Association Rule Mining with Security Based on Playfair Cipher Technique

P. Jagannadha Varma, Amruthaseshadri,.M. Priyanka, M.Ajay Kumar, B.L.Bharadwaj Varma.

Department of Computer Science and Engineering
Lendi Institute of Engineering and Technology, Jonnada, Vizianagram.
.

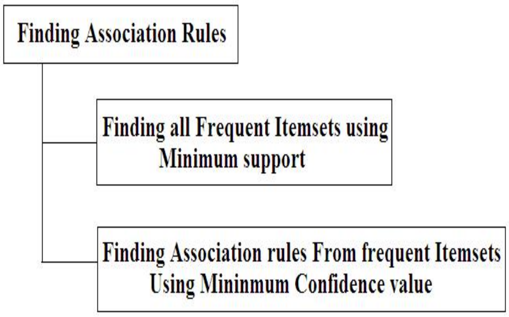
Abstract: Due to the vast increase in data in the present scenario, data mining tasks and methods are in usage. The security attacks are so common day by day over the web. Association rule mining (ARM) is one of the popular data mining methods that discover interesting correlations amongst a large collection of data, which appears incomprehensible. It is an important data mining model studied extensively by the database and data mining community. Association rules are widely used in various areas such as telecommunication networks, market and risk management, and inventory Control etc. Apriori algorithm is one of the techniques for generating frequent itemsets. In order to provide security for obtained frequent itemset ,we are implementing a new technique named playfair cipher. In this technique, we use alphabets for providing secure transmission of message. Since it will be in unreadable form.

Key Words:- Apriori Algorithm, Playfair cipher algorithm, Association Rule Mining.

1.INTRODUCTION :

Generating interesting association rules by using "Apriori algorithm". Transmission of the generated rule by using "Playfair cipher technique". Appriori algorithm is innovative way to find association rules on large scale, allowingimplication outcomes that consist of more than one item based on minimum support threshold.

Playfair cipher technique encrypts pairs of letters (digraphs), instead of single letters as in the simple substitution cipher. The Playfair is significantly harder to break since the frequency analysis used for simple substitution ciphers does not work with it.


Finding Association Rules
Finding all Frequent Itemsets using Minimum support
Finding Association rules From frequent Itemsets Using Mininmum Confidence value

2. ASSOCIATION RULES:

Association rule are the statements that find the relationship between data in any database. For example take a scenariothat you have entered into a hardware shop to buy a cabinet. Then usually we buy peripheral devices like Web cam, Headphones, Stereo set. If shop keeper cracks this set of complimentary goods he will enjoy profit by offering the attractive discount on them, which can be generally termed as combo pack.

Association rule has two parts Antecedent and Consequent. For example, "{cabinet} => {stereo set}". Here cabinet is the antecedent and stereo set is the consequent. Antecedent is the item that found in database, and consequent is the item that found in combination with the first. Association rules are generated during searching for frequent patterns .

The problem of finding association rules is divided into two sub problems: first is to find frequent item sets and second is to find association rules from these item sets.For important relationships association rule uses the criteria of Support‟ and „Confidence‟ that are explained below:

2.1. Apriori Algorithm

Apriori is the Latin word and its meaning is „from what comes before‟. Apriori uses bottom up strategy. It is the most famous and classical algorithm for mining frequent patterns. This algorithm works on categorical attributes.

