Frequent pattern mining for parameterised automatic variable key-based cryptosystems

Shaligram Prajapat

International Institute of Professional Studies,
Devi Ahilya University, India
Email: Shaligram.prajapat@gmail.com

Abstract: Huge amount of information is exchanged electronically in most enterprises and organisations. In particular, in all financial and e-business set ups the amount of data stored or exchanged is growing enormously over public network among variety of computing devices. Securing this gargantuan sized input is challenging. This paper provides a framework for securing information exchange using parametric approaches with AVK approach and investigating strength of this cryptosystem using mining algorithms on symmetric key-based cryptosystem. This work demonstrates association rule application as one of the component of cryptic mining system used to process the encrypted data for extracting use full patterns and association. The degree of identified patterns may be use full to rank the degree of safety and class of cryptic algorithm, during auditing of security algorithms.

Keywords: mining algorithms; symmetric key cryptography; automatic variable key; AVK.

Reference to this paper should be made as follows: Prajapat, S. (2020) 'Frequent pattern mining for parameterised automatic variable key-based cryptosystems', *Int. J. Business Intelligence and Data Mining*, Vol. 16, No. 1, pp.33–48.

Biographical notes: Shaligram Prajapat has received his PhD from Mauling Azad National Institute of Technology (M.A.N.I.T. Bhopal) India and associated with Devi Ahilya University Indore India. His research interests include data mining, algorithms, and security. He received his MS, MTech and MPhil in Computer Science from Devi Ahilya University Indore. He is member of IE (India) UACEE and fellow member of ISCA, Vigyan Bharti, Cryptology Research Society of India (CRSI), ISROSET, ACM, IEEE, CSI, CSTA, IAENG and ISTE.

This paper is a revised and expanded version of a paper entitled 'Cryptic-mining: association rules extractions using session log' presented at 15th International Conference on Computational Science and its Applications (ICCSA) 2015, Banff, AB, Canada, 22–25 June 2015.

1 Introduction

Recently all information transmission including financial and e-commerce takes place among entities which are far apart or does not know ever before, but participates in transaction. During this information transaction they uses public network in encrypted formats using symmetric key algorithms (in case of large information contents).

Analysing a large collection of cipher logs with the aim of finding association rules is interesting and hard problem in cryptic mining domain. Classification and clustering of cipher logs are also essential components of cryptic mining. The term 'CRYPTIC MINING' explores cross-fertilisation scope among two established domain of cryptography and data mining. Both areas have their own style of dealing with problems, algorithms, and their applications. Data mining perspective explores large collection of item sets or database to extract useful novel patterns, for overall organisational growth (Ashrafi et al., 2004). The data dependant technique exploits capabilities by prediction of future items sets or associations. Conventionally, these algorithms are developed and applied for business market prediction and analysing enterprise data. On the other side of the coin, Cryptography domain hides the patterns among input data to make the plain text indistinguishable or scrambled to prevent from tracing or maliciously uncovering by unauthorised parties such as hacker or cryptanalyst. In cryptanalysis process, successive attempts for detecting association among plaintext from and encrypted text, exploring partially or complete information without knowing the secret key is made (Ashrafi et al., 2005). The try is made to find out some of the components from possibility set P = {plaintext, key and list of weakness in cipher text leading to plain text or key}.

Thus cryptic mining framework makes optimum use of cryptography (hide properties or patterns of data), and data mining (reveal patterns in data) discipline. It is also worth mentioning here that, data mining is useful for bulk (large volume) data; on the other hand, cryptography is inefficient in handling large amount of data. So it opens scope for cryptic mining to strengthen cryptic algorithm for large input. It is advantageous to merge two to exploit benefits of one field over another.

Rivest (1993) equivalence table extends below the concept of cryptography for cryptic mining discipline.

Table 1 Correlation of terms of cryptography and cryptic mining

S. no.	*Cryptographic terms*	*Cryptic mining terms*
1	Secret key	Target function
2	Attack type	Learning protocols
3	Exact inference	Approximate
4	Uni-city distance	Sample complexity

For exchanging large information securely on public network symmetric key-based cryptosystem is preferred and to extend its security key length is extended that requires large effort to compute and process ciphertext. Automatic variable key (AVK) is being developed as an alternative to this as an optimised parameterised cryptosystem. This time variant key approach enabled with advanced techniques like machine learning, data mining and soft computing will empower AVK cryptosystem. In the next sections, illustration AVK-based cryptosystem with its analysis by checking association rules is presented for symmetric key cryptographic technique of information exchange.

2 Cryptic mining

Formally, cryptic mining can be defined as a set of cryptic algorithms that extracts patterns of probable relationships in plaintext, ciphertext, or both (not based on cause-effect relationships). Moreover, cryptic mining methods explores significant

relationships, not just between key and key length, but also among number of parameters used for key computation, key strength, ciphertext-plaintext relationship and running encryption/decryption key cryptanalyst have to find out correlations and determine what is significant. Following are some essential components of cryptic mining.

