2 Time Needed to Encrypt 500 Items

- 1000 items/rows of data \longrightarrow time was 216 milliseconds, plain text file size was 33 kb, as shown in figure 4-3 below.

```
Microsoft Visual Studio Debug Console
username: Amer
password: $mr08713
-----
Encrypting in progress...
please wait...
items count = 1000

#####
#                                     #
# Time Taken: 216 milliseconds        #
#                                     #
#####

Finished.....
```

Figure 4-3 Time Needed to Encrypt 1000 Items

- 5000 items/rows of data \longrightarrow time was 955 milliseconds, plain text file size was 162 kb, as shown in figure 4-4 below.

```
Microsoft Visual Studio Debug Console
username: Amer
password: $mr08713
-----
Encrypting in progress...
please wait...
items count = 5000

#####
#                                     #
# Time Taken: 955 milliseconds         #
#                                     #
#####

Finished.....
```

Figure 4-4 Time Needed to Encrypt 5000 Items

- 10000 items/rows of data \longrightarrow time was 1763 milliseconds, plain text file size was 323 kb, as shown in figure 4-5 below.

```
Microsoft Visual Studio Debug Console
username: Amer
password: $mr08713
-----
Encrypting in progress...
please wait...
items count = 10000

#####
#                                     #
# Time Taken: 1763 milliseconds       #
#                                     #
#####

Finished.....
```

Figure 4-5 Time Needed to Encrypt 10000 Items

- 50000 items/rows of data \longrightarrow time was 8645 milliseconds, plain text file size was 1612 kb, as shown in figure 4-6 below.

```
Microsoft Visual Studio Debug Console

username: Amer
password: $mr08713
-----
Encrypting in progress...
please wait...
items count = 50000

#####
#                                     #
# Time Taken: 8645 milliseconds      #
#                                     #
#####

Finished.....
```

Figure 4-6 Time Needed to Encrypt 50000 Items

- 100,000 items/rows of data \longrightarrow time was 16955 milliseconds, plain text file size was 3223 kb, as shown in figure 4-7 below.

```
Microsoft Visual Studio Debug Console
username: Amer
password: $mr08713
-----
Encrypting in progress...
please wait...
items count = 100000

#####
#                                     #
# Time Taken: 16955 milliseconds      #
#                                     #
#####

Finished.....
```

Figure 4-7 Time Needed to Encrypt 100,000 Items

- 500,000 items/rows of data \longrightarrow time was 85419 milliseconds, plain text file size was 16114 kb, as shown in figure 4-8 below.

```
Microsoft Visual Studio Debug Console

username: Amer
password: $mr08713
-----
Encrypting in progress...
please wait...
items count = 500000

#####
#                                     #
# Time Taken: 85419 milliseconds      #
#                                     #
#####

Finished.....
```

Figure 4-8 Time Needed to Encrypt 500,000 Items

- 1,000,000 items/rows of data \longrightarrow time was 170555 milliseconds, plain text file size was 32227 kb, as shown in figure 4-9 below.

```

Microsoft Visual Studio Debug Console

username: Amer
password: $mr08713
-----
Encrypting in progress...
please wait...
items count = 1000000

#####
#                                     #
#   Time Taken: 170555 milliseconds   #
#                                     #
#####

Finished.....

```

Figure 4-9 Time Needed to Encrypt 1,000,000 Items

In Figure 4-10 bellow, the author has demonstrated all count of items encrypting time in one chart.

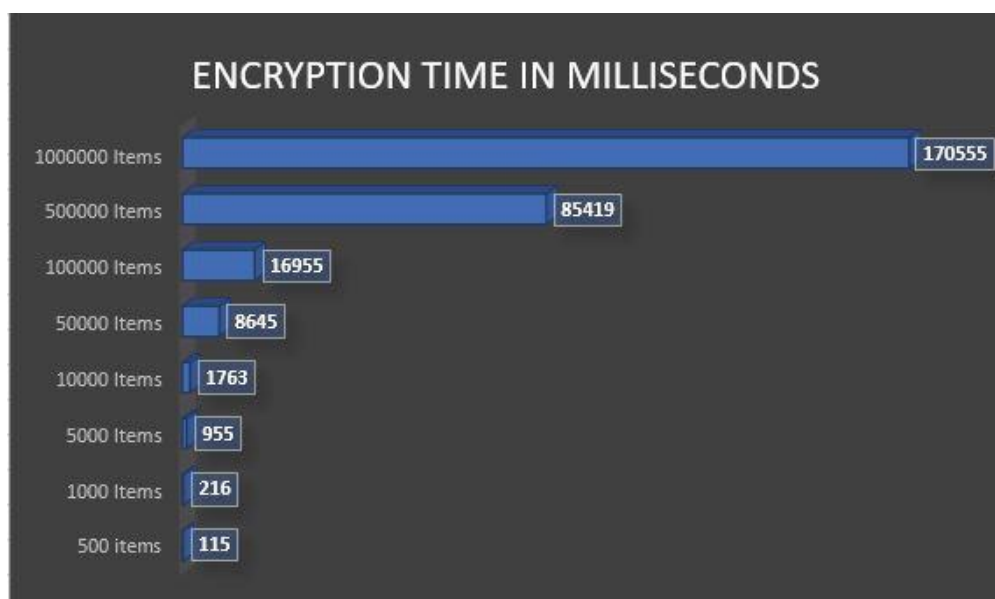


Figure 4-10 Chart Shows Time Needed to Encrypt Different Count of Items

4.4 Decryption Module

In this module, the cipher text returned into its original plain text.

