CRYPTOGRAPHIC STRENGTH ESTIMATION USING SPURIOUS KEYS WITH CONSIDERATION TO INFORMATION CONTENT IN THE MESSAGE

Mekala Rama Rao[1], L Pratap Reddy[2] , BHVS Narayana Murthy[3] and Maruti Sairam Annaluru[4]

[1,2]Department of Electronics and Communication Engineering, Jawaharlal Nehru Technological University Hyderabad, Telangana, India
[3,4]Research Centre Imarat, Hyderabad, Telangana, India

ABSTRACT

Among the available private key cryptosystems, namely stream ciphers and block ciphers, the advantage of block ciphers is that they can be synchronized i.e. losing one ciphertext can not affect the correctness of the decryption of the following blocks. The encrypter used in block ciphers is a memoryless device. Block ciphers can be easily standardized due to the fact that they transmit information in blocks. But the disadvantage is that identical plaintexts result in identical ciphertexts. These data patterns are not hidden by the algorithm, resulting in higher influence of cryptanalysis process. Strength of block ciphers is exposed mainly into the exploration of weakness of cryptosystem. Barring this approach, strength estimation based on spurious key analysis is proposed in this paper.

Till recent period, strength of a Cryptosystem is identified with the increasing key length. However, as per Shannon's proposal, strength of a Cryptosystem is dependent on Message also. Depending on the length of the message and the message space, we can estimate the actual strength of a Cryptosystem. As part of Shannon's model, spurious keys is the concept adopted for identifying the strength of the Cryptosystem. Standard block ciphers; ARC2, Blowfish, CAST, DES; are evaluated to understand Shannon's principle of Information Theoretic approach using Spurious keys. Spurious key generation algorithm is designed, developed for evaluating the strength of Cryptosystem. Spurious key logic and Key scheduling logic are the two main blocks of the proposed approach. Behavior of Spurious keys is evaluated on message text of two languages, through selection of ten different sub key spaces. Each sub key space is independent of other and is constructed with 10^8 keys from the total key space of 2^{64}. It is observed that the number of spurious keys identified in each sub key space is almost close to similar value of the respective language. Comparison is made through this evaluation to explore algorithmic strength with that of computational burden of the algorithm, which will help selection of algorithm based on the critical requirements of the field. The very purpose is to examine the possibilities of considering spurious key analysis as one of the strongest methods to estimate the strength of a Cryptosystem. Spurious key analysis is performed on two sets of plaintexts containing two different scripts namely English and Devanagari.

Natarajan Meghanathan et al. (Eds) : ICCSEA, WiMoA, SPPR, GridCom, CSIA - 2017
pp. 81– 95, 2017. © CS & IT-CSCP 2017 DOI : 10.5121/csit.2017.71107

KEYWORDS

Cryptography, Spurious Keys, Complexity, Strength of Algorithms, cryptanalysis, Language Based Security

1. INTRODUCTION

With the increased digital access in the areas of communication and financial transactions, shared data is becoming more and more personal in nature. The increase in personification of data is more significant these days. Huge amount of personal data is being shared every second. In this era of digital transactions, the security of one's wealth depends on the cryptosystem used by the e-wallet provider. The privacy of a social networking user depends on the strength of the cryptosystem used. The ever increasing growth of communication networks and the emergence of Internet Of Things (IoT) demands for undesired disclosure of information.

Present era of information explosion through social networking, it is undesirable to rely on concealment systems [3]. Information-theoretic analysis of information hiding [4] suggests that the capacity of data hiding suffers issues like side information being available to the intruder, tapping of wires, low rates of reliable transmission etc. Ideally, the system being used to transmit information should provide security even when the process of transmission is kept open. Cryptosystems are required to serve this purpose. The electronic data need to be maintained confidential for a long period. But security systems cannot provide long term security due to the increasing computational power. In the present trend of cloud computing, the owner of the data may not be confident that no adversary has access to the data, as it is being transmitted through a public network. If the original data is accessed, copied and stored then re-encryption of the data cannot guarantee any further security as the data is already stored. Deletion of data may not be completely guaranteed. Moreover, if quantum computers are used, then computing factors of large numbers and logarithms are also feasible, where RSA algorithm may not guarantee security.

Knowing the strength of a cryptosystem is vital in many applications where secure communication is desired. Cloud computing emphasized the need for data security [9]. Intruders can identify the vulnerabilities of a user whose security can be easily exploited, in such cases security of the entire cloud or network is at risk. Poor user security behaviour is a significant, perhaps even the major, detriment of the level of security incidents a company suffers [10]. Using a secure cryptosystem and efficient key management processes are necessary for such environment as most of the e-commerce applications are being run by cloud [11]. Key Dependent Messages (KDM) is another concept of encryption which allows requested plaintexts to depend on the underlying decryption key [36]. This concept is tested as an attack on block ciphers. This attack is called key dependent input (KDI) attack and it was found that for every function, a KDI secure encryption scheme can be built [37]. Encryption scheme of key dependent message (KDM) secure is reported to be secure even against an adversary who has access to encryptions of messages that depend on the secret key [38]. With ever increasing attacks on personal data and banking system through cyber space, it is important to have new dimensional measures to estimate the strength of a cryptosystem. One such method which considers message as a text and measures the strength of the system based on the language of the message, is presented in this paper.

1.1. Secrecy System Overview

A secrecy system, in general, can be viewed as a transformation from message space to ciphertext space [1]. This transformation should be reversible in nature to assure unique mapping between plaintext and ciphertext. Though encryption and decryption is same, the unique mapping between

these two spaces is determined by the chosen key. In these private key cryptosystems, sender and receiver should share a secret key which is unknown to the intruder. However, construction of algorithm, chosen key and the construction of message provides apriori knowledge to the intruder. The process of cryptanalysis involves obtaining posteri probabilities of a cryptogram from the set of keys and messages that are assumed to be associated with that cryptogram. The cryptosystem which provides less posteri probabilities from the apriori knowledge is treated as a strong cryptosystem.