2.1 Cryptic association rule discovery (cryptic ARM)

Cryptic ARM (Prajapat et al., 2015a): ideally in a cryptosystem one key must be independent from other key, but in practice it may not be true. For example, in case of parameterised AVK model some parameters (used in previous key construction) may be used to construct present or future keys. Cryptic association rule attempts to discovery discoverers useful association rule to identify relationships or association among keys (or a part of key as parameters) generated in discrete sessions and dependencies to explore the frequent parameters of key. The generated rules may be of huge size, so reduction is essential using the methods presented in Ashrafi et al. (2004, 2005, 2007), Taniar et al. (2008), Prajapat and Thakur (2015a) and Daly and Taniar (2004).

2.2 Cryptic clustering

To find out type or information about cryptic algorithms generating ciphers, cryptic clustering algorithms are devised (Prajapat et al., 2016; Prajapat and Thakur, 2015b). Classified ciphers may be useful in identifying degree of weakness in cryptic clusters in terms of size of their clusters. Based on similarity of cryptograms or ciphers the useful information can be extracted (such as key length), By grouping cipher text into several clusters based on similarity of key patterns, key length, cipher type and other attribute will provide useful parameter for identification of weaknesses and strength of cryptosystem can be expressed and hence its auditing can be done.

2.3 Cryptic forecasting

Based on historic information of cipher-log and training set a cryptosystem can be devised to forecast the system behaviour from past experience (Prajapat et al., 2015b).

3 Cryptanalysis of symmetric and parameterised cryptosystem

Enabling Cryptanalyst with advanced tools is ever demand for identification of weakness and flaws of cryptosystem. In polynomial time, Cryptanalyst is interested to extract useful guess for detecting original information, from huge corpus of ciphers. Cryptanalyst may have captured large database and corpus containing variety of ciphers and hash files. When a cipher text is inserted into this dataset, it might be mixed within other ciphers generated from various other schemes including variations in key size, protocol, type of ciphers generation algorithm, degree of exposures of information about key space and many other information related to plaintext, cipher text, relationship between them. The cryptanalyst may develop a mechanism that will classify/sort/ group according to cipher type.

For security domain, standard literature of data mining insinuates about classification of credit card transactions into fraudulent or genuine based upon the credit history and patterns of earlier transactions (Prajapat and Thakur, 2015a). These techniques are also useful for decision making of new transaction that falls into one of the given categories: {authorised transaction state, state of asking for further identification before authorisation, state of unauthorised transaction, critical state – with not to authorise but contact police}.

Table 2 Availability of information and cryptanalysis tasks

Type of attack	*Available information for cryptanalysis*		
	Encryption algorithm	*Cipher text (to be decoded)*	*Addition requirements*
Cipher text only	Available	Available	--
Known plaintext	Available	Available	One or more (P, C) pairs, where P = plain text, C = cipher text
Chosen plaintext	Available	Available	Cipher text chosen by cryptanalyst and corresponding decrypted plain text, i.e., (C, p = (C))
Chosen text	Available	Available	Plain text chosen by cryptanalyst and corresponding cipher text, cipher text chosen by cryptanalyst and corresponding decrypted plain text

The 'stream cipher' and 'block ciphers' are the two variants of symmetric key cryptography. Data mining algorithms for mining of stream data are available to analyse rapid and continuous changing data streams. Such streams may be generated Available from network traffic, multimedia streams, patient monitoring system (ICU patients) and stock market price graphs. Online analysis with limited amount of memory and continuously changing data creates various difficulty levels (Ashrafi et al., 2004). In early 21st century, data stream mining work was in inception phase.

The classification approaches for mining high-speed data streams has been presented (Domingos and Hulten, 2000) in literature. For association rule, by exploring frequent patterns in data streams in multiple time granularities is also available (Giannella, 2003) similarly for cipher some models are also available (Prajapat and Thakur, 2016a, 2016b).

The clustering of clustering data streams, concepts theory and practices (Guha et al., 2003) provides insights on unlabeled datasets. For detecting changes in data streams the change detection (Kifer et al., 2004) processes are also reported. Dimensionality reduction concepts are also available for adaptive unsupervised stream mining. Time series analysis for learning of data is also available. Similar to decade's history of databases and cryptography, machine learning techniques are now applied to improve data stream mining.

Now, it has been well established for data analytics, artificial intelligence and computational intelligence. The classical example of learning can be seen in e-mail accounts, i.e., automatic classification of e-mails into useful mails and SPAMS. In literature, the landmark article of Rivest (1993) delineates a survey of the relationship between the fields of cryptography and machine learning. Along with an emphasis on how each field has contributed ideas and techniques to the other. The work gives

directions for future cross-fertilisation. The established correlations are depicted in Table 1.

