The 3rd International Congress on Technology, Communication and Knowledge (ICTCK)

28-29 December 2016
Islamic Azad University – Mashhad Branch

Cooperative Methodology to Generate a New Scheme for Cryptography

Samaher Al-Janabi
Department of Computer Science, Faculty of Science for Women (SCIW)
University of Babylon, Babylon, Iraq
samaher@uobabylon.edu.iq

Ibrahim Al-Shourbaji
Computer Network Department, Computer Science and Information System College
University of Jazan, Saudi Arabia
i_shurbaji@yahoo.com

***Abstract*—In this paper, a novel method named as Frequency Pattern-Knowledge Constructions (FP-KC) is developed. This method attempts to develop Frequency Pattern (FP) Growth data mining algorithm using several knowledge constructions to find the association rules and minimize the shared information (i.e. fined frequent item set), FP-KC combines the criteria of Principal Component Analysis (PCA) with FP-Growth techniques. These criteria include eigenvalues, cumulative variability and scree plot. There are several reasons for developing the FP-Growth data mining algorithm to build up a novel FP-KC technique that can find the association rules, including: (a) the size of an FP-tree is typically smaller than the size of the uncompressed data because many records in a dataset often have a few items (b) to give the best result in the case that all the records have the same set of items; (c) FP-Growth is an efficient algorithm because it illustrates how a compact representation of the transaction dataset helps to efficiently generate frequent item sets; and (d) The run-time performance of FP-Growth depends on the compaction factor of the dataset, while the enhanced algorithm in Subliminal Channel (SC) depends on both the position of a character in the alphabet and its position in the plain rule word (i.e. rules resulting from association rules FP-KC), with a specific function to determine the cipher rule character. To evaluate the efficiency of the proposed method, four case studies were used. Based on the results, the proposed method can be considered as an efficient technique for secure mining of association rules of partitioned data compared with the traditional method.**

Keywords: Data Mining – Subliminal Cryptography – Association Rules –Constructions – Principle Component Analysis.

INTRODUCTION

Cryptography is the art or science encompassing the principles and methods of transforming an intelligible message into one that is unintelligible and then transforming that message back to its original form. The databases of companies, organizations and scientific institutions contain confidential and critical information that needs to be protected against disclosure or use by unauthorized persons [1]. Cryptographic techniques can play a vital role in this process.

An association rules mining technique is proposed whose purpose is to extract association rules and to hide the sensitive rules without violating the confidentiality and privacy of the data [2]. This technique has also been used by several researchers to help enterprises avoid the risk that could occur when sharing the data [3].

Data mining is the analysis of (often large) observational datasets to find unsuspected relationships and to summarize the data in novel ways that are both understandable and useful to the owner of the data. It is not just a single method or technique but rather a spectrum of different approaches that search for hidden patterns and relationships among vast databases. Its use is daily becoming more widespread because

it empowers companies to uncover profitable patterns and trends from their existing databases [4]. For example, this kind of information can be used to assist with identifying business choices in sales and the marketing of new products and to shape future business decisions.

Most tools operate by gathering all data into a central site and then running an algorithm against that data. However, privacy concerns can prevent the building of a centralized warehouse. Data may be distributed among several custodians,

none of which are allowed to transfer their data to another site [5]. The Subliminal Channel (SC) is a covert communication that can be used to send a secret message to an authorized receiver, but the message cannot be discovered by any unauthorized receiver or third party [6].

This paper addresses the problem of computing association mining rules within a scenario. This scenario assumes that all sites have the same schema within the shared databases but that each of them has information on different histories. The main goal of this work is not only to produce global association rules, but also to result in adequately securing the shared information about each site.

The remainder of the paper is organized as follows: Section 2 provides a general overview of SARM, SARM, FP-Growth, SC and PCA strategies: Section 3 presents the proposed FP-KC Algorithm: Section 4 suggests algorithms for SC. Section 5 provides four case studies to test the proposed algorithm effecincy and the paper is concluded in Section 6.