FREQUENT ITEMSETS GENERATION OF THE APRIORI ALGORITHM

```
Step 1: K=1
Step 2: Fk={i/i  € I ^ σ ({i}) ≥ N* minsupl} {Find a;; the frequent 1-itemsets}
Step 3: repeat
Step 4: k=k+1
Step 5: Ck=apriori-gen(Fk-1)
Step 6: For each transaction t€ T do
Step 7: Ct=subset(Ck,t)
Step 8: For each candidate itemset  C € Ct do
Step 9: σ(c)=σ©+1
Step 10: End for
Step 11: End for
Step 12: Fk={C /C € Ck ^ σ (c) ≥ N* Minsup} * (extract the frequent K-itemset)
Step 13: Untill Fk= Ø
Step 14: Result= UFk
```

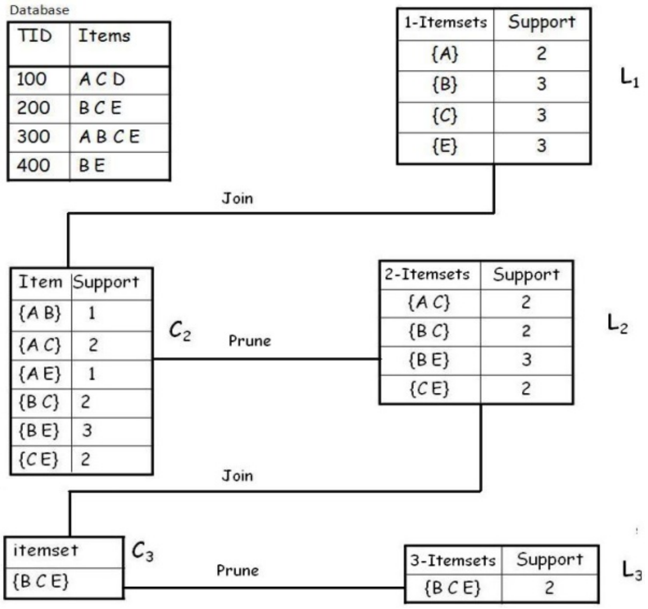
RULE GENERATION OF THE APRIORI ALGORITHM

```
Step 1: For each frequent K-itemset fk, K≥ 2 do
Step 2: H1= {i/I € fk}
Step 3: Call ap_genrules(fk,H1)
Step 4: End for
```

PROCEDURE ap_gen rules(fk,Hm)

```
Step 1: K=f |k|
Step 2: m=|Hm|
Step 3: if k>m+1 then
Step 4: Hm+1=apriori_gen(Hm)
Step 5: for each hm+1 € Hm+1 do
Step 6: Conf = σ(fk) |σ(fk-hm+1)
Step 7: if cont ≥ min conf then
Step 8: output the rule(fk-hm+1) →hm+1
Step 9: else
Step 10: delete hm+1 Hm+1
Step 11: end if
Step 12: end for
Step 13 : all ap_genrules(fk.Hm+1)
Step 14:end if
```

Generating frequent item sets:



Database

TID	Items
100	A C D
200	B C E
300	A B C E
400	B E

1-Itemsets	Support
{A}	2
{B}	3
{C}	3
{E}	3

Item	Support
{A B}	1
{A C}	2
{A E}	1
{B C}	2
{B E}	3
{C E}	2

2-Itemsets	Support
{A C}	2
{B C}	2
{B E}	3
{C E}	2

itemset
{B C E}

3-Itemsets	Support
{B C E}	2

Figure : Generation of Candidate Itemsets and Frequent Itemsets (Min_supp=2 (20%))

3. SECURING THE FREQUENT ITEMSETS:

Cryptography is a Greek word which means secret writing. Today this term refers to the science and art of transforming messages to make them secure and immune to attacks [1]. For the purpose of security and privacy, we need to encrypt the message at the sender side and decrypt it at the receiver side. So cryptography is the study of creating and using encryption and decryption techniques. Cryptography is divided into two types, Symmetric Key Cryptography and Asymmetric key Cryptography.