When the user mines the cipher data in cloud, downloads the results locally as a ciphertext and then decrypt the results locally. It means, uploading data, stored data, the mining process, and the results of the mining process all will be in a cipher mode, which grants the whole thesis scenario an enhanced protection from any data leakage and unauthorized access to query and the three states of data (data at rest, data in motion and data in use).

The author has decrypted all the encrypted data as explained in 4.3, using 256 double key algorithm, and the timing for each size was as follows:

- 500 items/rows of data \longrightarrow time was 116 milliseconds, cipher text file size was 40 kb, as shown in figure 4-11 below.



```

Microsoft Visual Studio Debug Console

username: Amer
password: $mr08713
-----
Decrypting in progress...
please wait...
items count = 500

#####
#                                     #
# Time Taken: 116 milliseconds #
#                                     #
#####

Finished.....

```

Figure 4-11 Time Needed to Decrypt 500 Items

- 1000 items/rows of data \longrightarrow time was 199 milliseconds, cipher text file size was 64 kb, as shown in figure 4-12 below.

```
Microsoft Visual Studio Debug Console

username: Amer
password: $mr08713
-----
Decrypting in progress...
please wait...
items count = 1000

#####
#                                     #
# Time Taken: 199 milliseconds #
#                                     #
#####

Finished.....
```

Figure 4-12 Time Needed to Decrypt 1000 Items

- 5000 items/rows of data \longrightarrow time was 855 milliseconds, cipher text file size was 318 kb, as shown in figure 4-13 below.

```
Microsoft Visual Studio Debug Console

username: Amer
password: $mr08713
-----
Decrypting in progress...
please wait...
items count = 5000

#####
#                                     #
# Time Taken: 855 milliseconds #
#                                     #
#####

Finished.....
```

Figure 4-13 Time Needed to Decrypt 5000 Items

- 10000 items/rows of data \longrightarrow time was 1775 milliseconds, cipher text file size was 635 kb, as shown in figure 4-14 below.

```
Microsoft Visual Studio Debug Console
username: Amer
password: $mr08713
-----
Decrypting in progress...
please wait...
items count = 10000

#####
#                                     #
# Time Taken: 1775 milliseconds #
#                                     #
#####

Finished.....
```

Figure 4-14 Time Needed to Decrypt 10000 Items

- 50000 items/rows of data \longrightarrow time was 8391 milliseconds, cipher text file size was 3174 kb, as shown in figure 4-15 below.

```
Microsoft Visual Studio Debug Console

username: Amer
password: $mr08713
-----
Decrypting in progress...
please wait...
items count = 50000

#####
#                               #
# Time Taken: 8391 milliseconds #
#                               #
#####

Finished.....
```

Figure 4-15 Time Needed to Decrypt 50000 Items

- 100000 items/rows of data \longrightarrow time was 16888 milliseconds, cipher text file size was 6348 kb, as shown in figure 4-16 below.

```
Microsoft Visual Studio Debug Console
username: Amer
password: $mr08713
-----
Decrypting in progress...
please wait...
items count = 100000

#####
#                               #
# Time Taken: 16888 milliseconds #
#                               #
#####

Finished.....
```

Figure 4-16 Time Needed to Decrypt 100000 Items

- 500000 items/rows of data \longrightarrow time was 84311 milliseconds, cipher text file size was 31739 kb, as shown in figure 4-17 below.

```
Microsoft Visual Studio Debug Console

username: Amer
password: $mr08713
-----
Decrypting in progress...
please wait...
items count = 500000

#####
#                                     #
# Time Taken: 84311 milliseconds #
#                                     #
#####

Finished.....
```

Figure 4-17 Time Needed to Decrypt 500000 Items

- 1000000 items/rows of data \longrightarrow time was 168411 milliseconds, cipher text file size was 63477 kb, as shown in figure 4-18 below.

```

Microsoft Visual Studio Debug Console

username: Amer
password: $mr08713
-----
Decrypting in progress...
please wait...
items count = 1000000

#####
#                                     #
# Time Taken: 168411 milliseconds #
#                                     #
#####

Finished.....

```

Figure 4-18 Time Needed to Decrypt 1000000 Items

Figure 4-19 shows time needed to decrypt different count of items.