OVERVIEW OF SARM, FP-GROWTH, SC AND PCA

SARM

The association mining rule problem can be defined as follows: Let {i1, i2, … , in} be a set of items. Let DB be a set of transactions, where each transaction T is an utmost, such that T $\subseteq$ I. Given an item set X $\subseteq$ I, a transaction T contains X if and only if X $\subseteq$ T. An association rule is an implication of the form X $\Rightarrow$ Y, where X $\subseteq$ I, Y $\subseteq$ I and X $\cap$ Y = ϕ. The rule X $\Rightarrow$ Y has support s in the transaction database DB if s% of transactions in DB contain X $\cup$ Y. The association rule holds in the transaction database DB with confidence c if c% of transactions in DB that contain X also contain Y. An item set X with k items is called the k-item set. The problem of mining association rules is to find all rules whose support and confidence are higher than a certain user-specified minimum support and confidence.

In this simplified definition of the association rules, missing items, negatives and quantities are not considered. In this respect, the transaction database DB can be seen as a 0/1 matrix where each column is an item and each row is a transaction. In this paper, we use this view of association rules.

Distributed Mining of Association Rules: The above problem of mining association rules can be extended to distributed environments. Let us assume that a transaction database DB is horizontally partitioned among n stories (namely, S1, S2, … , Sn), where DB = DB1 $\cup$ DB2 $\cup$ … $\cup$ DBn and DBi resides at side Si (1 $\leq$ i $\leq$ n). The item set X has local support count of X.supi at site Si if X.supi of the transactions contain X. The global support count of X is given as X.sup = Pn i=1 X.supi. An item set X is globally supported if X.sup $\geq$ s × (Pn i=1 |DBi|). Global confidence of a rule X $\Rightarrow$ Y can be given as {X $\cup$ Y}.sup/X.sup.

The set of large item sets L (k) consists of all k-item sets that are globally supported. The set of locally large item sets LLi (k) consists of all k-item sets supported locally at site Si. GLi (k) = L (k) \ LLi (k) is the set of globally large k-item sets locally supported at site Si.

The aim of distributed association rule mining is to find the sets L (k) for all k > 1 and the support counts for these item sets, and from this compute association rules with the specified minimum support and confidence. A fast algorithm for distributed association rule mining is given by Tan et al. [7]. Their procedure for fast-distributed mining of association rules is summarized below.

Candidate Sets Generation: Generate candidate sets CGi (k) based on GLi (k − 1), i.e. item sets that are supported by the Si at the (k - 1)-th iteration, using the classic Apriori candidate generation algorithm. Each site generates candidates based on the intersection of globally large (k - 1) item sets and locally large (k - 1) item sets.

Local Pruning: For each X 2 CGi (k), scan the database DBi at Si to compute X.supi. If X is locally large at Si, it is included in the LLi (k) set. It is clear that if X is supported globally, it will be supported in one site.

Support Count Exchange: LLi (k) are broadcast, and each site computes the local support for the items in $\cup$ iLLi (k).

Broadcast Mining Results: Each site broadcasts the local support for item sets in [iLLi (k). From this, each site is able to compute L (k).

The real deception of the enemy is the unified deception. Depending on this concept, our encrypting approach is suggested. It converts the plain text to an encrypted form. This means that the enemy cannot sense any encryption operation. When we want to implement subliminal methods, we must have the following [8, 9]:

Natural language dictionaries;

Natural language grammar with its semantics;

Natural language story generation techniques.

Subliminal methods are good ciphering methods of deceiving the enemy. Most importantly, they are suitable for transforming small messages. In order to solve this point, we will attempt to combine both the concept of data mining and

the Subliminal Channel (SC) methods. We can summarize the main challenges as follows: we need to suggest a fast and secure algorithm that has the ability to extract the association rules to encrypt and decrypt the rules through the SC.

FP-Growth Algorithm

FP-Growth is a method for mining frequent sets of items without candidate generation. It constructs a highly compact data structure called an FP-tree (Frequent Pattern tree) to compress the original transaction database, rather than simply employing a generate-and-test strategy like Apriori methods [7]. It is considered as a powerful computational tool in generating association rules compared to the Apriori algorithm. It adopts a divide-and-conquer strategy by compressing the database representing frequent items to an FP-tree that retains all the essential information and by dividing the compressed database into a set of conditional databases, each associated with one frequent item set and mining each of them separately.