The security of a cryptosystem is considered to be based on complexity theory [2]. The complexity of a cryptosystem is either algorithmic complexity or key complexity. When a cryptosystem is complex, it takes more time for the cryptanalyst to break it. The system can then be considered as secure till that time i.e. computationally infeasible. Security of these systems lies in the fact that brute force attack becomes impractical with the increased size of the key and when the algorithm is complex. It should be noted here that there are methods available which take less time than the exhaustive key search [5]. Linear and differential cryptanalysis on various cryptosystems proved that the success rate of breaking ciphers is increasing with the improved computational speed [6].

1.2. Cryptographic Strength Estimation Review

Cryptosystems are considered to be secure by the fact that the construction of S-boxes is non-linear in nature [7]. Under the additional hypothesis that these S-boxes constitute overdefined system of algebraic equations, XSL attack was performed on Serpent and it was proved that the security of ciphers does not grow exponentially with the number of rounds [8]. Way back in 1977, it was suggested that an exhaustive key search on parallel machines can break the NBS data encryption standard. The algorithmic complexity of a cryptosystem is achieved by iterating a weaker function in various rounds. It is later proved that DES can be reduced to 8 rounds and can be broken in less than 2 minutes and the complexity does not grow exponentially [12]. If DES is reduced to 15 rounds, then it is breakable faster than exhaustive search. Differential cryptanalysis was successful on FEAL cryptographic algorithm also [13]. The attack is based on chosen plaintext attack where cryptanalyst has a bunch of plaintexts to apply on a known algorithm and obtain corresponding ciphertexts. It is a black box concept where the cryptanalyst chooses two particular plaintexts whose differential value is known and calculates differential value of the corresponding ciphertexts. After analyzing several such plaintext–ciphertext combinations, a relation is derived between plaintext – ciphertext pair, there by exploiting the complexity of the algorithm. The security of optical cryptosystems is also exploited using known plaintext attack [14]. IDEA cryptosystem was also broken using an advanced differential attack, namely narrow biclique method which uses meet-in-the middle attack [15]. Though the complexity of this cryptanalysis is more, it is proved that even the complex algorithms can be broken using advanced computational systems. In 2012, cryptanalysis was demonstrated on full AES also [16]. AES is successfully cryptanalysed within 8 rounds. Though cryptanalysis of higher rounds was not successful for AES, 9th and 10th round cryptanalysis was successfully performed on AES like systems [17].

Many researchers formulated various cryptanalysis methods over the years and are successful in breaking ciphers up to few rounds. UCL crypto group has summarized almost all the cryptanalysis methods available during 1990s [41]. If the security of a cryptosystem lies in the nonlinearity of its algorithm, S box more specifically, then identifying the near linearness is the key factor of linear cryptanalysis. In 1994, linear cryptanalysis method was proposed for DES [18]. It is performed on SPECK [19], reduced round SIMON [20], FEAL [21], Serpent [22], PRESENT [23] and AES [24] algorithms in recent years. Biclique cryptanalysis is also demonstrated on MIBS□80 and PRESENT□80 block ciphers [25] and AES [26] algorithm. It is

reported that the vulnerability of DES to bruteforce attack can be overcome by the 64 bit key cryptosystem, Blowfish [27]. Several investigations are performed and suitable key lengths are proposed for various cryptosystems [28]. Later, the use of biometric [29], face hashing [31], face recognition [32], hand written signature [33] and voice [30] based key generation was suggested by several authors. But with the known plaintext attack, the information about the key can be obtained. The cryptanalyst obtains several plaintext-ciphertext pairs which are generated using the same key. On the basis of this apriori knowledge, the key is determined for use in reading later cryptograms for which he need not know the plaintext. The key factor in complexity of increased key length is that the time taken to apply all possible keys becomes infeasible when the key length is increased. But, the interesting fact is that the cryptanalyst actually needs the sub keys, but not the actual key to break the cipher. Frequent changing of the key is considered to increase the security. The security may be increased by frequent changing of the password but the advantage is small [39]. This is because, even in exhaustive search, the attacker is expected to have a successful guess halfway only through the search i.e. if the total number of keys is K, then only K/2 keys are sufficient to guess the actual key. Key recovery was practically performed on 10 rounds AES algorithm [40], which is considered to be the strongest cryptosystem.

Linear cryptanalysis is a known plaintext attack that uses linear relation between inputs and outputs of an encryption algorithm that holds with certain probability. This approximation can be used to assign probabilities to the possible keys and locate the most probable one. Once linear approximations of the S-boxes are found, the problem is to find a way to combine those individual approximations to establish a final approximation of the cipher that involves plaintext bits, ciphertext bits and key bits only.

Differential cryptanalysis is another powerful method which analyses the differences of cyphertext pairs resulted from plaintext pairs of particular difference. Linear transformations like bit permutations, key additions etc cause differences between two texts. For an S box with n input bits and m output bits, 2n input combinations are possible and for every input difference, getting 2m output differences is an apriori knowledge to the crypt analyst. Cryptanalyst examines all these possible differences and tabulates the number of occurrences of each difference. These probabilities are used in identifying the original key. Differential cryptanalysis also requires r-1 rounds to be analyzed. Let Δp is the plaintext difference and Δc is the resultant ciphertext difference. If any plaintext pair with Δp probability provides a ciphertext pair with Δc probability after r-1 rounds, then that pair is called right pair. These right pairs are used in analyzing the behavior of algorithm for multiple plaintexts in order to break it. As a counter measure to differential cryptanalysis, the differential propagation inside an S-box is maintained as low as possible and the number of S-boxes is maintained as high as possible.

Differential-linear cryptanalysis is a chosen plaintext attack where the linear cryptanalysis is used to provide the differential characteristics of the cipher. An 8 round attack against DES recovers 10 bits of key with 512 chosen plaintexts. However, expanding the attack to higher number of rounds is not found in literature. The strength of a cryptosystem is also estimated using side channel attacks like power consumption [34] and throughput [35] of the system. Throughput of a system lies in the algorithmic complexity which can be exploited using linear and differential cryptanalysis. With the decreased power consumption rates in lightweight cryptographic algorithms, the strength comparison requires a new measure than power.

All these attacks, and many more attacks introduced thereafter, are applied on various ciphers either to analyze their behaviour or for key recovery. Increasing number of rounds, increasing size of the key, building more complex S-box structures, using more S-boxes etc. are considered to be counter measures for all these attacks. These counter measures are based on increasing time to analyze a cryptosystem. All these counter measures are actually adding more complexity to the algorithm and consuming more power and space. This increased complexity results in reduced

speed and throughput of a block cipher. The message is also considered as a mere bunch of 0`s and 1`s. The information contained in the message is ignored. The security that can be provided by the message itself is another important aspect that is yet to be taken care of.

