FUNDAMENTALS OF CRYPTANALYSIS: ATTACKING STREAM CIPHERS

A.Mahendar[1], Dr. E. Srinivas Reddy[2],

[1]*Research scholar, Department of Computer Science & Engineering, Acharya Nagarjuna University, AP, India,* ***mahi.adapa@gmail.com***

[2] *Professors, Department of Computer Science & Engineering, Acharya Nagarjuna University. AP, India,* ***edara_67@yahoo.com.***

Abstract

This article contains an elementary introduction to the cryptanalysis of stream ciphers. Initially, a few historical examples are given to explain the core aspects of cryptography and the various properties of stream ciphers. We define the meaning of cryptographic strength and show how to identify weaknesses in a cryptosystem. Then, we show how these cryptographic weaknesses can be exploited and attacked by a number of cryptanalytic techniques. The academic literature includes many articles there use complex mathematical notations to represent trivial cryptanalytic techniques. Contrarily, this paper tries to put forward the most commonly used cryptanalytic techniques by using only simplistic and comprehensible examples. The addressed techniques are used in several scientific articles to mount practical attacks on real cryptosystems.

Index Terms: *Cryptography, Stream Cipher, Cryptanalysis, Cryptanalysis Techniques.*

---------- *** ----------

1. INTROCTION

Cryptography is a complex and mathematically challenging field of study. It involves taking some data or message and obfuscating it so that it is unreadable by parties that the message was not intended. Before the message becomes encrypted it is referred to as the plain text. Once a message becomes encrypted it is then referred to as the cipher text. The study of cipher text in an attempt to restore the message to plaintext is known as cryptanalysis. Cryptanalysis is equally mathematically challenging and complex as cryptography. Because of the complexity involved with Cryptanalysis work this document is only focused on the basic techniques needed to decipher mono-alphabetic encryption ciphers and cryptograms.

The only application referenced in this document is the CRyptoANalysis ToolKit (CRANK). This program can be found at http://crank.sourceforge.net/ [1]. A basic understanding of cryptanalysis is essential to appreciating the complexities of a good cryptographic algorithm. For example a manager of a software company or someone who is involved with code auditing would find it is essential that good well tested algorithms are used instead of a weak in house cipher. This paper will give you the basic tools necessary to begin a rudimentary examination of a cipher.

1.1 Cryptographic strength

The strength of a cryptographic algorithm is expressed in the total amount of computations an adversary needs to perform to recover the secret key. It is often referred to as the computational complexity of the cipher.
For a perfectly secure cipher, the computational complexity is the same as the key space, sometimes referred to as the entropy of the key. The key space refers to the set of all possible keys and represents the total number of combinations using all secret key bits. The size of the key (key-size) is the amount of bits n which define the size of the complete key space 2n.

The naive method to recover the secret key is to try simply all combinations. Such methodology is often referred to as an exhaustive search or a brute-force attack. It would most likely require almost 2n computations to determine the secret key. To be precise, on average an adversary finds the secret key halfway. When lucky, the key can be determined at an early stage, but it might just as well be one of the last tries. Interestingly, this property already shows that the number of computations required determining the key is by definition lower than the actual key space.

A cipher that is perfect should be the base for a secure cryptosystem. In practice how- ever, most ciphers are (slightly) weaker than the full entropy of their secret key. With clever optimizations it is often possible to find the secret key by doing far fewer computations than the actual entropy would require. This is called the actual attack complexity of a cipher.

2. DEFINITION OF TERMINOLOGY

This section will define several terms as well as give a brief introduction into cryptography. A term used specifically for cryptanalysis is called *known text*. Known text is when there is an encrypted message and a known corresponding plaintext. This may not be the whole message but perhaps a section of the message, e.g. every message sent ends with the plaintext letters "EOT". By using cipher text with known text you can attempt to deduce the complete key used to encrypt all messages, which will greatly facilitate future deciphering [2].