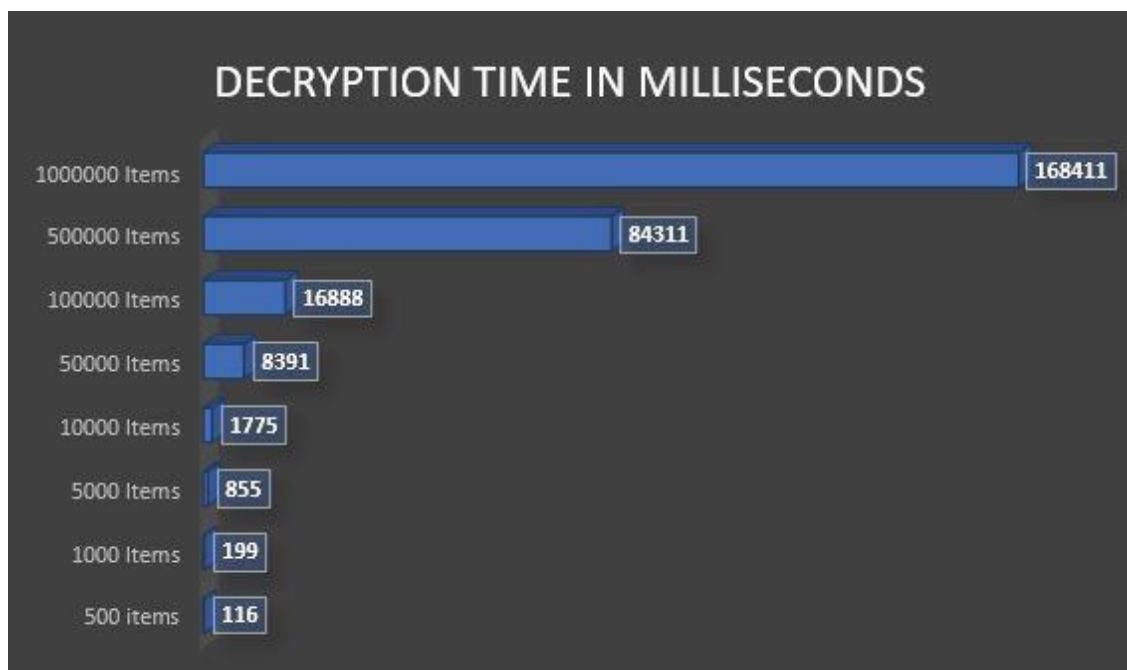


Figure 4-19 Decryption Chart

Table 4-1 shows the differences in file sizes of plaintext vs ciphertext for the same count of items with encryption vs decryption time comparison. Because of use of hexadecimal characters, file size of ciphertext has a double size of plaintext file when comparing the two file sizes of plaintext vs ciphertext generated from same plaintext. In another word, each character of plain text would be two characters in cipher. But the time spent in decryption is almost lesser than time spent in encryption as shown in the table.

Items Count	Encryption		Decryption	
	Plain Size in kb	Time in Milliseconds	Cipher Size in kb	Time in Milliseconds
500 items	17	115	32	116
1000 Items	33	216	64	199
5000 Items	162	955	318	855
10000 Items	323	1763	635	1775
50000 Items	1612	8645	3174	8391
100000 Items	3223	16955	6348	16888
500000 Items	16114	85419	31739	84311
1000000 Items	32227	170555	63477	168411

Table 4-1 Encryption vs Decryption

figure 4-20 a chart shows differences in file sizes of plaintext vs ciphertext for the same count of items with encryption vs decryption time comparison.

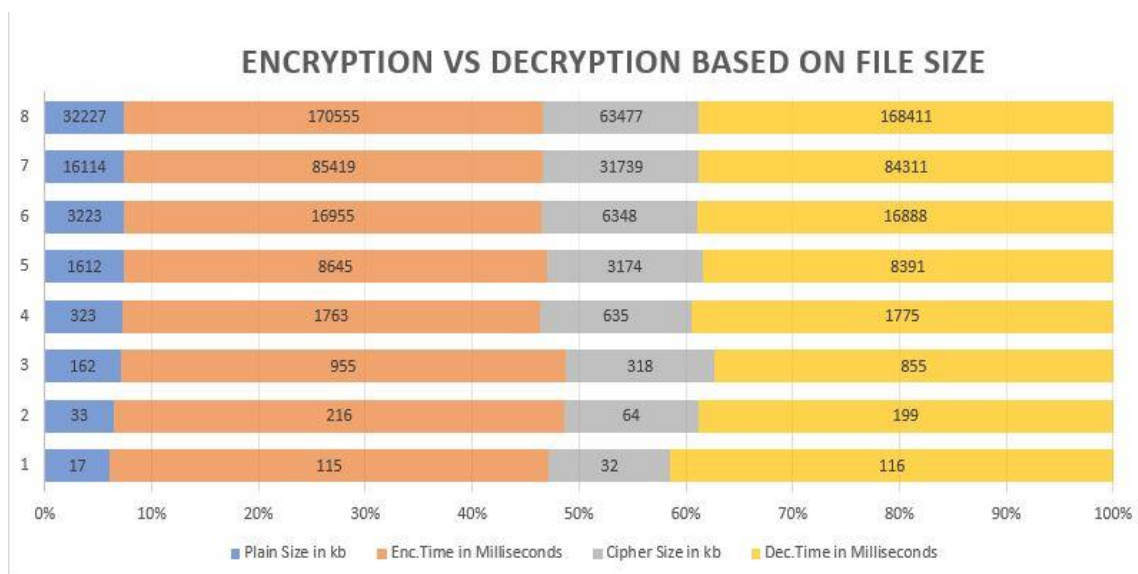


Figure 4-20 Plain vs Cipher in Size and Time

4.5 Association Rule

As explained in chapter two/data mining techniques/association rules. The author applied the Affinity rule on cipher as shown in the figures below on plain and cipher.

Proposed all transactions shall be implemented in cloud environment at ciphertext. However, to add reliability for this thesis, applied the transactions in local environment a plaintext to compare the results if they were true or not.

Both results obtained from the plaintexts and ciphertexts are identical. In figures 4-21 and 4-22, the transactions executed in cloud at cipher. Proposed the question shown in figure 4-21 “how many transactions do you have” which could be replaced in real applications through the use of barcode reader.