FP-Growth is a seminal algorithm proposed in this work for mining frequent item sets for association rules. Figure 1 shows a dataset that contains ten transactions and five items. Each node in the tree contains the label of an item along with a counter that shows the number of transactions mapped onto the given path. Initially, the FP-tree contains only the root node, represented by the null symbol [10].

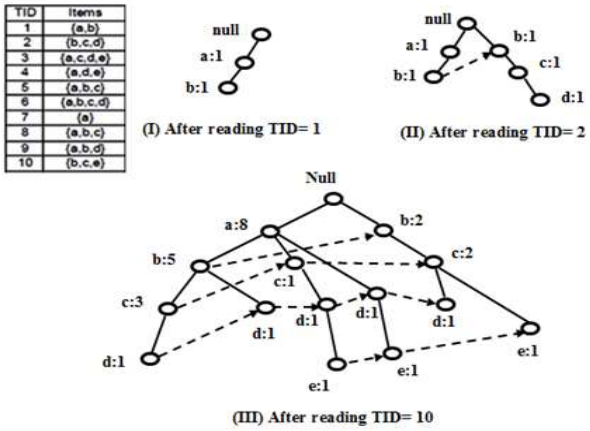

Fig. 1. Construction of an FP-tree

The dataset is scanned once to determine the support count of each item. Infrequent items are discarded, while the frequent items are sorted into decreasing support counts. For the dataset shown in Fig. 1, a is the most frequent item, followed by b, c, d and e.

The algorithm makes a second pass over the data to construct the FP-tree. After reading the first transaction *{a,b}*, the nodes labeled as *a* and *b* are created. A path is then formed from ***null*** —> *a* —> *b* to encode the transaction. Every node along the path has a frequency count of 1.

After reading the second transaction {b, c, d}, a new set of nodes is created for items *b*, *c* and *d.* A path is then formed to represent the transaction by connecting the nodes **null** —> *b* —> *c* —> *d.* Every node along this path also has a frequency count equal to one. Although the first two transactions have an item in common, which is *b*, their paths are disjointed because the transactions do not share a common prefix.

The third transaction *{a,c,d,e}*, shares a common prefix item (which is *a*) with the first transaction. As a result, the path for the third transaction, **null** —> *a* —> *c* —> *d* —> *e,* overlaps with the path for the first transaction, **null** —>*a* —> *b.* Because of their overlapping path, the frequency count for node *a* is incremented to two, while the frequency counts for the newly created nodes, c, *d,* and e, are equal to one. This process continues until every transaction has been mapped onto one of the paths given in the FP-tree.

The main reasons for the proposed FP-KC are as follows:

- The size of an FP-tree is typically smaller than the size of the uncompressed data because many records in dataset often have a few items in common.
- To give good results all the records should have the same set of items.
- FP-Growth is an interesting algorithm because it illustrates how a compact representation of the transaction dataset helps to efficiently generate frequent item sets**.**
- The run-time performance of FP-Growth depends on the compaction factor of the dataset.

As a result, the FP-Growth will find all the frequent item sets ending with a particular suffix by employing a divide-and-conquer strategy to split the problem into smaller sub problems.

SC

Subliminal Channel (SC) is a secure communication channel that cannot be read or detected by those whom it is not intended. The SC was first proposed by Gustavus Simmons in 1984. It is widely used in digital signature applications to transfer secret messages in a normal looking communication.

It is considered as a solution to the prisoners problem; consider two prisoners in separate cells who want to exchange messages, but must do so through the warden who demands full view of the messages (that is, no encryption). It enables the prisoners to exchange secret information through messages that appear to be innocuous. It requires prior agreement on the part of the prisoners. For example, if an odd length word corresponds to ``1'' and an even length word corresponds to ``0'', then the previous sentence contains the subliminal message ``101011010011'. So, a subliminal channel is a way of embedding information in public communication in an undetectable way. Some sort of shared secret (a key, knowledge of what to look for) is needed to reconstruct the subliminal information [11]. Thus SC can be used to send a message over an insecure channel and it cannot be detected by third party or unauthorized receiver.