These available literatures forms foundation for making assumptions with application of mining techniques for development of effective crypto systems. In Prajapat and Thakur (2016a), Rivest (1993), Tjioe and Taniar (2005), Gaurav et al. (2008), Rao (2003) towards classification of cipher some work at IIT-K has been done. Ideally, there should not be any pattern in packets after receiving cipher texts. Theoretically, all generated ciphers must be uniform, but, their classification algorithms observe some pattern in the ciphers. These patterns can be used to identify or guess about cipher text and receive knowledge of enciphering algorithms. Though, ciphertext can be obtained easily, but to identify which encryption algorithm is used for generating that cipher is difficult task for cryptanalyst. In Prajapat and Thakur (2016a), 'ciphertext only' experiments was carried out to identify the encryption algorithm for producing the given cipher text. For their experiment Blowfish, RC4 ciphers were taken for classification using SVM and ANN.

4 AVK model for symmetric cryptosystem

AVK concept is taking shape for symmetric key encryption decryption techniques. Professor C.T. Bhunia, a pioneer researcher in this domain proposed key variability concept. Later, Chakrabarti introduced some concepts and extended work. Fibonacci-Q matrix (Girish, 2002; Maheshwari, 2001), sparse approach (Prajapat et al., 2012) are alternative variants for realisation of time variant key towards symmetric key for AVK-based methods. A novel approach of parameter-based key computation has been proposed and analysed for adding more security in key exchange (Prajapat et al., 2014).

Table 3 AVK model of symmetric cryptosystem

Session ID	*Alice (T_x)*	*Bob (R_x)*	*Bob(T_x)*	*Alice(R_x)*	*Remarks*
Initial agreement	00000010 (K_a)	10	00000110 (K_b)	110	For next slot Alice and Bob will use 00000110, 00000010 respectively
S_1	00000011 = D_1 compute $C_1 = D_1 K_b$ and transmit C_1	Compute $C_1 K_a$ and get plaintext D_1 = 00000011	00000111 = D_2 compute $C_2 = D_2 K_b$ and transmit C_2	Compute $C_2 K_b$ and get plaintext D_2 = 00000111	Alice and Bob will compute new keys by compute $D_2 K_b$ and $D_1 K_a$ respectively for new session
S_2	00000100 = D_1 compute $C_1 = D_1 K_b$ and Transmit C_1	Compute $C_1 K_a$ and get plaintext D1 = 00000100	00001000 = D_2 compute $C_2 = D_2 K_b$ and transmit C_2	Compute $C_2 K_b$ and get plaintext D_2 = 00001000	Compute new session keys similar to previous step, i.e., Alice and Bob will compute new keys by compute $D_2 K_b$ and $D_1 K_a$ respectively for new session

Table 4 Transaction log of parameter usage for computation of shared key of AVK

Data	*Shared key*	*Key with parameters*
D_1	SK_1	$SK_1 = f(p_1, p_3, p_4, p_6)$
D_2	SK_2	$SK_2 = f(p_3, p_5)$
D_3	SK_3	$SK_3 = f(p_4, p_5, p_6)$
D_4	SK_4	$SK_4 = f(p_2, p_3, p_5)$
D_5	SK_5	$SK_5 = f(p_1, p_2)$
D_6	SK_6	$SK_6 = f(p_1, p_2, p_3, p_6)$

Table 5 Frequent single parameters

C_1		
Parameter	*1-Frequent*	*Minimum frequent parameter*
p_1	3	Parameter with minimum frequency = p_4
p_2	3	
p_3	4	
p_4	2	
p_5	3	
p_6	3	

Table 6 Paired parameter frequent sets

C_2		
Parameter pair	*Frequent set*	*Minimum frequent parameters*
p_1, p_2	2	(p_1, p_5) and (p_2, p_4) = $(p_1$ XOR $p_5)$ XOR $(p_2$ XOR $p_4)$
p_1, p_3	2	
p_1, p_4	1	
p_1, p_5	0	
p_1, p_6	2	
p_2, p_3	2	
p_2, p_4	0	
p_2, p_5	1	
p_2, p_6	1	
p_3, p_4	1	
p_3, p_5	2	
p_3, p_6	2	
p_4, p_4	1	
p_4, p_5	2	
p_4, p_6	1	

Thus in this way, Alice has sent 00000011, 00000100 in session 1 and 2 respectively and Bob has exchanged 00000111, 00001000 respectively. Recently, for generation of AVK under various approaches in cryptographic system are being investigated for random

input parameters (Prajapat et al., 2014). In Girish (2002), intelligent techniques have been pointed out for shared key generation. In case of multiparty communication, the concept of shared key is used to enhance the security level. Chakrabarti et al. (2008) have proposed techniques that are based on minimal frequent set, candidate generation, partition scheme, intersection of item-set count. These methods are based on feature analysis, centroid analysis, inter-centroid distance, extraction scheme of vowel, index position of character, support analysis and confidence rule. The minimal frequent set is formed by minimum probability of combination of items. The shared key is the XOR of XOR of each of the pairs of elements of the set. Among the combination of the keys only (p_1, p_5) and (p_2, p_4) have least probability and it is zero. So, minimal frequent set = {p_1, p_5, p_2, p_4} and shared key = (p_1 XOR p_5) XOR (p_2 XOR p_4).

Key evaluation based on candidate generation, shared key = p_4 XOR (p_1 XOR p_5) XOR (p_2 XOR p_4).