The system has four transactions, the first transaction has three items indexed (0,1,2). The second transaction has 2 items indexed (0,2). The third transaction has two items indexed (0,3). The fourth transaction has three items indexed (1,4,5).


```

C:\Users\Dell\Desktop\cipher\Debug\cipher.exe
items count = 50000
How many transactions do you have? : 4

How many items in transaction #1?: 3
enter item #1 : 0
item: 57d18604eff138fbb23b9b19f810e970bb23bb43eeeeae93340d839370728e9a0
enter item #2 : 1
item: b7d3b440791384b9883219b7dabccd7efe0f3343ec0e83e7463c3f150516d51c
enter item #3 : 2
item: b7d3b440791384b9863219b7dabccd78fe0f3343c00e83e7403c3f150516d51c

How many items in transaction #2?: 2
enter item #1 : 0
item: 57d18604eff138fbb23b9b19f810e970bb23bb43eeeeae93340d839370728e9a0
enter item #2 : 2
item: b7d3b440791384b9863219b7dabccd78fe0f3343c00e83e7403c3f150516d51c

How many items in transaction #3?: 2
enter item #1 : 0
item: 57d18604eff138fbb23b9b19f810e970bb23bb43eeeeae93340d839370728e9a0
enter item #2 : 3
item: b7d3b440791384b9bc1c19b7dabccd5afe0f3343ec0e83e7623c3f150716d51c

How many items in transaction #4?: 3
enter item #1 : 1
item: b7d3b440791384b9883219b7dabccd7efe0f3343ec0e83e7463c3f150516d51c
enter item #2 : 4
item: 15457a06e3cbb2eb9c920d4f389aa3717d2fc9b1765c437548ee39252be00d3a
enter item #3 : 5
item: abe3700453776e3794a8d195c6accb7857378187e5e4a3e35c1a97e305da9b80

    transaction #1 --> 3
57d18604eff138fbb23b9b19f810e970bb23bb43eeeeae93340d839370728e9a0
b7d3b440791384b9883219b7dabccd7efe0f3343ec0e83e7463c3f150516d51c
b7d3b440791384b9863219b7dabccd78fe0f3343c00e83e7403c3f150516d51c
    transaction #2 --> 2
57d18604eff138fbb23b9b19f810e970bb23bb43eeeeae93340d839370728e9a0
b7d3b440791384b9863219b7dabccd78fe0f3343c00e83e7403c3f150516d51c
    transaction #3 --> 2
57d18604eff138fbb23b9b19f810e970bb23bb43eeeeae93340d839370728e9a0
b7d3b440791384b9bc1c19b7dabccd5afe0f3343ec0e83e7623c3f150716d51c
    transaction #4 --> 3

```

Figure 4-21 Applying Affinity Rule in Cloud #1

After completion of reading of the transactions, “enter your choice” pops up to select the operation needed next. Option 1 “calculate support” and option 2 “calculate

confidence” are belonging to affinity rule mining or end the program if not interested in calculation.

By selecting option 1, “how many items” pops up, then selecting the items the author aims to calculate support for them. Selected items indexed (0,2). The support for these to items is 50% since they are repeated twice together in 4 transactions.

No selecting option 2 to calculate confidence for the same items indexed (0,2). The result is 66%.

```

transaction #4 --> 3
b7d3b440791384b9883219b7dabccd7efe0f3343ec0e83e7463c3f150516d51c
15457a06e3cbb2eb9c920d4f389aa3717d2fc9b1765c437548ee39252be00d3a
abe3700453776e3794a8d195c6accb7857378187e5e4a3e35c1a97e305da9b80

Enter your choice:
    1. calculate support
    2. calculate confidence
    3. end program
>> 1
how many items? : 2
enter items:
0 2
support = %50
Time taken: 2 milliseconds
Enter your choice:
    1. calculate support
    2. calculate confidence
    3. end program
>> 2
how many items? : 2
enter items:
0 2
enter base item:
0
confidence = %50 / %75 = %66
Time taken: 6 milliseconds

```

Figure 4-22 Applying Affinity Rule in Cloud #2

The same steps shall be repeated in local system at plaintexts as shown in figures 4-23 and 4-24. And it is clear that all the results are identical.

```
C:\Users\Dell\Desktop\cipher\Debug\cipher.exe
items count = 50000
How many transactions do you have? : 4

How many items in transaction #1?: 3
enter item #1 : 0
item: bread
enter item #2 : 1
item: coffee
enter item #3 : 2
item: cheese

How many items in transaction #2?: 2
enter item #1 : 0
item: bread
enter item #2 : 2
item: cheese

How many items in transaction #3?: 2
enter item #1 : 0
item: bread
enter item #2 : 3
item: butter

How many items in transaction #4?: 3
enter item #1 : 1
item: coffee
enter item #2 : 5
item: cake
enter item #3 : 4
item: tea

        transaction #1 --> 3
bread
coffee
cheese
        transaction #2 --> 2
bread
cheese
        transaction #3 --> 2
bread
butter
        transaction #4 --> 3
```

Figure 4-23 Applying Affinity Rule Locally #1

```
transaction #4 --> 3
coffee
cake
tea

Enter your choice:
    1. calculate support
    2. calculate confidence
    3. end program

>> 1
how many items? : 2
enter items:
0 2
support = %50
Time taken: 2 milliseconds
Enter your choice:
    1. calculate support
    2. calculate confidence
    3. end program

>> 2
how many items? : 2
enter items:
0 2
enter base item:
0
confidence = %50 / %75 = %66

Time taken: 3 milliseconds
```

Figure 4-24 Applying Affinity Rule Locally #2

4.6 Avalanche Effect

As explained in 3.3.9 in chapter three, the author has calculated the avalanche effect by flipping one bit in one key (first key) of the algorithm used in this thesis, for 506 items/rows, using the equation:

$$\frac{\text{No of flipping bits in cipher text}}{\text{No of bits in cipher text}} \times 100\%$$

With different key length as below:

- The key length shall be dynamic. Its length would be random based on original item length without amplifying, the maximum avalanche effects was 42.57% and the minimum avalanche effect was 3.9%, with total average for 506 items are 14.98%, as shown in figure 4-25 below.

```

Select Microsoft Visual Studio Debug Console
username: Amer
password: $mr08713
-----
processing..
please wait..
items count = 506
-----
Avalanche Effect By Flipping #1 Bit is = %14.9866;-----> 19413 bits have changed from 129536
MAX Avalanche Effect is: %42.5781
MIN Avalanche Effect is: %3.90625

#####
#                                     #
# Time Taken: 159 milliseconds #
#                                     #
#####
Finished.....