PCA

Principle Component Analysis (PCA) is one of the most widely used dimension reduction technique, which is often applied to identify patterns in complex data. In PCA, a set of *p* associated variables is transformed into a smaller set of constructs called the principal components (PCs) which is used to discover and interpret the dependences that exist among the variables in a compact manner and to examine the relationships that may exist among them [12]. The PCA can be computed according to the following steps [13, 14]:

- **Step1:** Compute the standardized data matrix **Z** =[Z1, Z2, … , Zm] based on
  $Zi = (Xi - \mu i) / \sigma ii$ from the original dataset.
- **Step2:** Compute the Eigenvalues:
  If **B** is an *m*-by-*m* matrix and **I** is an *m*-by-*m* identity matrix (one's on the main diagonal and zeros elsewhere), then the scalars (numbers of dimension 1 × 1) λ1, λ1, … , λm, are the Eigenvalues of B, if they satisfy $|B - \lambda I| = 0$.
- **Step3:** Compute Eigenvectors:
  Let **B** an *m*-by-*m* matrix and λ to be an eigenvalues of **B**, then nonzero *m*-by-1 vector **e** is an eigenvector of B, if $Be = \lambda e$.
- **Step4:** Compute *ith* principal components:
  The *ith* principal component of the standardized data matrix **Z** = [Z1, Z2, … , Zm] is given by $Yi = \mathbf{e}^{T}i\mathbf{Z}$

where ***ei*** refers to the *ith* eigenvector (discussed below) and $\boldsymbol{e}^{T}i$ refers to the transpose of ***ei***.

The principal components are linear combinations Y1, Y2, … , Yk of the standardized variables in Z such that (1) the variances of the Y*i* are as large as possible and (2) the Y*i* are uncorrelated. The first principal component is the linear combination $Y1 = \mathbf{e}^{T}1Z = e11Z1 + e12Z2 + \cdots + e1m\ Zm$, which has greater variability than any other possible linear combination of the Z variables. Thus,

- The first principal component is the linear combination Y1 = eT1Z, which maximizes Var (Y1) = eT1ρ e1.
- The second principal component is the linear combination Y2 = eT2Z, which is independent of Y1 and maximizes Var (Y2) = eT2ρ e2.
- The *ith* principal component is the linear combination Y*i* = e_iX, which is independent of all the other principal components Y*j* , *j* < *i* , and maximizes var (Y*i*) = eTiρ e*i*.

THE PROPOSED FP-KC ALGORITHM

The primary goal of Dimension Reduction (DR) methods is to use the correlation structure among the predicator variables in order to reduce the three main dimensions. But how to combine these dimensions without a significant loss of information?. This question is still considered as one of the unsolved open problems in Knowledge Discovery in Data Mining (KDD) [15, 16]. The proposed FP-KC algorithm uses a set of steps which are executes as follows:

Algorithm 1: The Pseudo code of FP-KC

Input: *Principle Component (PC) Database, Set of knowledge construction, min-sup*

Output*: set of association rules*

STEP1*: Construct the FP-tree*

Scan the PC Database

Collect F, the set of frequent items and their support counts

Sort F in support count descending order as L, the list of frequent items

Create the root of an FP-tree and label it as "Null"

STEP2*: Test the Knowledge Constriction Conditions*

For *each record in PC database test the KC and* ***do***

If *the record not verification eigenvalue criterion* ***Then***

remove the record from PC database

Else if *the record not verify proportion of variance explained criterion* ***Then***

remove the record from PC database

End if

Else if**, the record not verify the scree plot criterion* ***Then

remove the record from PC database

End if

***Else**, go to Step3*

End if

End For

STEP3: *Built FP-KC*

For each record, verify Knowledge Constriction Conditions

Select and sort the frequent items in records according to the order of L

Call insert-Tree procedure

STEP4: *Mining the FP-KC using FP-Growth procedure*

If** the Tree contains a single path* ***Then

For** each combination of nodes CN in path* ***do

Generate pattern CN U α

Support_count = min_support count of nodes

Else if** the tree contains multi paths* ***Then

***For** each nodei in the header of tree*

Generate pattern CN = nodei U α

Support_count = nodei _support count of nodes

Construct CN condition pattern then CN conditional FP-TreeCN

End if

Else If** TreeCN < > ϕ* ***Then

Go to step 4.