Table 7 Session wise parameter used for key computation in AVK

Data	*Parameters for key →*					
	P_1	P_2	P_3	P_4	P_5	P_6
D_1	1	0	1	1	0	1
D_2	0	0	1	0	1	0
D_3	0	0	0	1	1	1
D_4	0	1	1	0	1	0
D_5	1	1	0	0	0	0
D_6	1	1	1	0	0	1

During the first pass support court are: {p_1: 3, p_2: 3, p_3: 4, p_4: 2, p_5: 3, p_6: 3}, we get most frequent = p_3. In second pass support counts are: {(p1, p2): 2, (p1, p3): 2, (p1, p4): 1, (p1, p5): 0, (p2, p3): 2, (p2, p4): 0, (p2, p5): 1, (p3, p5): 2, (p4, p5): 1}, we get most frequent = {(p1, p2), (p2, p3), (p3, p5)} == {p1, p2, p3, p5} so shared key may be intersection of two element set = p3.

5 Association rule extractions

Approach 1: For investigation of association rule we consider the following session log with parameters for key construction (Table 6). Here we have only four-parameter space for key computations {p_1, p_2, p_4, p_5} within four sessions {s_1, s_2, s_3, s_4}. For investigation of association rules we assume minimum support of 0.5 and minimum confidence level of 0. 75.

Table 8 Session wise log of used parameters

Session ID	*Parameters for key*
s_1	p_1, p_2
s_2	p_1, p_2, p_4
s_3	p_1, p_5
s_4	p_2, p_4, p_5

Using straightforward approach we are computing all the combination of parameters that are in Domain space are used to identify which combinations are frequent. Confidence level association rules above threshold and that have the confidence are shown in Table 7.

Table 9 Parameters sets and corresponding frequencies

Parameter sets	*Frequency*
p_1	3
p_2	3
p_4	2
p_5	2
$\{p_1, p_2\}$	2
$\{p_1, p_4\}$	1
$\{p_1, p_5\}$	1
$\{p_2, p_4\}$	2
$\{p_2, p_5\}$	1
$\{p_4, p_5\}$	1
$\{p_1, p_2, p_4\}$	1
$\{p_1, p_2, p_5\}$	0
$\{p_1, p_4, p_5\}$	0
$\{p_2, p_4, p_5\}$	1
$\{p_1, p_2, p_4, p_5\}$	0

With minimum support of 0.5, the parameter set that occur in at least two transactions are explored, and would be frequent for this case is given in Table 8.

Table 10 Parameters sets and corresponding decisions

Parameter group	*Decisions for frequent*	*Result set*
Single	All individual frequent	p_1, p_2, p_4, p_5
2-parameter sets	Only 2 parameters out of 6	(p_1, p_2) and (p_2, p_4)
3-parameter sets	None	Null set
4-parameter sets	None	Null set

Exclusion of non-frequent parameters will result in Table 11.

Table 11 All frequent parameter sets

Parameters	*Frequency*
p_1	3
p_2	3
p_4	2
p_5	2
p_1, p_2	2
p_2, p_4	2

Table 10 illustrates determination (identification) process of useful association rules from all possible combinations using desirable confidence. Only the rule $p_5 \rightarrow p_2$ qualifies the criteria and insinuates the possibility of next parameters for the computation of key in subsequent sessions

Table 12 Possible parameters association rules

Possible rule	*Confidence*	*Desirable confidence*
$p_1 \rightarrow p_2$	2/3	<75%
$p_2 \rightarrow p_1$	2/3	<75%
$p_2 \rightarrow p_4$	2/3	<75%
$p_5 \rightarrow p_2$	3/3	>75%

Approach 2: Consider the following table A where we have only four parameters for key computations {p_1, p_2, p_4, p_5} and we have only four-session information where each session has corresponding parameter information of the session. We are interested in finding association rule with a minimum 'support' of 50% and minimum 'confidence' of 75%.

Table 13 Possible parameters association rules

Session ID	*Parameters*
S_1	p_1, p_2
S_2	p_1, p_2, p_4
S_3	p_1, p_5
S_4	p_2, p_4, p_5

Initially list of all parameter combination from domain space appearing in session are searched for frequent, and used for framing association rules with necessary confidence using these frequent combinations. Let following set A, is the initial step of, four parameter-based communication. All the combinations 4 parameters corresponding frequency of appearance in the transaction is specified below in the format: {parameter combination: frequency}.

$$A = \{\{p_1 : 3\}, \{p_2 : 3\}, \{p_4 : 2\}, \{p_5 : 2\}, \{(p_1, p_2) : 2\}, \{(p_1, p_4) : 1\}, \{(p_1, p_5) : 1\}, \{(p_2, p_4) : 2\}, \{(p_2, p_5) : 1\}, \{(p_4, p_5) : 1\}, \{(p_1, p_2, p_4) : 1\}, \{(p_1, p_2, p_5) : 0\}, \{(p_1, p_4, p_5) : 0\}, \{(p_2, p_4, p_5) : 1\}, \{(p_1, p_2, p_4, p_5) : 0\}\}$$