```

Figure 4-25 Avalanche Effect of Without Amplifying

- The key length shall be semi dynamic. It would be either 16 bytes or more than 16 bytes. And the resulting cipher will vary from 16 to 32 bytes. The maximum avalanche effect was 44.14% and the minimum avalanche effect was 20.7%, with total average avalanche effect for 506 items are 24.91%, as shown in figure 4-26 below.

```

Microsoft Visual Studio Debug Console
username: Amer
password: $mr08713
-----
processing..
please wait..
items count = 506
-----

Avalanche Effect By Flipping #2 Bit is = %24.9135 :----> 32272 bits have changed from 129536
MAX Avalanche Effect is: %44.1406
MIN Avalanche Effect is: %20.7031

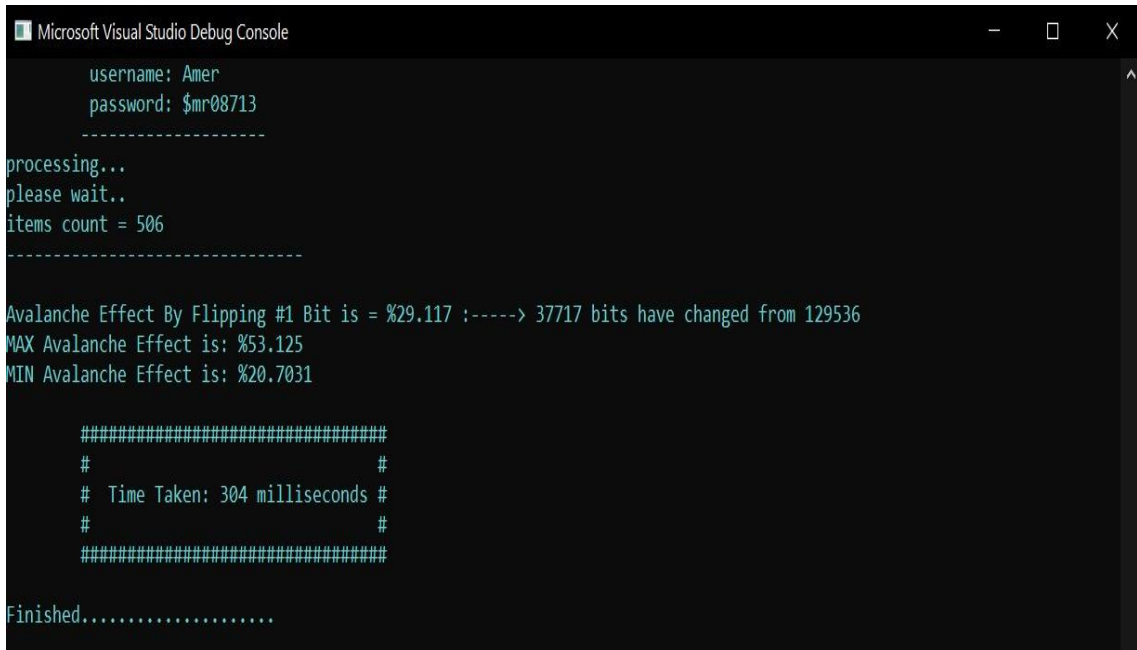
#####
#                                     #
# Time Taken: 170 milliseconds #
#                                     #
#####

Finished.....

```

Figure 4-26 Avalanche Effect with Amplifying from 16 bytes/128 bits → 32 bytes/256 bits

- The key length shall be bipolar. It will be either 16 bytes or 32 bytes. And the resulting cipher will be either 16 or 32 bytes. The maximum avalanche effect was 53.12% and the minimum avalanche effect was 20.7%, with total average avalanche effect for 506 items are 29.11%, as shown in figure 4-27 below.



```

Microsoft Visual Studio Debug Console
username: Amer
password: $mr08713
-----
processing..
please wait..
items count = 506
-----

Avalanche Effect By Flipping #1 Bit is = %29.117 :-----> 37717 bits have changed from 129536
MAX Avalanche Effect is: %53.125
MIN Avalanche Effect is: %20.7031

#####
#                                     #
# Time Taken: 304 milliseconds #
#                                     #
#####

Finished.....