End if

End For

End If

***STEP5:** End FP-KC Algorithm*

Procedure for Inserting-Tree

Step1:

If the Tree has N childs Then
Set i = i +1
Else, Create New Node (N)
i =1; N.link = Tree
End if

Step2:

If remind list < > ϕ Then
For j =1 to No. of element do
Call insert-tree (P, N)
INCREAMENT j
End For
End If

SUGGEST ALGORITHMS FOR SC

Counter Algorithm

In this algorithm, a counter is used to cipher the plain rule, when the counter is equal to a special value (that we determine), then we initialize it. In order to cipher a rule, we deal with this rule as a collection of characters, separate each of them from the rule and take its code and then the following algorithm is implemented to encrypt the rule.

Counter Algorithm

Initialize the (Max) Counter Value
Counter = 1
Repeat
Cipher Rule = Plain Rule + Counter
Counter = Counter + 1
If Counter > Max then Counter = 1
Until No Plain Rule

In order to decrept the ciphered rule, the following algorithm is applied.

Initialize the (Max) Counter Value
Counter = 1
Repeat
Plain Text = Cipher Text - Counter
Counter = Counter + 1
If Counter > Max then Counter = 1
Until No Cipher Text

The Polygram Separation Algorithm

This algorithm separates the characters of a message using separation mapping method and generates a meaningful ciphered text from the new order of characters. This will results in enhancing the e security. The method of separation mapping depends on taking two characters from the plain rule word; send one of them to the first and another to the last to

create a meaningful ciphered rule. The example below illustrates the polygram separation mapping method.

C I P H E R T E X T

C P E T X T E R H I

In order to construct the plain rule, we take the first character from two words, one from the beginning and another from the end of the ciphered rule as shown in the example below:

C P E T X T E R H I

C I P H E R T E X T

Polygram Separation Algorithm

This algorithm depends on both the position of a character in the alphabet and its position in the plain rule word. The modulo operation (mod 3) is applied to find the ciphered characters depend on specific functions that show below. The following algorithm shows the encryption steps.

Repeat

X = characters order in alphabet + character order in the plain rule word.

Y = X mod 3

$$cipher\ character = \begin{cases} plain\ rule\ if & Y=0 \\ right\ \ of\ plain\ rule\ if & Y=1 \\ left\ \ of\ plain\ rule\ if & Y=2 \end{cases} \quad (1)$$

Until No Plain Rule

While the following algorithm is applied to extract the plain rule.

Repeat

X = characters order in alphabet + character order in the plain rule word.

Y = X mod 3

$$plain\ character = \begin{cases} plain\ rule\ if & Y=0 \\ right\ \ of\ plain\ rule\ if & Y=1 \\ left\ \ of\ plain\ rule\ if & Y=2 \end{cases} \quad (2)$$

Dictionary Method

This approach of encryption is easy to implement and use because it does not depend on the orders of characters in the alphabet or word, also it does not need specific values such as a counter to cipher the plain rule. The main idea from this method is to give each plain rule word a synonym in the cipher rule to ensure the meaning of the ciphered rule word does not exactly match the meaning of the plain rule and the encryption method is more effective and secure.

CASE STUDIES

To test the performance of the above suggested methods in section 5 that are built based on the use of data mining and subliminal cryptography techniques, some plain rules with different characters are used as examples to test the proposed algorithm behaviors. Before running the system for each algorithm, some important information should be collected by the system. This information can be used when some parameters need to be entered to the system. Those algorithms will be evaluated by the proposed FP-KC method and based on their performance; it will select the best one. In order to achieve this goal, four cases studies were used to test the efficiency of the proposed method and their results are explained as follows:

Case Study 1

The Plain Rule = see ^ you →soon, confidence =95%

Cipher:

the ciphering process executes according to following steps:

- Initialize the counter = 1.
- Determine the maximum counter value, let it equal to 10.
- Repeat the following process for all the characters in the Rule:

Take the next character = s, code (s) = 19.
Cipher = code (s) + 1 = 19 + 1 = 20,
Code (20) = t, the word generator = take.
Increment counter = 2, 2 <= 10.
Take the next character = e, code (e) = 5.