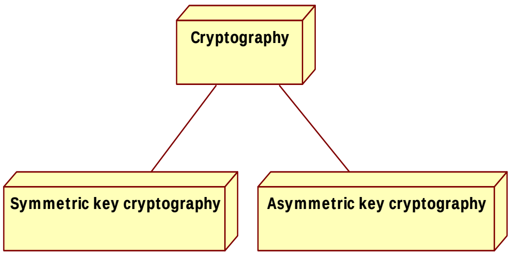


Figure: Types of cryptography

- In Symmetric Key Cryptography two types of ciphers, substitution cipher and transposition cipher are used [1]. In substitution cipher one symbol of the plaintext is replaced by another symbol. Substitution cipher has further two types.
- Mono alphabetic substitution cipher, in which a character in the plaintext is always changed to the same character in the cipher text. The well known example of Mono alphabetic substitution cipher is the CAESAR cipher which always change a to d. In poly alphabetic substitution cipher a single character in the plaintext is changed to many characters in the cipher text. The well known example of poly alphabetic substitution cipher is VIGENERE cipher which changes a single character in the plaintext into many characters in the cipher text by considering the position of character in the plaintext.
- In transposition cipher the characters in the plaintext are swapped to get the cipher text i.e. the characters retain their plaintext form but their position is changed. The plaintext is organized into two dimensional table and columns are interchanged according to a predefined key.

3.1 PLAYFAIR CIPHER:-

- The Playfair cipher encrypts pairs of letters (digraphs), instead of single letters. This is significantly harder to break since the frequency analysis used for Simple substation cipher is considerably more difficult.
- Memorization of the keyword and 4 simple rules is all that is required to create the 5 by 5 table and use the cipher.

The Playfair cipher uses a 5 by 5 table containing a key word or phrase. To generate the table, one would first fill in the spaces of the table with the letters of the keyword (dropping any duplicate letters), then fill the remaining spaces with the rest of the letters of the alphabet in order (to reduce the alphabet to fit you can either omit "Q" or replace "J" with "I"). In the example to the right, the keyword is **"KEYWORD"**.

To encrypt a message, one would break the message into groups of 2 letters. If there is a dangling letter at the end, we add an X. For example "Secret Message" becomes "SE CR ET ME SX SA GE". We now take each group and find them out on the table. Noticing the location of the two letters in the table, we apply the following rules, in order.

K	**E**	**Y**	**W**	**O**
R	**D**	A	B	C
F	G	H	I	L
M	N	P	Q	S
T	U	V	X	Z

1. If both letters are the same, add an X between them. Encrypt the new pair, re-pair the remaining letters and continue.
2. If the letters appear on the same row of your table, replace them with the letters to their immediate right respectively, wrapping around to the left side of the row if necessary. For example, using the table above, the letter pair GI would be encoded as HL.

3. If the letters appear on the same column of your table, replace them with the letters immediately below, wrapping around to the top if necessary. For example, using the table above, the letter pair KF would be encoded as RM.
4. If the letters are on different rows and columns, replace them with the letters on the same row respectively but at the other pair of corners of the rectangle defined by the original pair. The order is important - the first letter of the pair should be replaced first. For example, using the table above, the letter pair EB would be encoded as WD.

Plaintext: SECRET MESSAGE
Digraphs : SE CR ET ME SX SA GE
Cipher text : NO RD KU NK QZ PC ND

Here if in a digraph, if any of the character is repeated we have to place 'X' after first character in the digraph. In the above example, SS is repeated so it is written as SX and SA.

To decipher, ignore rule 1. In rules 2 and 3 shift up and left instead of down and right. Rule 4 remains the same. Hence the same keyword is used to decrypt the message.

ALGORITHM PLAYFAIR
Step 1: Input the keyword
Step 2:Set_key(Keyword);
Step 3: Key_gen();
Step 4:encrypt(plaintext)
Step 5:decrypt(ciphertext)

SET_KEY(KEYWORD):This procedure removes the repeated charecters in the given key and makes the key ready to fit into the matrix.

KEY_GEN():This procedure replaces "i "in place of " j ".

ENCRYPT(PLAINTEXT):Encrypts the given text by implementing the rules of play fair cipher technique. DECRYPT(CIPHERTEXT):Decrypts the cipher text by implementing the rules of play cipher technique.

4. FUTURE ENHANCEMENT:

For future enhancement the security algorithm could be modified.