```

Figure 4-27 Avalanche Effect with Amplifying either 16 bytes/128 bits or 32 bytes/256 bits

- The key length shall be constant/fixed. It will be 32 bytes always. And the resulting cipher will be 32 bytes. The maximum avalanche effect was 55.46% and the minimum avalanche effect was 46.48%, with total average avalanche effect for 506 items 50.25%, as shown in figure 4-28 below.

```
Select Microsoft Visual Studio Debug Console
username: Amer
password: $mr08713
-----
processing..
please wait..
items count = 506
-----
Avalanche Effect By Flipping #1 Bit is = %50.2501 ;----> 65092 bits have changed from 129536
MAX Avalanche Effect is: %55.4688
MIN Avalanche Effect is: %46.4844

#####
#                                     #
# Time Taken: 367 milliseconds #
#                                     #
#####

Finished.....
```

Figure 4-28 Avalanche Effect with Amplifying to 32 bytes/256 bits

CHAPTER FIVE: Conclusion and Future Work

5.1 Achievements

In this research, the author has strived to find a new, fast, abstinent easy encryption method to avoid the other heavy and slow encryption methods to encrypt data stored in cloud. Using symmetric encryption algorithm row by row enables processing the encryption, decryption, and affinity rule operations in a flat and easy way.

Despite there is possibility to show this algorithm as a casual one that many could emulate one similar, but the manipulations that the author has performed through encrypt data row by row in large key size was a cornerstone to achieve the goal of this thesis.

Hence the security level was proper in relation of key size (256 bits) and the good average of avalanche effect obtained when change one bit in one key. And the time for encryption-decryption was very good when compared to block cipher encryptions methods.

Note that the avalanche effect was getting better results when enlarges the key size. With 256 bits key size has got the better results than smaller key sizes.

Also, the author has used two symmetric keys algorithm which means encrypted the data twice rather than once.

Row by row/item by item encryption is a must to conduct some important mining algorithms like one used in the thesis. Switching the state of the cipher to be other than row by row cipher would represent an obstacle in maintaining mining operations on cipher texts.

My algorithm could encrypt-decrypt any kind of characters like separator, full stop, space, bracket, and semicolon....etc.).

Using same seed for a specific word would; somehow; lower the level of security but this is inevitable here since needs to have my affinity rule applied on the cipher text. In another word, using different functions to generate the random keys result to have different ciphers for the same text/word and that will lead to a real chaos that would finally lead to lose the basic aim of this encryption technique.

Recently, saving data in cloud became familiar but it has its own disadvantage regarding letting a third party/CSP to disclose the important information the owner uploads. This encryption shall preserve the integrity of information before transmitting to the cloud, this will prevent any third party from catching a glance on data in its three states (data at rest, data in motion and data in use). In addition to hide the information of the owner once the cloud become under cyber-attack.

5.2 Drawbacks

When compare the two file sizes of plaintext vs generated ciphertext from same plaintext, the file size of the ciphertext has a double size of the plaintext file the, and that is reasonable when using hexadecimal characters. Since, each character in plain text would be two characters in cipher.

Symmetric key encryption is notorious when the data owner faces one of his staff of employees leaves his job so the data owner must change the symmetric key used before.

Based on avalanche effect results, my algorithm is not effective unless using key size of 256 bits.

5.3 In General

Author assumes that all the results were shown here are widely appropriate when speaking regarding security, speed, and agility of execution altogether. removing one option of (security, speed, agility of execution) might downgrade this study and make it fragile to be criticized specially when comparing to other studies of the same goals. But once looking at this study as a method to apply security with some mining analytical operations shall lift the hand of this study higher than other studies.

In this thesis the author has answered all the questions of the research, as follow:

- regarding Q1, the answers are in in chapter four/4.5 “Association Rule”. Hence, could get the same result if applied the same rules at plaintext.
- Q2 answers are in chapter three/3.3.4 “Key Generation”, 3.3.4.1 “Key 1 Generation” and 3.3.4.2 “Key 2 Generation”. Hence, could explain the functions used in generating the two keys in a pseudo random section for each key. And how could use the first character of username as a seed for the function to generate key1 and first character of password as a seed for the function to generate key2.
- Q3 answer is that the algorithm has shown a 100% of morphing the plaintext to another totally different cipher, and that leads to have mining results totally different than the mining results when apply same mining algorithm at the plaintext. And no one could guess what the real extracted encrypted texts are unless after decrypted. Especially, the author hasn’t encrypted the numbers of the affinity rule, rather he has encrypted the texts contained in this rule.
- Q4 answer is the decryption process of the ciphertexts, and result of the mining process gave me the exact plaintext before encryption process with same locations and index numbers of each item.

- Q5 answers are in chapter three/ 3.3.9 “Avalanche Effect” and chapter four/4.10 “Avalanche Effect”.

5.4 Future Work

The belief of there is no doubt “no study shall be reach completeness” is not an exemption here. Since the author believes this study represents a concept that needs to be enriched more and more. So, the first future work recommends is to apply this algorithm or other similar algorithms on huge size of data to show the speed once data accumulated.

The author may not recommend sophisticating the encryption algorithm while it provides a proper level of security, but trying other algorithms are likable when speaking about security. Trying other association rule methods using same encryption algorithm are a must in future.