Cipher = code (e) + 2 = 5 + 2 = 7.
Code (7) = g, the word generator = guitar.
Increment counter = 3, 3 <= 10.
Take the next character = e, code (e) = 5.
Cipher = code (e) + 3 = 5 + 3 = 8.
Code (8) = h, the word generator = home.
Increment counter = 4, 4 <= 10.
Take the next character = y, code (y) = 25.
Cipher = code (y) + 4 = 25 + 4 = 29 *mod* 26 = 3.
Code (3) = c, the word generator = care.
Increment counter = 5, 5 <= 10.
Take the next character = o, code (o) = 15.
Cipher = code (o) + 5 = 15 + 5 = 20.
Code (20) = t, the word generator = this.
Increment counter = 6, 6 <= 10.
Take the next character = u, code (u) = 21.
Cipher = code (u) + 6 = 21 + 6 = 27 *mod* 26 = 1.
Code (1) = a, the word generator = and.
Increment counter = 7, 7 <= 10.
Take the next character = s, code (s) = 19.
Cipher = code (s) + 7 = 19 + 7 = 26.
Code (26) = z, the word generator = zoom.
Increment counter = 8, 8 <= 10.
Take the next character = o, code (o) = 15.
Cipher = code (o) + 8 = 15 + 8 = 23.
Code (23) = w, the word generator = write.
Increment counter = 9, 9 <= 10.
Take the next character = o, code (o) = 15.
Cipher = code (o) + 9 = 15 + 9 = 24.
Code (24) = x, the word generator = x.
Increment counter =10, 10 <= 10.
Take the next character = n, code (n) = 14.
Cipher = code (n) + 10 = 14 + 10 = 24.
Code (24) = x, the word generator = x.

The Cipher Rule = *take* ***guitar home and care this and zoom write x x.***

Decipher

The deciphering process executes according to the following steps:

Initialize the counter = 1.

Determine the maximum counter value; let it equal (10).

Repeat the following process for all words in the cipher text:

Take the next word = take, code (t) = 20, 20 > 1.
Plain = code (t) - 1 = 20 - 1 = 19, Code (19) = s.
Increment counter = 2, 2 <= 10.
Take the next word = guitar, code (g) = 7, 7 > 2.
Plain = code (g) - 2 = 7 - 2 = 5, Code (5) = e.
Increment counter = 3, 3 <= 10.
Take the next word = home, code (h) = 8, 8 > 3.
Plain = code (h) - 3 = 8 - 3 = 5, Code (5) = e.
Increment counter = 4, 4 <= 10.
Take the next word = care, code (c) = 3, 3 > 4, False.
Plain = (3 - 4) + 26 =, Code (25) = y.
Increment counter = 5, 5 <= 10.
Take the next word = this, code (t) = 20, 20 > 5.
Plain = 20 - 5 = 15, Code (15) = o.
Increment counter = 6, 6 <= 10.
Take the next word = and, code (a) = 1, 1 > 6, False.
Plain = (1 – 6) + 26 = 21, Code (21) = u.
Increment counter = 7, 7 <= 10.
Take the next word = zoom, code (z) = 26, 26 > 7.
Plain = 26 - 7 = 19, Code (19) = s.
Increment counter = 8, 8 <= 10.
Take the next word = write, code (w) = 23, 23 > 8.
Plain = 23 - 8 = 15, Code (15) = o.
Increment counter = 9, 9 <= 10.
Take the next word = x, code (x) = 24, 24 > 9.
Plain = 24 - 9 = 15, Code (15) = o.
Increment counter = 10, 10 <= 10.
Take the next word = x, code (x) = 24, 24 > 10.
Plain = 24 - 10 = 14, code (14) = n.

The Plain Rule = see you soon

Case Study 2

If we have the plain rule = see ^ you →soon, confidence = 95%

A. *Cipher*

Cipher Rule = second edition or season or no one until your email

B. *Decipher*

The Plain Text = see you soon

Case Study 3

If we have the plain *r*ule = see ^ you →soon, confidence =95%

Cipher

s in alphabetic = 18, s in plain = 1, X = 18 + 1 =19 mod 3 =1, plain = t (Right)
e in alphabetic = 4, e in plain = 2, X = 4 + 2 =6 mod 3 =0, plain = e (Itself)
e in alphabetic = 4, e in plain = 3, X = 4 +3 =7 mod 3 =1, plain = f (Left)