2. CONCEPT OF SPURIOUS KEYS

Many researchers believed that it is important to create uncertainty in original data in order to make it secure [42]. Purpose of S-boxes is for meeting this requirement of creating uncertainty. It is achieved by a principle that if an S-box is complete then its inverse is not complete. It is therefore believed that the key cannot be discovered by applying the inverse of an S-box.

Every message in general posses some amount of apriori knowledge. This apriori knowledge is useful to cryptanalyst. When an encrypted message is decoded using various keys, most of the keys may decrypt the cryptogram in such a way that it can be easily eliminated as a wrong key. The easy elimination of such keys is possible due to the fact that they contain gibberish (non-text characters). The cryptanalyst knows that the message cannot contain non-text characters. Some of the keys will result in text like message after decryption. However, the resultant text like message may or may not posses any meaning. All these keys are the probable keys. One of them is actual key. These keys introduce ambiguity to cryptanalyst and are called Spurious Keys.

Spurious key analysis is not applied in historical measures for cryptographic strength estimation. But the role of redundancy and unicity distance in the non-linearity of a language was studied as a separate entity. The study presented in this paper combines these language aspects with security measure which results in spurious key analysis. Present study concentrates on establishing spurious keys as a vital component for cryptographic strength estimation. Efforts are made to investigate and validate the suitability of this measure. However, this is just an alternative to existing methods when plaintext and ciphertext are considered as mere bit streams. But these texts contain crucial information of the data being sent or received. If that information content is taken into consideration, then this analysis is vital for strength estimation.

2.1. Spurious Key Model

Assuming that the original message needs to be decrypted is a text, by applying a ciphertext only attack on the cryptosystem; all the decryptions do not provide a text-like-text as the output. This provides a scope for the cryptanalyst to eliminate all those decryptions which do not look like text. The search for original decryption can now be narrowed. But, in the process of identifying the original message, the cryptanalyst meets certain set of keys which provide meaningful output. These keys are spurious keys and are not possible to eliminate without knowing some extra information about the message. The primary goal of generating spurious keys is to identify the strength of a specific algorithm being used. The more the spurious keys, the more difficult it is to identify the actual key. All these keys are equiprobable keys where any one of them can be the actual key. The cryptanalyst applies the keys until the satisfactory results are obtained. As all these keys provide a text-like-text as the output, fetching only one key among them is a challenging task. In order to understand the cryptanalyst's perspective, a simulation model can be designed for generation of spurious keys.

The process of generating spurious keys involves two major steps, one is encrypting a message using a specific key and the other is decrypting the ciphertext using random keys for identifying spurious keys based on the decryption. A spurious key generation algorithm is proposed with an assumption that the plaintext is a meaningful text. The flow chart for spurious key generation algorithm is given in Figure 1.

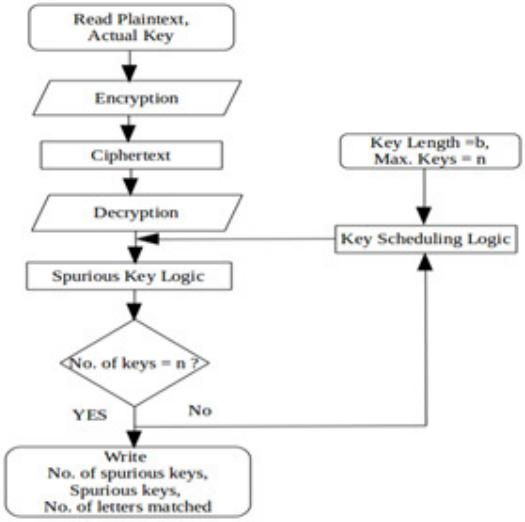

Figure 1. Spurious key generation algorithm

Initially, a ciphertext is obtained by applying the plaintext and a key to the encryption algorithm. The plaintext can be retrieved back by applying the ciphertext and the key to decryption algorithm. A single key is required to obtain the ciphertext from the plaintext, using any cryptographic algorithm. This ciphertext is assumed to be available to the cryptanalyst. The cryptanalyst tries to apply various keys to identify the original message. To do so, the cryptanalyst makes some assumptions like the text might belong to a specific language, the content might be related to a specific task etc.

The encryption and decryption blocks can be of any cryptosystem, like AES, DES, Blowfish, ARC4 etc. These models can be implemented in any programming language like C, Python etc. In this method, encryption is carried out only once, using the actual key. But decryption is performed continuously using various keys. Each key will provide decrypted text which may or may not be a meaningful text. The process of applying various keys is considered as a specific block called key scheduling logic. Another block is called spurious key logic. It is used for identifying whether a decrypted message is text-like-text or not. As the cryptanalyst does not know the actual key being used, he may try to apply various expected keys depending on apriori probabilities. This process can be continued for a predefined time period or for all the possible keys in the range of key size of the algorithm being used. The key scheduling logic is an important block in the process of generating spurious keys. The main objective of this block is to generate keys continuously with the key size specific to the encryption algorithm used.

Due to the limitations in computational capabilities, it is difficult to apply all possible keys to the algorithm and record the spurious keys. Application of all possible keys depends on the key size of the algorithm and the computational capability of the system being used for simulation. For example, applying all possible keys present in the entire key space to AES algorithm is not feasible on a single computer even with the present day high speed computers. Hence, a specific method is required for scheduling the keys. This is to be done with an upper limit on the number of keys. Key scheduling can be done until a specific time period (or) for a specific number of keys. In the first method, the keys are generated and applied to the decryption algorithm for a specific time period and the number of spurious keys is calculated. In the second method, the keys are generated for a specific key space without considering the time.