Now set B is computed with minimum support of 0.5 parameter set, appearing in at least 2 sessions would be frequent. By default individual parameters are frequent, 2-parameter term sets, 2 out of 6 are frequent, and none of the 3-parameter sets and 4-parameter is frequent.

$$B = \{\{p_1 : 3\}, \{p_2 : 3\}, \{p_4 : 2\}, \{p_5 : 2\}, \{(p_2, p_4) : 2\}$$

Since, rules >= specified minimum confidence are confident, so, we can compute, If two 2-parameter sets (p_1, p_2) and (p_2, p_4) lead to required confidence of 0.75. Every 2-parameter sets (p_i, p_j) can lead to two rules $p_i \rightarrow p_j$, $p_j \rightarrow p_i$, if both satisfy the required

confidence. $p_1 \rightarrow p_2$ with confidence of 2/3 = 67%, $p_2 \rightarrow p_1$ with confidence of 2/3 = 67%, $p_2 \rightarrow p_4$ with confidence of 2/3 = 67%, $p_5 \rightarrow p_2$ with confidence 100%, Further, only the last rule $p_5 \rightarrow p_2$ has confidence above minimum 75% required and qualifies.

Limitation of Approach 2: We can observe that only 4 parameters grow quickly with the number of parameters. Which is very small number for now specifically for 24 = 16. But for 8 parameters it would have 256 combinations and if 10 parameters then 1,024 combinations and in general 2^n combinations. This exponential growth is undesirable.

Approach 3: Modified approach assumes ignoring the parameter which does not appear or having frequency zero, counting all possible parameter combinations.

Table 14 All probable parameter combination with non zero frequency

Session ID	*Parameters*	*Combinations*
S_1	p_1, p_2	$\{p_1, p_2\}$
S_2	p_1, p_2, p_4	$\{p_1, p_2\}, \{p_1, p_4\}, \{p_2, p_4\}, \{p_1, p_2, p_5\}$
S_3	p_1, p_5	$\{p_2, p_5\}$
S_4	p_2, p_4, p_5	$\{p_2, p_4\}, \{p_2, p_5\}, \{p_4, p_5\}, \{p_2, p_4, p_5\}$

Now we compute frequent parameters with corresponding frequencies: $\{\{p_1: 3\}, \{p_2: 3\}, \{p_4: 2\}, \{p_5: 2\}, \{(p_1, p_2): 2\}, \{(p_1, p_4): 1\}, \{(p_1, p_5): 1\}, \{(p_2, p_4): 2\}, \{(p_2, p_5): 1\}, \{(p_4, p_5): 1\}, \{(p_1, p_2, p_4): 1\}, \{(p_2, p_4, p_5): 1\}\}$. Eleven frequent entries are obtained, while is significantly less then 15. This would be significant for big parameter tables developed over large number sessions with passage of time. But a better alternative can be found out using apriori algorithm. In its first phase, parameters >= minimum support are found and in second part, satisfying association rules are determined using the information of frequent parameters. We assume information of five sessions with six parameters space for key $P= \{p_1, p_2, p_4, p_5, p_{10}, p_{11}\}$.

Table 15 Five sessions with corresponding parameter

Session ID	*Parameter*
S1	p_1, p_2, p_{10}, p_4
S2	p_1, p_2, p_4
S3	p_1, p_5, p_{11}
S4	p_1, p_4, p_5
S5	p_2, p_4, p_5

Computing L_1: 1-frequent parameters $L_1 = \{\{p_1: 4\}, \{p_2: 3\}, \{p_4: 4\}, \{P_5: 3\}\}$, Candidate Set C_2: 2-parameter set $C_2 = \{\{(p_1, p_2): 2\}, \{(p_1, p_4): 3\}, \{(p_1, p_5): 2\}, \{(p_2, p_4):3\}, \{(p_2, p_5): 1\}, \{(p_4, p_5): 2\}\}$, inferred association rules = $\{p_1 \rightarrow p_4 = 0.74, p_4 \rightarrow p_1 = 0.75, p_2 \rightarrow p_4 = 1, p_4 \rightarrow p_2 = 0.75\}$.

6 Apriori approach for parameter prediction

Consider a Transaction log with information about 25 sessions, i.e., $S = \{S_1, S_2, \ldots, S_{25}\}$ with exchanging the key using parameters only method from parameter space of

16 possibilities, i.e., P = {p_1, p_2, …, p_{16}}. A key of a particular session will be random selection of some parameters from P and then applying secret algorithm to compute key of that particular session. In the AVK environment, We assume that cryptanalyst or hacker somehow recorded traces of parameters used in few sessions say 25, without the information of function he may be interested to know the frequent parameters or guessing future parameters based on association rules, applied on these parameter to recomputed the future key session.