y in alphabetic = 24, y in plain = 1, X = 24 +1 =25 mod 3 =1, plain = z (Right)
o in alphabetic = 14, o in plain = 2, X = 14 +2 =16 mod 3 =1, plain = p (Right)
u in alphabetic = 20, u in plain = 3, X = 20 +3 =23 mod 3 =2, plain = t (Left)
s in alphabetic = 18, s in plain = 1, X = 18 +1 =19 mod 3 =1, plain = t (Right)
o in alphabetic = 14, o in plain = 2, X = 14 +2 =16 mod 3 =1, plain = p (Right)
o in alphabetic = 14, o in plain = 3, X = 14 +3 =17 mod 3 =2, plain = n (Left)
n in alphabetic = 13, n in plain = 4, X = 13 +4 =17 mod 3 =2, plain = m (Left)

Cipher Rule = take east feature zoo park taxi this public network media

Decipher

t in alphabetic = 19, s in plain = 1, X = 19 + 1 =20 mod 3 =2, plain = s (Left)
e in alphabetic = 4, e in plain = 2, X = 4 + 2 =6 mod 3 =0, plain = e (Itself)
f in alphabetic = 5, e in plain = 3, X = 5 +3 =8 mod 3 =2, plain = e (Left)
z in alphabetic = 25, y in plain = 1, X = 25 +1 =26 mod 3 =2, plain = y (Left)
p in alphabetic = 15, o in plain = 2, X = 15 +2 =17 mod 3 =2, plain = o (Left)
t in alphabetic = 19, u in plain = 3, X = 19 +3 =22 mod 3 =1, plain = u (Right)
t in alphabetic = 19, s in plain = 1, X = 19 +1 =20 mod 3 =2, plain = s (Left)
p in alphabetic = 15, o in plain = 2, X = 15 +2 =17 mod 3 =2, plain = o (Left)
n in alphabetic = 13, o in plain = 3, X = 13 +3 =16 mod 3 =1, plain = o (Right)
m in alphabetic = 12, n in plain = 4, X = 12 +4 =16 mod 3 =1, plain = n (Right)

Plain Rule = see you soon

Case Study 4

Plain Rules:
Neural network ^ knowledge discovery ⇒ data mining; confidence =90%
Data mining ^ knowledge discovery ⇒ neural network; confidence =84%
Knowledge discovery ⇒ neural network ^ data mining; confidence =78%

If we have the first Plain Rule= neural network and knowledge discovery lead to data mining

Cipher

N in alphabetic =14, N in plain rules =1, X=14+1=15 Mod 3=0, Plain=N (itself), the generating word= New
N in alphabetic =14, N in plain rules =2, X=14+2=16 Mod 3=1, Plain=O (right), the generating word= objective.
A in alphabetic =1, A in plain rules =3, X=1+3=4 Mod 3=1, Plain=B (right), the generating word= books.
K in alphabetic =11, K in plain rules =4, X=11+4=15 Mod 3=0, Plain=K (itself), the generating word=know.
D in alphabetic =4, D in plain rules =5, X=4+5=9 Mod 3=0, Plain=D (itself), the generating word=distributed.
L in alphabetic =12, L in plain rules =6, X=12+6=18 Mod 3=0, Plain=L (Left), the generating word=lastly.
T in alphabetic =20, T in plain rules =7, X=20+7=27 Mod 3=0, Plain=T (itself), the generating word= to.
D in alphabetic =4, D in plain rules =8, X=4+8=12 Mod 3=0, Plain=D (itself), the generating word= developing
M in alphabetic =13, M in plain rules =9, X=13+9=22 Mod 3=1, Plain=N (right), the generating word= network.

Cipher Rules = News: Obtrusion Black Khan Destroy Land Tome Doluar Nichol.