References

- (PDF) *A Study on Indexes and Index Structures*. (n.d.). Retrieved January 6, 2021, from https://www.researchgate.net/publication/333843847_A_Study_on_Indexes_and_Index_Structures
- 10View of Security in Data Mining- A Comprehensive Survey*. (n.d.).
- Agrawal, R., & S&ant, R. (n.d.). *Fast Algorithms for Mining Association Rules*.
- Aljawarneh, S., Yassein, M. B., & Talafha, W. A. (2017). A resource-efficient encryption algorithm for multimedia big data. *Multimedia Tools and Applications*, 76(21), 22703–22724. <https://doi.org/10.1007/s11042-016-4333-y>
- Ashraf, I. (2014). *International Journal of Advance Research in Data Mining Algorithms and their applications in Education Data Mining*. October.
- Baek, J., Vu, Q. H., Liu, J. K., Huang, X., & Xiang, Y. (2015). A secure cloud computing based framework for big data information management of smart grid. *IEEE Transactions on Cloud Computing*, 3(2), 233–244. <https://doi.org/10.1109/TCC.2014.2359460>
- Barsalou, L. (1992). Frames, Concepts, and Conceptual Fields. In *Frames, Fields, and Contrasts* (pp. 21–74).
- Category 8 // Encryption*. (2012).
- Chaudhary, P., & Gulati, N. (2016). Security in Data Mining. *International Journal of Engineering Science and Computing*, 4604. <https://doi.org/10.4010/2016.1148>
- Cryptography and Computer Privacy*. (n.d.). Retrieved July 24, 2021, from <https://www.apprendre-en-ligne.net/crypto/bibliotheque/feistel/index.html>
- Data Mining: Concepts and Techniques - 3rd Edition*. (n.d.). Retrieved December 23, 2020, from <https://www.elsevier.com/books/data-mining-concepts-and-techniques/han/978-0-12-381479-1>
- Dawood, O. A., Sagheer, A. M., & Al-Rawi, S. S. (2019). Design large symmetric algorithm for securing big data. *Proceedings - International Conference on Developments in ESystems Engineering, DeSE, 2018-Sept*(March 2019), 123–128. <https://doi.org/10.1109/DeSE.2018.00026>

Encryption Techniques for Big Data in a Cloud. (n.d.).

Farhan Bashir Shaikh and S. Haider, "Security threats in cloud computing," 2011. (n.d.).

Farhan Bashir Shaikh and S. Haider, "Security Threats in Cloud Computing," 2011 International Conference for Internet Technology and Secured Transactions, Abu Dhabi, 2011, Pp. 214-219. Retrieved January 3, 2021, from <https://ieeexplore.ieee.org/abstract/document/6148380>

Gentry, C. (2009). *A FULLY HOMOMORPHIC ENCRYPTION SCHEME A DISSERTATION SUBMITTED TO THE DEPARTMENT OF COMPUTER SCIENCE AND THE COMMITTEE ON GRADUATE STUDIES OF STANFORD UNIVERSITY IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY.*

George, G., Haas, M. R., & Pentland, A. (2014). From the editors: Big data and management. In *Academy of Management Journal* (Vol. 57, Issue 2, pp. 321–326). Academy of Management. <https://doi.org/10.5465/amj.2014.4002>

Gupta, M., & George, J. F. (2016). Toward the development of a big data analytics capability. *Information and Management*, 53(8), 1049–1064. <https://doi.org/10.1016/j.im.2016.07.004>

Han, J., Pei, J., & Yin, Y. (n.d.). *Mining Frequent P patterns without Candidate Generation.*

Hossain, M. A., Ullah, A., Khan, N. I., & Alam, M. F. (2019). Design and Development of a Novel Symmetric Algorithm for Enhancing Data Security in Cloud Computing. *Journal of Information Security*, 10(04), 199–236. <https://doi.org/10.4236/jis.2019.104012>

How much data is generated each day? | World Economic Forum. (n.d.). Retrieved June 19, 2021, from <https://www.weforum.org/agenda/2019/04/how-much-data-is-generated-each-day-cf4bddf29f/>

Hussein Saeed, E. M., & Hussain, R. M. (2019). Encryption of Association Rules Using Modified Dynamic Mapping and Modified (AES) Algorithm. *1st International Scientific Conference of Computer and Applied Sciences, CAS 2019*, 204–209. <https://doi.org/10.1109/CAS47993.2019.9075701>

Jilke, H. (n.d.). *OPUS: An Efficient Admissible Algorithm for Unordered Search.*

K., R., & K., M. (2017). e-Governance using Data Warehousing and Data Mining. *International Journal of Computer Applications*, 169(8), 28–31. <https://doi.org/10.5120/ijca2017914785>

- Karimunnisa, S., & Kompalli, V. S. (2019). Cloud computing: Review on recent research progress and issues. *International Journal of Advanced Trends in Computer Science and Engineering*, 8(2), 216–223. <https://doi.org/10.30534/ijatcse/2019/18822019>
- Kumar, V., & Sharma, S. (n.d.). *Cryptompress: A Symmetric Cryptography algorithm to deny Bruteforce Attack*.
- Ladekar, S. (2014). Best Practices for Information Security Breach Management. *ResearchGate*, July, 0–44.
- Lai, J., Deng, R. H., Pang, H., & Weng, J. (2014). LNCS 8712 - Verifiable Computation on Outsourced Encrypted Data. In *LNCS* (Vol. 8712). Springer International Publishing Switzerland.
- Larose, D. T., & Larose, C. D. (2014). *Discovering Knowledge in Data: An Introduction to Data Mining*. www.wiley.com.
- LASKARI, E. C., MELETIOU, G. C., TASOULIS, D. K., & VRAHATIS, M. N. (2003). *Data Mining and Cryptology*. *Iccmse*, 346–349. https://doi.org/10.1142/9789812704658_0078
- Mikalef, P., Krogstie, J., & Pavlou, P. (n.d.). *Understanding and Designing Trust in Information Systems View project SOCRATIC-Social Creative Intelligence Platform for achieving Global Sustainability Goals View project*.
<https://www.researchgate.net/publication/337543997>
- Monshizadeh, M., & Yan, Z. (2014). Security related data mining. *Proceedings - 2014 IEEE International Conference on Computer and Information Technology, CIT 2014*, 775–782. <https://doi.org/10.1109/CIT.2014.130>
- Niranjan, B. A., Deepa Shenoy, P., R, V. K., α, N. A., σ, N. A., Deepa Shenoy ρ, P., & R GΩ, V. K. (2016). *Security in Data Mining-A Comprehensive Survey Security in Data Mining-A Comprehensive Survey Security in Data Mining-A Comprehensive Survey*.
<https://www.researchgate.net/publication/330213114>
- Qi, X., & Zong, M. (2012). An Overview of Privacy Preserving Data Mining. *Procedia Environmental Sciences*, 12(Icse 2011), 1341–1347. <https://doi.org/10.1016/j.proenv.2012.01.432>
- Rashid, A., & Chaturvedi, A. (2019). Cloud Computing Characteristics and Services A Brief Review. *International Journal of Computer Sciences and Engineering*, 7(2), 421–426.