6.1 Transaction log containing parameter traces

The frequency of each parameter in the session logs is given in following set, where set element = {parameter, frequency of parameter} is listed below: {{p_1: 4}, {p_2: 13}, {p_3: 10}, {p_4: 11}, {p_5: 9}, {p_6: 9}, {p_7: 10}, {p_8: 2}, {p_9: 11}, {p_{10}: 6}, {p_{11}: 2}, {p_{12}: 1}, {p_{13}: 2}, {p_{14}: 1}, {p_{15}: 4}, {p_{16}: 2}}.

Table 16 Log of large session with corresponding parameter

S_1	p_1, p_2, p_4, p_6, p_{16}
S_2	p_1, p_3, p_4, p_6
S_3	p_4, p_5, p_7, p_9, p_{10}
S_4	p_2, p_4, p_6, p_3, p_9
S_5	p_2, p_3, p_5, p_7, p_9
S_6	p_{10}, p_{15}
S_7	p_1, p_2, p_4, p_6, p_{10}
S_8	p_8, p_{10}, p_{15}
S_9	p_2, p_3, p_4, p_5, p_6
S_{10}	p_2, p_3, p_5, p_7, p_9
S_{11}	p_2, p_4, p_9
S_{12}	p_2, p_4, p_6, p_7, p_9
S_{13}	p_1, p_2, p_3
S_{14}	p_3, p_4, p_5, p_7, p_9
S_{15}	p_5, p_6
S_{16}	p_7
S_{17}	p_7, p_8, p_9
S_{18}	p_1, p_2, p_4, p_6
S_{19}	p_2, p_3, p_5, p_7, p_9
S_{20}	p_4, p_5, p_7, p_9
S_{21}	p_{10}, p_{15}, p_{16}
S_{22}	p_2, p_3, p_4, p_6
S_{23}	p_5, p_7, p_9, p_{10}, p_{11}
S_{24}	p_{11}, p_{12}, p_{13}
S_{25}	p_{13}, p_{14}, p_{15}

6.2 Phase 1: Computation of frequent set

Assuming support of parameters (25% supports in 25 sessions) to occur in at least 7 sessions for computing first frequent parameter sets L1:

L_1: first frequent parameter set

Table 17 1-frequent parameter

Parameter	p_2	p_3	p_4	p_5	p_6	p_7	p_9
Frequency	13	10	11	9	9	10	11

Computation of C_2: there are 21 candidate for, 2-parameter set of C_2

$(p_2, p_3), (p_2, p_4), (p_2, p_5), (p_2, p_6), (p_2, p_7), (p_2, p_9)$
$(p_3, p_4), (p_3, p_5), (p_3, p_6), (p_3, p_7), (p_3, p_9)$
$(p_4, p_5), (p_4, p_6), (p_4, p_7), (p_4, p_9)$
$(p_5, p_6), (p_5, p_7), (p_5, p_9)$
$(p_6, p_7), (p_6, p_9)$
(p_7, p_9)

C_2

Table 18 2-candidate parameter set

Parameter set	*Frequency*
(p_2, p_3)	9
(p_2, p_4)	8
(p_2, p_5)	4
(p_2, p_6)	8
(p_2, p_7)	4
(p_2, p_9)	6
(p_3, p_4)	5
(p_3, p_5)	4
(p_3, p_6)	5
(p_3, p_7)	4
(p_3, p_9)	6
(p_4, p_5)	4
(p_4, p_6)	9
(p_4, p_7)	3
(p_4, p_9)	4
(p_5, p_6)	1
(p_5, p_7)	7

Table 18 2-candidate parameter set (continued)

Parameter set	*Frequency*
(p_5, p_9)	7
(p_6, p_7)	1
(p_6, p_9)	2
(p_6, p_9)	9

L_2: the frequent 2-parameter set

Table 19 2-frequent session with corresponding parameter

(p_2, p_3)	9
(p_2, p_4)	8
(p_2, p_6)	8
(p_4, p_6)	9
(p_5, p_7)	7
(p_5, p_9)	7
(p_7, p_9)	9

C_3: candidate sets of 3-parameter set and frequency

Table 20 3-frequent parameter candidate

Candidate set 3-parameter set	*Frequency*
p_2, p_3, p_4	4
p_2, p_3, p_5	4
p_2, p_4, p_6	8
p_5, p_7, p_9	7

L_3: 3-frequent-parameter set or L3

Table 21 3-frequent parameter set

3-frequent parameter set	*Frequency*
p_2, p_4, p_6	8
p_5, p_7, p_9	7

6.3 Phase 2: Computation of association rule

We compute 3-frequent-parameter set, i.e., L2,

Taking one parameter in antecedence from {p2, p4, p6} we get:

$p_2 \rightarrow p_4, p_6$

$p_4 \rightarrow p_2, p_6$

$p_6 \rightarrow p_2, p_4$

Rules with 2-parameter set in antecedence position

$p_4, p_6 \rightarrow p_2$,
$p_2, p_6 \rightarrow p_4$,
$p_2, p_4 \rightarrow p_6$

Taking support = 8 computation of confidence of association rules for parameters p_2, p_4, p_6 are given in Table 22.