Decipher

We will take only the first character from each word and perform the following steps:
N in alphabetic=14, N in plain=1, X=14+1=15 mod 3=0, plain =N (itself)
O in alphabetic=15, O in plain=2, X=15+2=15 mod 3=2, plain =N (left)
B in alphabetic=2, B in plain=3, X=2+3=5 mod 3=2, plain =A (left)
K in alphabetic=11, K in plain=4, X=11+4=15 mod 3=0, plain =K (itself)
D in alphabetic=4, D in plain=5, X=4+5=9 mod 3=0, plain =D (itself)
L in alphabetic=12, L in plain=6, X=12+6=18 mod 3=0, plain =L (itself)
T in alphabetic=20, T in plain=7, X=20+7=27 mod 3=0, plain =T (itself)
D in alphabetic=4, D in plain=4, X=4+8=12 mod 3=0, plain =D (itself)
N in alphabetic=14, M in plain=9, X=14+9=23 mod 3=2, plain =M (left)

CONCLUSION

The discovery of frequent patterns, associations and correlation relationships among huge amounts of data are important and useful in many applications.

In this paper, a novel method is proposed, namely, Frequency Pattern-Knowledge Constructions (FP-KC). The criteria of PCA with FP-Growth techniques were combined to find the association rules and minimize the shared information (i.e. fined frequent item set). The proposed method depends on both the position of a character in the alphabet and its position in the plain rule word, with specific functions to determine the position of the characters in the ciphered rule word. To test the precision of the FP-KC method, four case studies were used. Based on the results, the proposed method can be considered as an efficient technique for secure mining of association rules of partitioned data compared with the traditional method.

REFERENCES

Al-Janabi, S., & Al-Shourbaji, I. (2016) A Study of Cyber Security Awareness in Educational Environment in the Middle East. Journal of Information & Knowledge Management, 15(01), 1650007.

Jain, D, Khatri, P, Soni, R, Chaurasia, B. K (2012) Hiding sensitive association rules without altering the support of sensitive item (s). In: 2th international conference on Advances in Computer Science and Information Technology. Networks and Communications, pp. 500-509

Lou, H, Ma, Y, Zhang, F, Liu, M, Shen, W (2014) Data mining for privacy preserving association rules based on improved MASK algorithm. In: 18 th International Conference onComputer Supported Cooperative Work in Design (CSCWD), pp. 265-270

Al-Janabi, S., Patel, A., Fatlawi, H., Kalajdzic, K., & Al Shourbaji, I. (2014) Empirical rapid and accurate prediction model for data mining tasks in cloud computing environments. In: International Congress on Technology, Communication and Knowledge (ICTCK), pp. 1-8.

Ali, S. H (2013) Novel Approach for Generating the Key of Stream Cipher System Using Random Forest Data Mining Algorithm. In: 6th International Conference on Developments in eSystems Engineering (DeSE), pp. 259-269

Zhenjie H, Dan C, Yumin W (2005) Multi-Signature with Anonymous Threshold Subliminal Cannel for Ad-Hoc Environments. In: 19th AINAInternational Conference on Advanced Information Network and Application , pp. 67-71

Tan P.N, Steinbach. M, Kumar V. (2006) Introduction to Data Mining. University of Minnesota, USA

Kimia A. A, Savova G, Landschaft A, Harper M. B (2015) An Introduction to Natural Language Processing: How You Can Get More From Those Electronic Notes You Are Generating. *Pediatric emergency* care, **31**:7 doi: 10.1097/PEC.0000000000000484

Shapiro S. C (1992) Encyclopedia of Artifical Intellegence. John Willey & Sons, New York, USA

Rajaraman, A, Ullman, J. D (2012) Mining of massive datasets. Cambridge University Press, UK

Moskowitz S. A (2013) U.S. Patent No. 8,612,765. Patent and Trademark Office,Washington, USA

Zhao L, Kang H. S, Kim S.R (2013) Improved Clustering for Intrusion Detection by Principal Component Analysis with Effective Noise Reduction. Information and communication Technology. In: International Conference on information and communication technology, pp. 490-495.

Kantardzi, M. (2011) Data mining: concepts, models, methods, and algorithms. John Wiley & Sons, New Jersey, USA

Burges C. J (2010) Geometric methods for feature extraction and dimensional reduction-a guided tour. Data mining and knowledge discovery handbook. Springer, US

Maldonado S, Weber R, Famili F (2014) Feature selection for high-dimensional class-imbalanced data sets using Support Vector Machines. Information Sciences **286**: 228-246 doi:10.1016/j.ins.2014.07.015

Ali S. H, (2012) A novel tool (FP-KC) for handle the three main dimensions reduction and association rule mining. In: 6th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT), pp. 951-961