MODIFIED VERSION OF PLAYFAIR CIPHER

The problems in 5×5 matrix playfair cipher arise when either I or J , or both appear in the key word. Since this technique only works with alphabetics so we cannot encrypt the numeric characters. Moreover alphanumeric characters make the cipher text stronger to break it. So, In this study we proposed 7×5 matrix playfair cipher which efficiently handles these problems. In 7×5 matrix playfair cipher any alphanumeric can be selected as a key word. . The matrix is filled in the order, where we first place the keyword and the remaining cells are filled with alternatively placed alphabets and numbers. In this matrix we use the alphabets a-z and numbers 1 to 9.

To encrypt the plaintext, the same rules presented in [3] are followed with the following modification.

- In the below example, we have taken the keyword as "IDEA145" and have the filled the matrix
- We have taken the plain text as "CHANGES THE LIFE"
- Now we encrypt the message "CHANGES THE LIFE" by using modified playfair cipher.

Table: Modified 7*5 playfair cipher

I	D	E	A	1	4	5
B	2	C	3	F	6	G
7	H	8	J	9	K	L
M	N	O	P	Q	R	S
T	U	V	W	X	Y	Z

Plaintext: CHANGES THE LIFE
Diagraphs: CH AN GE ST HE LI FE
Cipher text: 28 DP C5 MZ 8D 75 C1

CH	AN	GE	ST	HE	LI	FE
⇩	⇩	⇩	⇩	⇩	⇩	⇩
28	DP	C5	MZ	8D	75	C1

By decrypting the above cipher text we get a single sentence "CHANGES THE LIFE" which is the original plaintext message. The mapping of the cipher text into plaintext is shown in Fig. There correspondence between cipher text and plaintext is one-to-one. So there is no confusion in the decryption.

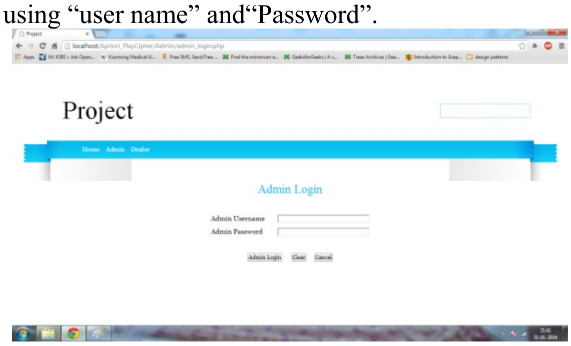
APPLICATION:

The above paper can be applied in the real time scenarios. For example we have taken a scenario of a super market, in which there will be an agreement between Admin and Dealer. Admin for a super market wants to know frequently sold complimentary items to increase his sales. For this he uses apriori algorithm, it generates the frequently sold item set. There may be a scenario of generating of more than one rule by apriori algorithm; in this case admin selects the rule of his interest and remaining stock in his super market. . After that to have secure transmission between admin and dealer who can also be termed as supplier. Admin encrypts the rule by using play cipher technique and sends to the dealer. Now comes the dealer's role, he decrypts it using same play cipher algorithm and places the required stock to the admin. Here we also providing authentication for both admin and dealer.

Screenshots for the above application

1) Admin login:-

When Admin wants to use the application.He authenticates by using "user name" and"Password".