Table 22 Association rules with corresponding support and confidence

Rule	*Support of (p_2, p_4, p_6)*	*Frequency of antecedence*	*Confidence*
$p_2 \rightarrow p_4, p_6$	8	13	0.61
$p_4 \rightarrow p_2, p_6$	8	11	0.72
$p_6 \rightarrow p_2, p_4$	8	9	0.89
$p_4, p_6 \rightarrow p_2$	8	9	0.89
$p_2, p_6 \rightarrow p_4$	8	8	1
$p_2, p_4 \rightarrow p_6$	8	8	1

With support = 7, computation of confidence for p_5, p_7, p_9 is shown in Table 23.

Table 23 Frequent rule base used for computation of AVK

Rule	*Support*	*Frequency of antecedent*	*Confidence*
p5 → p7, p9	7	9	0.78
p7 → p5, p9	7	10	0.7
p9 → p5, p7	7	11	0.64
p7, p9 → p5	7	9	0.78
p5, p9 → p7	7	7	1
p5, p7 → p9	7	7	1

With confidence = 0.7 cryptanalyst or hacker may infer all-7 rules (without rule number 3). So 5 of them also satisfy after checking L_2 also we get $p_2 \rightarrow p_3$ and $p_3 \rightarrow p_2$ and they both have confidence. $p_4 \rightarrow p_2$; $p_4 \rightarrow p_6$; $p_6 \rightarrow p_2$; $p_6 \rightarrow p_4$; $p_4, p_6 \rightarrow p_2$, $p_2, p_6 \rightarrow p_4$, $p_2, p_4 \rightarrow p_6$, $p_5 \rightarrow p_7$, $p_5 \rightarrow p_9$, $p_7 \rightarrow p_5$, $p_7 \rightarrow p_9$, $p_7, p_9 \rightarrow p_5$, $p_5, p_9 \rightarrow p_7$, $p_5, p_7 \rightarrow p_9$, $p_2 \rightarrow p_3$, $p_3 \rightarrow p_2$ (rules have been decomposed like $p_4 \rightarrow p_2, p_6$ by two rules $p_4 \rightarrow p_2$ and $p_4 \rightarrow p_6$). Thus, Association rule for predicting probable parameters from parameter space using association rule may provide hints for future parameters to predict key. But since both number of parameters as well as key of session is variable and changing from session to session, so the security of the system would not be compromised with the automatic variable scheme.

7 Future enhancement

Once Cryptic Mining algorithm has been implemented, tested and deployed, that can be deployed on the server for disfiguring the nature of information being transferred from servers and it would be possible to know which information going is genuine that is sent by some malicious application. Theft of such information could be analysed and

prevented. Honey pot system can save their information and can prevent stealing. Similarly, clustering of information request and reply pattern can provide useful insights.

The redundant and duplicate association rule reduction (Ashrafi et al., 2004, 2005, 2007; Tjioe and Taniar, 2005) and exception and negative rule detection techniques can be applied for shortening the decision time.

8 Conclusions

Cryptic mining tries to apply and use analysis of data from session logs, identifies patterns related to attacks (Prajapat and Thakur, 2016b). Finding the pre intimation of an attack can help to develop good prevention tool and techniques and seeing the action associated with an attack can help to locate vulnerabilities to control and possible damages. This scheme of secure information exchange over the network that may be useful for wired and wireless systems. AVK approach (Prajapat et al., 2012) is claimed to be secured but parameter-based communication would add extra security feature in the system.

Association rule for predicting probable parameters from parameter space using association rule may provide hints for future parameters to predict key. But since both number of parameters as well as key of session is variable and changing from session to session, so the security of the system would not be compromised with the automatic variable scheme. For improvement and learning Cryptic mining gains training database for mining from various sources and minimised association rules using approaches by Taniar et al. (2008) and Ashrafi et al. (2004, 2005). It tries to apply and use analysis of data from session logs, identifies patterns related to attacks. Finding the negative and exception rules pre intimation of an attack can help to develop good prevention tool and techniques and seeing the action associated with an attack can help locate vulnerabilities to control and possible damages.

References

Ashrafi, M.Z., Taniar, D. and Smith, K.A. (2004) 'A new approach of eliminating redundant association rules', *Proceedings of the 15th International Conference on Database and Expert Systems Applications (DEXA'2004)*, pp.465–474.

Ashrafi, M.Z., Taniar, D. and Smith, K.A. (2005) 'Redundant association rules reduction techniques', *Australian Conference on Artificial Intelligence 2005*, pp.254–263.

Ashrafi, M.Z., Taniar, D. and Smith, K.A. (2007) 'Redundant association rules reduction techniques', *International Journal of Business Intelligence and Data Mining*, Vol. 2, No. 1, pp.29–63.

Chakrabarti, P., Choudhary, A., Naik, N. and Bhunia, C.T. (2008) 'Key generation in the light of mining and fuzzy rule', *International Journal of Computer Science and Network Security (IJCSNS)*, Vol. 8, No. 9, pp.332–337.

Daly, O. and Taniar, D. (2004) 'Exception rules mining based on negative association rules', *Proceedings of the International Conference on Computational Science and Its Applications (ICCSA 2004)*, Part IV, Lecture Notes in Computer Science, Vol. 3046, pp.543–552, Springer.