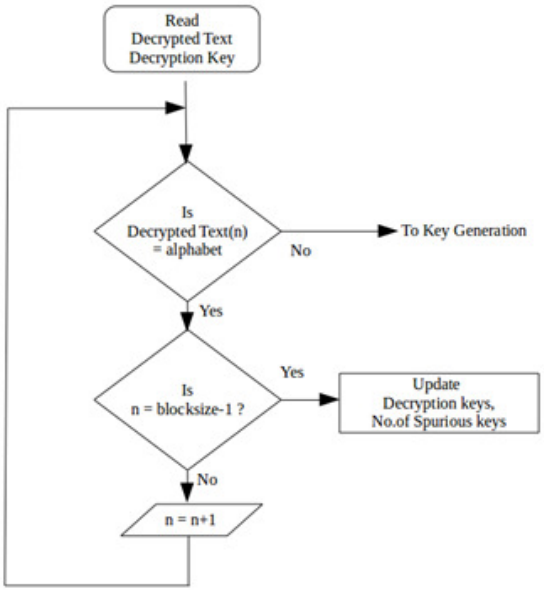

Figure 2. Flow chart for Spurious Key Logic

The logic involved in generating the spurious keys is shown in Figure 2. Spurious keys are those keys which generate a text like text. For example, upon applying a specific key, if the decrypted result is "hu□d\□]I", then the cryptanalyst can easily identify that the applied key is not the correct one. The cryptanalyst can eliminate such keys and narrow the process. The key space is defined by cryptanalyst depending on apriori probabilities i.e. the knowledge about the algorithm, language, process etc which can be predicted without actually knowing the actual plaintext, key etc,. The entire process can be continued for a predefined time period or for all the possible keys in the range of key size of the algorithm being used. The main objective of key scheduling block is to generate keys continuously with the key size specific to the encryption algorithm used. All possible numeric keys, i.e. 00000000 to 99999999 are generated using key scheduling logic block which is a simple 8 byte counter in this case.

As the number of spurious keys increases, the process of cryptanalysis becomes difficult. In turn, spurious keys increase security of the algorithm. The total message space contains numeric digits, alphabets and some special characters. Each message data is uniquely mapped to a separate ciphertext data. Due to this one-to-one mapping, a specific ciphertext transforms into the original message when decrypted using the actual key. The interesting fact is that the entire ciphertext space is not limited to the size of message text space i.e. there are many ciphertexts beyond the entire volume of message space. Due to this specificity in any cryptogram, the suspected keys can be reduced to a limited volume i.e. all those keys which provide gibberish can be easily eliminated from the expected key space. This is where the cryptanalyst has an advantage of limiting his search in order to break the code. When compared to exhaustive key search, the computational effort is reduced if the number of spurious keys is less. The strength of a cryptogram can hence be estimated based on its ability to produce the number of spurious keys. If all the keys are spurious then the system is a perfect secrecy system.

2.2 Limitations of Study

Though the paper is intended to establish spurious key analysis as a valuable parameter in estimating the strength of cryptosystems, the study is performed with certain limitations. The major limitation is in terms of key space. Though 264 keys are possible, entire key space is not

used for computational convenience. The algorithms used in this study accepts any ASCII character as key, but only 10^8 disjoint numeric keys are considered. Message size is also limited because of the nature of block cipher where they convert a message of any size into blocks of fixed size. Though, there are several cryptosystems available, only 64 bit block ciphers are considered for evaluation. Another important limitation is that the study is carried out with a presumption that the information content is also important in a cryptosystem.

3. EXPERIMENTAL RESULTS

All evaluations are performed on Intel® Core™ i7-4770 CPU @ 3.40GHz × 8 processor. The python cryptography toolkit is used for validations. Evaluation is carried out with the help of a common key, while varying the message text. Test samples of 50 varied text messages of 4 characters, 6 characters and 8 characters are considered for this evaluation. These 50 samples are taken at random from the dictionary. They hold no specific relation among themselves. The effect of message text size on each block cipher is also taken into account. Encryption is carried out with the key word '12345678'. Even though numeric key is considered, it was found that any key in the key space has the same effect. Decryption is performed on the ciphertext generated from the encryption process using 10^8 key combinations generated from key space of numeric keys of length 64 bits. While performing brute force, the decrypted message text is compared with the text space consisting of English alphabets. If the decrypted message text consists of any symbol other than English alphabets, then the corresponding key is considered as non spurious key. All spurious keys in the range of key space as defined, are listed out as spurious keys. The number of spurious keys observed for each message text is presented in the following section.

A sample spurious key for four algorithms, viz. DES, ARC2, Blowfish, CAST, with the respective text message as well as the decrypted text like message are listed in the table below. The plaintext messages are also listed in the Table 1.

Table 1. Sample spurious keys for English text

Algorithm	Plaintext	Ciphertext	Text like Message	Spurious Key
DES	alkaline	□d*□□W□□	xcKOwAVl	002c4e02
ARC2	duck	e2QwfoIOD2M=	QuDIlXoO	01189996
Blowfish	computer	W1x0/2DbLMQ=	mwethwzo	00177391
CAST	abacus	LQYdQMLqVU8=	MqgfDgHh	00124874