<https://doi.org/10.26438/ijcse/v7i2.421426>

Regulating the internet giants - The world's most valuable resource is no longer oil, but data | Leaders | The Economist. (n.d.). Retrieved December 18, 2020, from <https://www.economist.com/leaders/2017/05/06/the-worlds-most-valuable-resource-is-no-longer-oil-but-data>

Salam, M. I., Yau, W. C., Chin, J. J., Heng, S. H., Ling, H. C., Phan, R. C. W., Poh, G. Sen, Tan, S. Y., & Yap, W. S. (2015). Implementation of searchable symmetric encryption for privacy-preserving keyword search on cloud storage. *Human-Centric Computing and Information Sciences*, 5(1). <https://doi.org/10.1186/s13673-015-0039-9>

Samanthula, B. K., Albehairi, S., & Dong, B. (2019). A privacy-preserving framework for collaborative association rule mining in cloud. *Proceedings - 2019 3rd IEEE International Conference on Cloud and Fog Computing Technologies and Applications, Cloud Summit 2019*, 116–121. <https://doi.org/10.1109/CloudSummit47114.2019.00025>

Song, D. X., Wagner, D., & Perrig, A. (n.d.). *Practical Techniques for Searches on Encrypted Data.*

Sugumar, R., & Imam, S. B. S. (2015). Symmetric encryption algorithm to secure outsourced data in public cloud storage. *Indian Journal of Science and Technology*, 8(23). <https://doi.org/10.17485/ijst/2015/v8i23/79210>

Tian, Y. (2017). Towards the Development of Best Data Security for Big Data. *Communications and Network*, 09(04), 291–301. <https://doi.org/10.4236/cn.2017.94020>

Vashi, D., Bhadka, H. B., Patel, K., & Garg, S. (2019). Implementation of Attribute Based Symmetric Encryption through Vertically Partitioned Data in PPDM. *International Journal of Engineering and Advanced Technology*, 9(1), 868–874. <https://doi.org/10.35940/ijeat.a9395.109119>

Wang, C., Ren, K., & Wang, J. (2011). Secure and practical outsourcing of linear programming in cloud computing. *Proceedings - IEEE INFOCOM*, 820–828. <https://doi.org/10.1109/INFCOM.2011.5935305>

Wang, W. Y. C., Rashid, A., & Chuang, H. M. (2011). Toward the trend of cloud computing. *Journal of Electronic Commerce Research*, 12(4), 238–242.

What kind of servers do you use? what are the specs :P - Replit. (n.d.). Retrieved July 15, 2021,

from <https://replit.com/talk/ask/What-kind-of-servers-do-you-use-what-are-the-specs-P/11#149>

- Xu, L., Jiang, C., Wang, J., Yuan, J., & Ren, Y. (2014). Information security in big data: Privacy and data mining. *IEEE Access*, 2, 1149–1176.
<https://doi.org/10.1109/ACCESS.2014.2362522>
- Yakoubov, S., Gadepally, V., Schear, N., Shen, E., & Yerukhimovich, A. (2014). A survey of cryptographic approaches to securing big-data analytics in the cloud. *2014 IEEE High Performance Extreme Computing Conference, HPEC 2014*.
<https://doi.org/10.1109/HPEC.2014.7040943>
- Yildirim, E. (2016). The importance of information security awareness for the success of business enterprises. *Advances in Intelligent Systems and Computing*, 501(May), 211–222.
https://doi.org/10.1007/978-3-319-41932-9_17
- Youssef, A. (2016). Cloud Service Providers : A Comparative Study. *International Journal of Computer Applications & Information Technology*, 5(May 2014), 46–51. www.ijcait.com
- Zaki, M J, Parthasarathy, S., Ogihara, M., & Li, W. (1997). *New Algorithms for Fast Discovery of Association Rules* *. www.aaai.org
- Zaki, Mohammed J. (2000). Scalable algorithms for association mining. *IEEE Transactions on Knowledge and Data Engineering*, 12(3), 372–390. <https://doi.org/10.1109/69.846291>
- Zaki, Mohammed J, Parthasarathy, S., Li, W., Stolorz, P., & Musick, R. (1997). Parallel Algorithms for Discovery of Association Rules. In *Data Mining and Knowledge Discovery* (Vol. 1). Kluwer Academic Publishers.
- Zhu, X., & Wu, X. (2004). Class Noise vs. Attribute Noise: A Quantitative Study. *Artificial Intelligence Review*, 22(3), 177–210. <https://doi.org/10.1007/s10462-004-0751-8>