Domingos, P. and Hulten, G. (2000) 'Mining high-speed data streams', *Proceedings of the sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp.71–80.

Gaurav, S., Karnik, H. and Manindra, A. (2008) *Classification of Ciphers using Machine Learning*, IIT-K thesis [online] http://www.security.iitk.ac.in/contents/publications/more/ciphers_machine_learning.pdf (accessed February 2015).

Girish, C. (2002) *Classification of Modern Ciphers*, IIK thesis [online] http://www.security.iitk.ac.in/contents/projects/cryptanalysis/repository/girish.pdf (accessed February 2015).

Goswami, R., Chakrabarti, S., Bhunia, A. and Bhunia, C. (2013) 'Generation of automatic variable key under various approaches in cryptographic system', *Journal of the Institution of Engineers (India): Series B*, Vol. 94, No. 4, pp.215–220.

Guha, S., Meyerson, A., Mishra, N., Motwani, R. and Callaghan, L.O. (2003) *Clustering Data Streams: Theory and Practice*, January[online] http://infolab.stanford.edu/~loc/tkdepaper.pdf.

Kifer, D., Ben-David, S. and Gehrke, J. (2004) 'Detecting change in data streams', *Proceedings of the 30th VLDB Conference*, Toronto, Canada, pp.180–191.

Maheshwari, P. (2001) *The Classification of Ciphers*, IIT-K thesis.

Prajapat, S, Sharma, A., Swami, S., Rajput D., Singroli, B. and Thakur, R.S. (2014) 'Sparse approach for realizing AVK for symmetric key encryption', *International Journal of Recent Development in Engineering and Technology (IJRDET)*, Vol. 2, No. 4, pp.15–18.

Prajapat, S. and Thakur, R.S. (2013) 'Time variant approach towards symmetric key', *SAI-Conference London*, October, cosponsored by IEEE and Springer.

Prajapat, S. and Thakur, R.S. (2014) 'Association rules for parameter prediction of AVK', *Proceedings of International Conference on Intelligent, Computational and Informative Systems (ICICIS)*, GOA, India, pp.181–185.

Prajapat, S. and Thakur, R.S. (2015a) 'Cryptic-mining: association rules extractions using session log', *Computational Science and Its Applications, ICCSA 2015*, June, Springer LNCS, Vol. 9158, pp.699–711, DOI: 10.1007/978-3-319-21413-9_12.

Prajapat, S. and Thakur, R.S. (2015b) 'Various approaches towards crypt-analysis', *International Journal of Computer Applications*, Vol. 127, No. 14, pp.15–24.

Prajapat, S. and Thakur, R.S. (2016a) 'Cryptic mining for automatic variable key based cryptosystem', *Elsevier Procedia Computer Science*, Vol. 78, No. 78C, pp.199–209, doi:10.1016/j.procs.2016.02.034.

Prajapat, S. and Thakur, R.S. (2016b) 'Cryptic mining: apriori analysis of parameterized automatic variable key based symmetric cryptosystem', *International Journal of Computer Science and Information Security*, Vol. 14, No. 2, pp.233–246.

Prajapat, S., Jain, A. and Thakur, R.S. (2012) 'A novel approach for information security with automatic variable key using Fibonacci Q-matrix', *IJCCT*, Vol. 3, No. 3 [online] http://www.interscience.in (accessed December 2014).

Prajapat, S., Parmar, G. and Thakur, R.S. (2015a) 'Towards investigation of efficient cryptosystem using Sgcrypter', *Special Issue of International Journal of Applied Engineering and Research (IJAER)*, Vol. 10, No. 79, pp.853–858.

Prajapat, S., Thakur, A., Maheshwari, K. and Thakur, R.S. (2015b) 'Cryptic mining in light of artificial intelligence', *IJACSA*, Vol. 6, No. 8, pp.62–69.

Prajapat, S., Sharma, A. and Thakur, R.S. (2016) 'AVK based cryptosystem and recent directions towards cryptanalysis', *Journal of Internet Computing and Services (JICS)*, Vol. 5, pp.97–110 [online] http://dx.doi.org/10.7472/jksii.2016.17.5.97 (accessed January 2015).

Rao, M.B. (2003) *Classification of RSA and IDEA Ciphers*, IITK thesis [online] http://www.security.iitk.ac.in/contents/projects/cryptanalysis/repository/anoopjain.pdf (accessed March 2015).

Rivest, R.L. (1993) 'Cryptography and machine learning', *Lecture Notes in Computer Science Advances in Cryptology – ASIACRYPT '91*, Vol. 739, pp.427–439.

Taniar, D., Rahayu, W., Lee, V.C.S. and Daly, O. (2008) 'Exception rules in association rule mining', *Applied Mathematics and Computation*, Vol. 205, No. 2, pp.735–750.

Tjioe, H.C. and Taniar, D. (2005) 'Mining association rules in data warehouses', *International Journal of Data Warehousing and Mining*, Vol. 1, No. 3, pp.28–62.