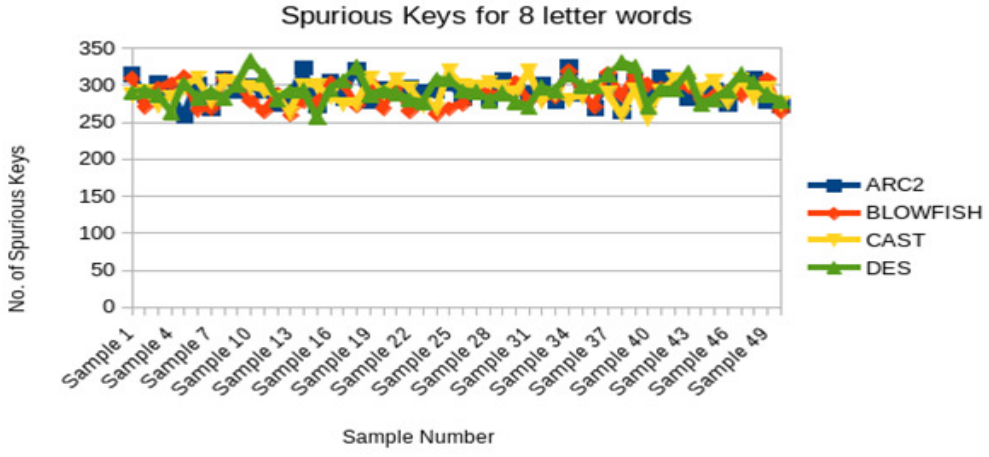

Figure 3. Spurious key analysis for 8 characters English Message

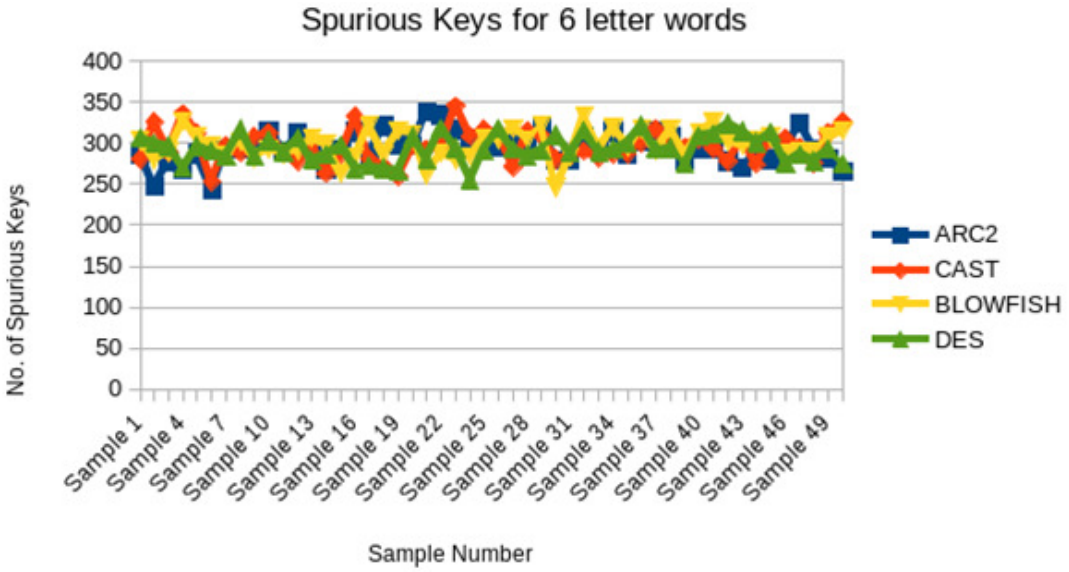

Figure 4. Spurious key analysis for 6 characters English Message

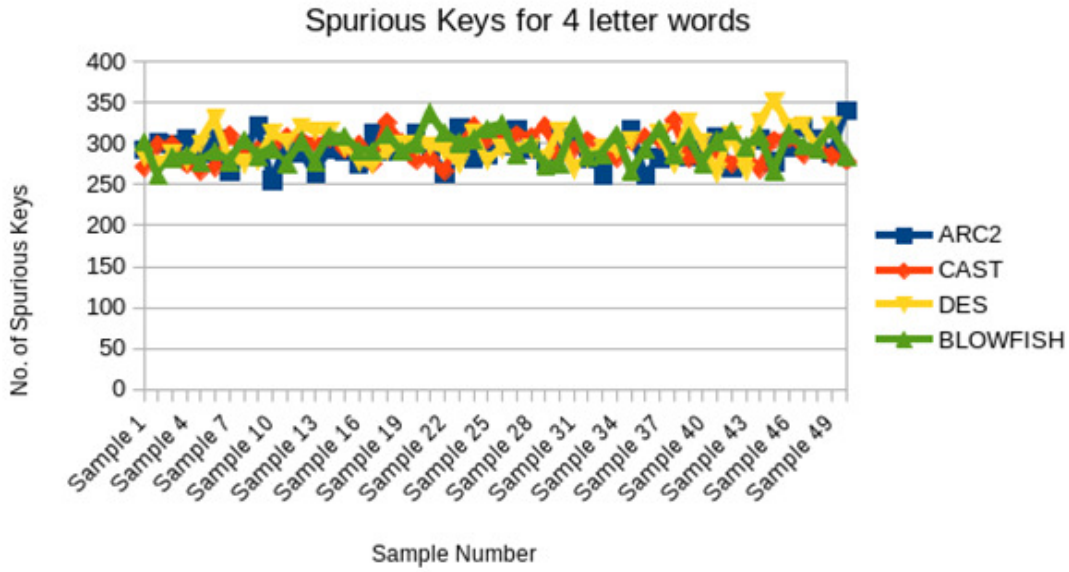

Figure 5. Spurious key analysis for 4 characters English Message

Comparative analysis of the entire evaluation is presented in the Table 2. The number of spurious keys is approximately same for each algorithm due to the fact that the message text is also same. The key length and message lengths are bounded by the block cipher principles based on which the spurious keys exhibit similar characteristic on all ciphers considered in this evaluation.

Table 2. The average number of spurious keys for English text

	ARC2	**CAST**	**BLOWFISH**	**DES**
4 LETTER	292.86	294.18	295.82	296.74
6 LETTER	294.56	296.22	298.2	293.2
8 LETTER	291.78	290.62	287.5	293.82

4. PERFORMANCE ANALYSIS OF BLOCK CIPHERS FOR DEVANAGARI SCRIPT

The spurious key analysis is performed for Devanagari Script also. Devanagari is a south and central Asian language described by Unicode along with Bengali, Gurumukhi, Gujarati, Oriya, Tamil, Telugu, Kannada and Malayalam scripts. This script is represented with U+0900 to U+097F. The characters that are likely to be present in a message are considered for the analysis, i.e. '□ $ॅ $ं $ः ऄ अ आ इ ई उ ऊ ऋ ऌ ऍ ऎ ए ऐ ऑ ऒ ओ औ क ख ग घ ङ च छ ज झ ञ ट ठ ड ढ ण त थ द ध न ऩ प फ ब भ म य र ऱ ल ळ ऴ व श ष स \ ह $□ $□ $़ s $ा $ि $ी $ु $ू $ृ $ॄ $ॅ $ॆ $े $ै $ॉ $ॊ $ो $ौ $् $□ $□ ॐ $ॎ $ॏ $□ $□ $□ क़ ख़ ग़ ज़ ड़ ढ़ फ़ य़ ऋ ऌ $ॢ $ॣ । ॥ ०१२ ३ ४ ५ ६ ७ ८ ९ ॰ ॱ ॲ □ □ □ □ □ □ ग़ ज़ ? ड़ ब़ '. The $ symbol may be ignored in the above character set. Unlike English language analysis, the numbers and some special characters are also considered in this analysis.