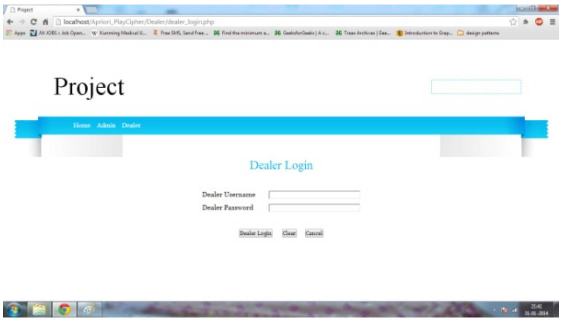


Admin Username
Admin Password

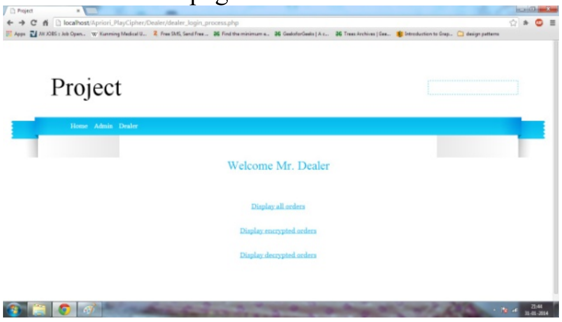
After authentication he performs various tasks like finding association rules by using apriori algorithm and encrypting the rule by using play cipher technique.

2) Dealer login:-

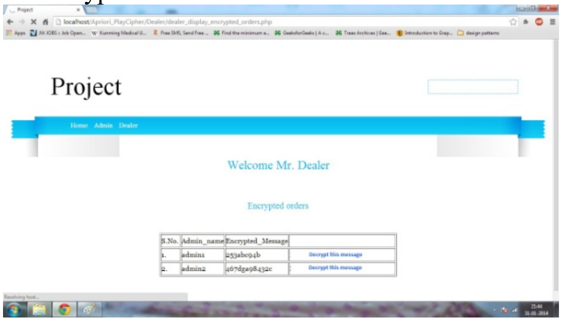
Dealer who wants to use the application services also authenticates same as Admin.



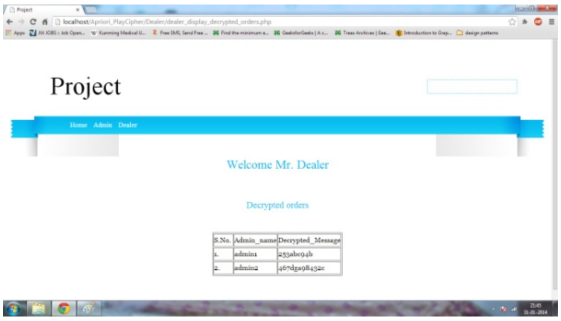
3) Dealer’s welcome page:



4) Dealer checks for various order form different admins and he decrypts the order he desired


Project
localhost/Apriori_PlayCipher/Dealer/dealer_display_encrypted_orders.php
Project
Home Admin Dealer
Welcome Mr. Dealer
Encrypted orders

S.No.	Admin_name	Encrypted_Message	
1.	admin1	253abc94b	Decrypt this message
2.	admin2	467dga98432c	Decrypt this message


Project
localhost/Apriori_PlayCipher/Dealer/dealer_display_decrypted_orders.php
Project
Home Admin Dealer
Welcome Mr. Dealer
Decrypted orders

S.No.	Admin_name	Decrypted_Message
1.	admin1	253abc94b
2.	admin2	467dga98432c

After he decrypts he places the required to respective admin.

REFERENCES:

[1] Introduction to Data Mining Pang-Ning Tan, Michael Steinbach, Vipin Kumar

[2] William Stallings' Cryptography and Network Security: Principles and Practice, 5e.

[3]www.cryptoolonline.org/index.php?lang=en

[4] en.wikipedia.org/wiki/Playfair_cipher

[5] An Algorithm for Frequent Pattern Mining Based On Apriori Goswami D.N.*, Chaturvedi Anshu.Raghuvanshi C.S. SOS In Computer Science Jiwaji University Gwalior.

[6] A Version of Playfair Cipher Using 5*5 Matrix. A. Aftab Alam, B. Shah Khalid, and C. Muhammad Salam.

[7] en.wikipedia.org/wiki/Data_mining.

[8] www.slideshare.net/zafarjcp/data-mining-association-rules-basics.

[9] Playfair Cipher - Rumkin.com

[10] learncryptography.com/playfair-cipher/