Table 3 presents some sample spurious keys for selected plaintexts with key as '12345678'. The resultant ciphertext and the decrypted text-like-text are also given in the table.

Table 3. Sample spurious keys for Devanagari script

Algorithm	Plaintext	Ciphertext	Text like Message	Spurious Key
DES	पक्षाघात	□□o&□Ţi	मंड़इक्ष□ऴ	0000202g
ARC2	अमीर	h□□mXW~\	तपछधइ□ि॔	00000043
CAST	अख़बार	□□#□#□¢	०ऐबऊकऑक़	00000588
Blowfish	तर्जुमान	□'g□□@□□	ढ़□क़बयग़िड	00000106

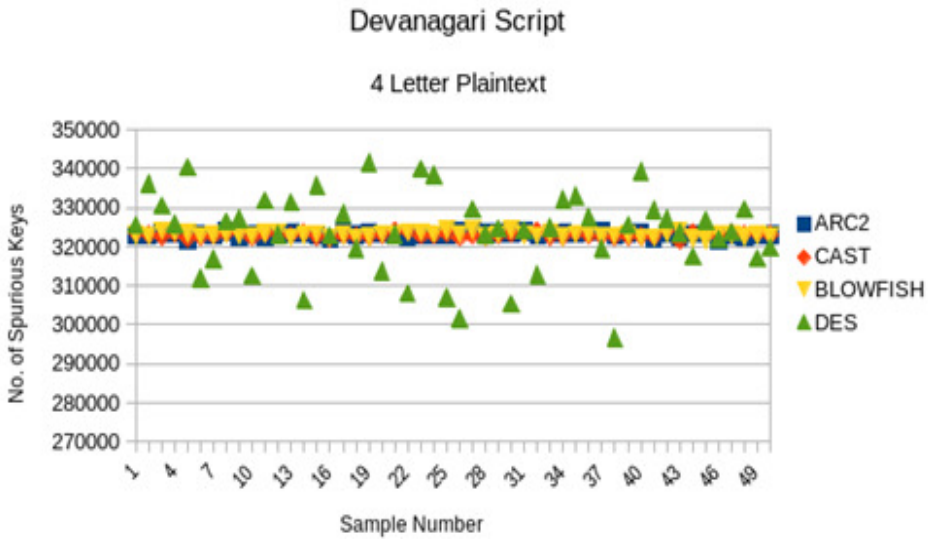

Figure 6. Spurious Keys for 4 Character Devanagari Plain-text

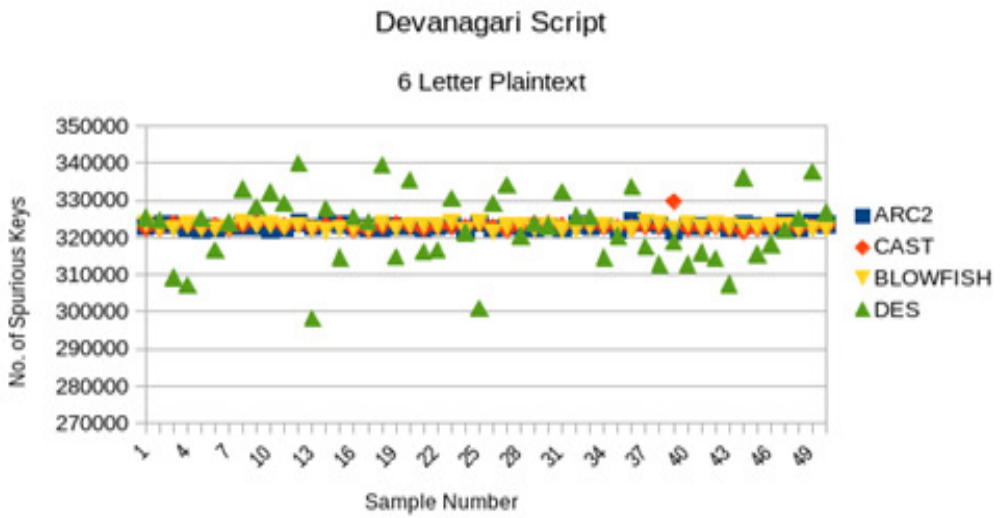

Figure 7. Spurious Keys for 6 Character Devanagari Plain-text

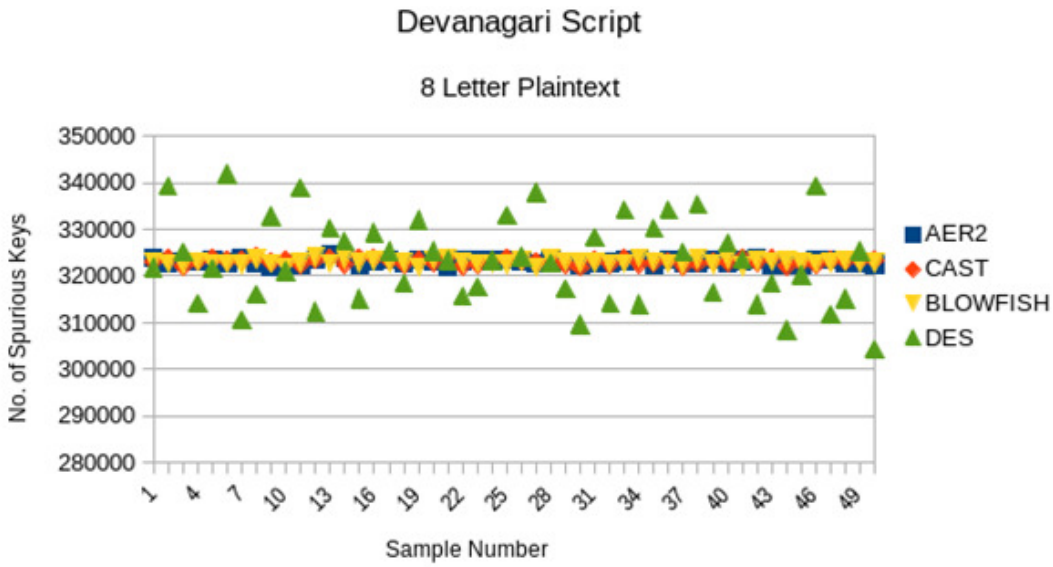

Figure 8. Spurious Keys for 8 Character Devanagari Plain-text

The average number of spurious keys for Devanagari script is summarized in table below. It can be observed that all the four block ciphers behave in similar manner for a given set of plaintexts.

Table 4. The average number of spurious keys for English text

	ARC2	**CAST**	**BLOWFISH**	**DES**
4 LETTER	323231.56	323051.72	323211.2	323007.04
6 LETTER	323064.44	323172.56	322964.58	323227.28
8 LETTER	323113.74	323111.14	323001.86	323122.3

5. CONCLUSIONS

Strength of the cryptosystem is generally identified with the evaluation of weaknesses in terms of algorithm and key. This approach is adopted by many researchers. However, the strength of the cryptosystem should also be identified with the associated strong parameters. An attempt is made in this paper to evaluate the strength of the cryptosystem with the help of spurious keys approach which is termed as strength because there is an inherent confusion arises out of the nature of spurious keys. During decryption, spurious key results in text-like message from the given ciphertext. In an ideal scenario, if every key in the key space acts like a spurious key then the cryptosystem will provide maximum strength which is not possible in real world.

In the present work, a sub set of key space is considered for evaluation of number of spurious keys in that range. Four block ciphers viz ARC2, Blowfish, CAST and DES are considered for this evaluation. Message space is assumed to be alphabets of the respective script only. In this work, we considered English and Devanagari scripts. We evaluated the possible number of spurious keys associated with the sub key space. A comparison is made in terms of varying text size.

All four block ciphers of the present work posses similar characteristics for the varying text size within the limit of the block length. Devanagari script is found with approximately 323000 spurious keys whereas English messages are found with approximately 294 spurious keys in the sub key space of 10^8 keys. Due to the large number of spurious keys associated with Devanagari script, it is difficult to identify the exact key of the cryptosystem, which is considered to be the strength of the system associated with the message texts of Devanagari.

It is necessary to explore the impact of numerals and special characters in the message text on the possible number of spurious keys. The evaluation is in progress. Similarly, evaluation on other block ciphers and varying key length is considered as a future task. The impact of script on cryptosystem is another factor to be considered for evaluation as a future task.

REFERENCES

[1] C. E. Shannon, "Communication theory of secrecy systems", Bell Syst. Tech. J., vol. 28, pp. 656-715, Oct. 1949.

[2] Goldwasser Shafi and Silvio Micali, "Probabilistic encryption." Journal of computer and system sciences 28.2 (1984): 270-299.

[3] Petitcolas et al, "Information hiding-a survey." Proceedings of the IEEE 87.7 (1999): 1062-1078.

[4] Moulin Pierre and Joseph O'Sullivan, "Information-theoretic analysis of information hiding." Information Theory, IEEE Transactions on 49.3 (2003): 563-593.

[5] Diffie Whitfield and Martin E. Hellman, "Special feature exhaustive cryptanalysis of the NBS data encryption standard." Computer 6 (1977): 74-84.

[6] Langford Susan K and Martin E. Hellman, "Differential-linear cryptanalysis." Advances in Cryptology—CRYPTO'94. Springer Berlin Heidelberg, 1994.

[7] Nyberg Kaisa, "Perfect nonlinear S-boxes." Advances in Cryptology—EUROCRYPT'91. Springer Berlin Heidelberg, 1991.

[8] Courtois Nicolas T and Josef Pieprzyk, "Cryptanalysis of block ciphers with overdefined systems of equations." Advances in Cryptology—ASIACRYPT 2002. Springer Berlin Heidelberg, 2002. 267-287.

[9] Kaufman Lori M, "Data security in the world of cloud computing." Security & Privacy, IEEE 7.4 (2009): 61-64.

[10] Leach John, "Improving user security behaviour." Computers & Security 22.8 (2003): 685-692.

[11] Rajesh et al, "Genetic algorithmic approach for dynamic request processing in agent cloud platform." Advance Computing Conference (IACC), 2015 IEEE International IEEE, 2015.

[12] Biham Eli and Adi Shamir, "Differential cryptanalysis of DES-like cryptosystems." Journal of CRYPTOLOGY 4.1 (1991): 3-72.

[13] Biham Eli and Adi Shamir, "Differential cryptanalysis of Feal and N-hash." Advances in Cryptology—EUROCRYPT'91. Springer Berlin Heidelberg, 1991.

[14] Rajput Sudheesh K and Naveen K. Nishchal, "Known-plaintext attack on encryption domain independent optical asymmetric cryptosystem." Optics Communications 309 (2013): 231-235.

[15] Khovratovich et al, "Narrow-Bicliques: cryptanalysis of full IDEA." Advances in Cryptology–EUROCRYPT 2012. Springer Berlin Heidelberg, 2012. 392-410.

[16] Biryukov Alex and Johann Großschädl. "Cryptanalysis of the full AES using GPU-like special-purpose hardware." Fundamenta Informaticae 114.3-4 (2012): 221-237.

[17] Jean et al, "Improved cryptanalysis of AES-like permutations." Journal of Cryptology 27.4 (2014): 772-798.

[18] Matsui Mitsuru, "Linear cryptanalysis method for DES cipher." Advances in Cryptology—EUROCRYPT'93. Springer Berlin Heidelberg, 1994.

[19] Liu Yu et al, "Linear cryptanalysis of reduced-round SPECK." Information Processing Letters 116.3 (2016): 259-266.

[20] Yu Xiao-Li et al, "Zero-Correlation Linear Cryptanalysis of Reduced-Round SIMON." Journal of Computer Science and Technology 30.6 (2015): 1358-1369.

[21] Leurent Gaëtan, Differential and Linear Cryptanalysis of ARX with Partitioning--Application to FEAL and Chaskey. Cryptology ePrint Archive, Report 2015/968, 2015.

[22] Biham et al, "Linear cryptanalysis of reduced round Serpent." Fast Software Encryption. Springer Berlin Heidelberg, 2002.

[23] Cho Joo Yeon, "Linear cryptanalysis of reduced-round PRESENT." Topics in Cryptology-CT-RSA 2010. Springer Berlin Heidelberg, 2010. 302-317.

[24] Mansoori S. Davood and H. Khaleghei Bizaki, "On the vulnerability of simplified AES algorithm against linear cryptanalysis." Int. J. Comp. Sci. Network Security 7.7 (2007): 257-263.

[25] Sereshgi Faghihi et al, "Biclique cryptanalysis of MIBS-80 and PRESENT-80 block ciphers." Security and Communication Networks 9.1 (2016): 27-33.

[26] Bogdanov et al, "Biclique cryptanalysis of the full AES." Advances in Cryptology–ASIACRYPT 2011. Springer Berlin Heidelberg, 2011. 344-371.

[27] Schneier Bruce, "Description of a new variable-length key, 64-bit block cipher (Blowfish)." Fast Software Encryption. Springer Berlin Heidelberg, 1994.

[28] Lenstra et al, "Selecting cryptographic key sizes." Journal of cryptology 14.4 (2001): 255-293.

[29] Chang et al, "Biometrics-based cryptographic key generation." Multimedia and Expo, 2004. ICME'04. 2004 IEEE International Conference on. Vol. 3. IEEE, 2004.

[30] Monrose et al, "Cryptographic key generation from voice." Security and Privacy, 2001. S&P 2001. Proceedings. 2001 IEEE Symposium on. IEEE, 2001.

[31] Teoh et al, "Personalised cryptographic key generation based on FaceHashing." Computers & Security 23.7 (2004): 606-614.

[32] Chen B and Vinod Chandran, "Biometric based cryptographic key generation from faces." Digital Image Computing Techniques and Applications, 9th Biennial Conference of the Australian Pattern Recognition Society on. IEEE, 2007.

[33] Freire et al, "Cryptographic key generation using handwritten signature." Defense and Security Symposium. International Society for Optics and Photonics, 2006.

[34] Mittal Mohit, "Performance Evaluation of Cryptographic Algorithms." International Journal of Computer Applications (0975–8887) Volume (2012).

[35] Mandal Pratap Chandra, "Evaluation of performance of the Symmetric Key Algorithms: DES, 3DES, AES and Blowfish." Journal of Global Research in Computer Science 3.8 (2012): 67-70.

[36] Black et al, "Encryption-scheme security in the presence of key-dependent messages." Selected Areas in Cryptography. Vol. 2595. 2002.

[37] Halevi Shai and Hugo Krawczyk. "Security under key-dependent inputs." Proceedings of the 14th ACM conference on Computer and communications security. ACM, 2007.

[38] Hofheinz Dennis. "Circular Chosen-ciphertext Security with Compact ciphertexts." Eurocrypt. 2013.

[39] Chiasson Sonia, and Paul C. van Oorschot. "Quantifying the security advantage of password expiration policies." Designs, Codes and Cryptography (2015): 1-8.

[40] Biryukov et al, "Key recovery attacks of practical complexity on AES-256 variants with up to 10 rounds." Advances in Cryptology–EUROCRYPT 2010. Springer Berlin Heidelberg, 2010. 299-319.

[41] Standaert et al, "Cryptanalysis of block ciphers: A survey." UCL Crypto Group (2003).

[42] Hussain Iqtadar and Tariq Shah, "Literature survey on nonlinear components and chaotic nonlinear components of block ciphers." Nonlinear Dynamics 74.4 (2013): 869-904.

[43] Braun et al, "Long term confidentiality: a survey." Designs, Codes and Cryptography 71.3 (2014): 459-478.

[44] Alléaume et al, "Using quantum key distribution for cryptographic purposes: A survey." Theoretical Computer Science 560 (2014): 62-81.

[45] Kong et al, "A comprehensive survey of modern symmetric cryptographic solutions for resource constrained environments." Journal of Network and Computer Applications 49 (2015): 15-50.

[46] Korner, "Compressing inconsistent data." Information Theory, IEEE Transactions on 40.3 (1994): 706-715.

[47] Witsenhausen Hans S, "The zero-error side information problem and chromatic numbers (corresp.)." Information Theory, IEEE Transactions on 22.5 (1976): 592-593.

[48] C. E. Shannon, "The zero-error capacity of a noisy channel," IRE Trans. Inform. Theory, vol. IT-2, pp. 8–19, 1956.

AUTHORS

Mr. Mekala Ramarao is a full time research scholar at JNTUH College of Engineering Hyderabad. He is a life time member of ISTE. His areas of interest include language based security, information theoretic security, embedded security and block chain technology. He is awarded Junior Research Fellowship (JRF) from University Grants Commission (UGC) under Rajiv Gandhi National Fellowship (RGNF) scheme.

Dr. L. Pratap Reddy received, B.E. degree from Andhra University (INDIA) in Electronics and Communication Engineering in 1985, the M.Tech. degree in Electronic Instrumentation from Regional Engineering College (WARANGAL) in 1988 and the Ph.D. degree from Jawaharlal Nehru Technological University (HYDERABAD) in 2001. From 1988 to 1990 he was lecturer in ECE Department of Bangalore Institute of Technology (BANGALORE), from 1991 to 2005 he was faculty member in JNTU College of Engineering (KAKINADA). Since 2006 he is with Department of Electronics and Communication Engineering at JNTU, Hyderabad. His current activity in research and development includes, apart from telecommunication engineering subjects, Pattern Recognition, Information Security, Embedded Systems and Linguistic processing of languages. He published 75 technical papers, articles and reports. He is active member in professional bodies like IETE, ISTE, IE, and CSI. At present he is Working Chairman of SWECHA and Treasurer of Free Software Movement of India.

BHVS Narayana Murthy, is the Director, Research Centre Imarat (RCI), a Defence R&D Organization (DRDO) laboratory. He is the fellow of INAE and IETE. He is a senior member of IEEE and Life Member of AeSI, CSI. His areas of interest include embedded systems, System on Chip, Re-configurable Intelligent systems, Sensor networks and Machine learning.

Maruti Sairam V Annaluru, is working as scientist in Research Centre Imarat(RCI), a Defence R&D Organization (DRDO) laboratory. He is life member of CSI. He is working on the design and development of Real-time Embedded avionics systems software. His areas of interest includes Safe programming, Model based software development, Information security in embedded systems.