There are several basic methods that can be used to encrypt a message. One method is called a *transpotional cipher*. This cipher only changes the order of the plaintext within the message, e.g. "LEAVE AT NOON" might become "EVAELTANOON". Another method is known as a *substitional cipher* . This method exchanges the characters in the plaintext with other characters defined by a *key* [3]. The key is the mapping of characters from the plain text to the cipher text as in the following:

ABCDEFGHIJKLMNOPQRSTUVWXYZ
zyxwvutsrqponmlkjihgfedcba

Using the same message from the above example this key would produce the following message: "OVZEVZGMLLM". This method of substitution is known as a Monoalphabetic Unilateral Substitution cipher. This term implies that for each letter in a plaintext message there is only one equivalent cipher character. (Note: The majority of this document will focus on these types of cipher systems. Monoalphabetic Unilateral Substitution systems will simply be referred to as a substitution cipher for the sake of clarity and brevity.)

3. STREAM CIPHERS

A stream cipher performs an encryption which is similar to the One-time Pad (OTP) en- crypt ion technique. It produces a large chunk of secret, random looking data and combines it with the plaintext to produce cipher text. Without the exact same data chunk, the plaintext cannot be uncovered from the cipher text. The random data represents a stream of bits which is derived from the secret key and is commonly referred to as key stream. A stream cipher contains some persistent memory, called the internal cipher state, which is initialized by the secret key and propagates to a successor state after each encryption step. The output of a strong stream cipher is comparable to (and should be indistinguishable from) a contiguous bit stream produced by a Pseudo Random Number Generator (PRNG).
To be more precise, we embed the remarks made in [2] and define a stream cipher as follows: an encryption function which operates on individual plaintext digits (usually bits) where its internal state is initialized with the secret key prior to encryption. The key stream varies, depending on the initialized secret key and the moment of encryption with respect to the propagation of the internal state. Encryption of plaintext and decryption of cipher text are both performed by the exclusive-or (XOR) operation, which is denoted by a $\oplus$ symbol and represents a bit-wise addition modulo two. A useful mathematical property of this operator is that it can be inverted. Therefore, it can be applied for encryption as well as decryption.

There are two types of stream ciphers, synchronous and self-synchronizing. In a synchronous stream cipher, the encryption bits are computed independently from the plaintext. Such ciphers are useful in situations when a communication channel is more prone to error. It might happen that just one badly transmitted bit is wrongly interpreted, which however does not directly affect the other bits that were transferred in a correct manner. Therefore, stream ciphers are very useful to encrypt streaming media where the speed of data-traffic is more important than the completeness and integrity of the data. Contrarily, a self-synchronizing stream cipher computes the successor of its internal state with a function over the previous state and the ciphertext. The internal state diverts from its original propagation path when a transmission error occurs. This dissertation focuses itself on the most widely used and best studied of the two, the synchronous stream ciphers. Therefore, a general reference to a stream cipher refers to a synchronous stream cipher.

4. BASIC CRYPTANALYSIS TECHNIQUES

One good method for solving basic substitution ciphers is with frequency counts. A frequency count can be conducted on a cipher to learn what the most and least common characters are in the cipher. The most common letters in the English language are E,T,N,R,O,A,I and S. These eight characters make up around 67% of the words in the English language. Vowels, A,E,I,O, and U make up around 40% of English text. The frequency may vary depending on what the plaintext is. For example, if the message is source code it will use many more symbols than a message that is just written in English. If

you conduct a frequency count of this paragraph your result0s6wEo4uld be: E, T, A, O, and S [3].

As you can see the results are not exactly the same. This is because the there are approximately 500 characters in the above paragraph. If you use a sample of 1000 characters or more your results will become more accurate. The frequency count of a single character is referred to as a *Unigraph* [4]. If the frequency of cipher text is actually the same as plaintext then the encoded method is actually a transpositional cipher instead of a substitution cipher. Consider the following example.

Plaintext:

IF WE DO NOT PROPERLY PROTECT THE USERS DATA WE CAN SIMPLY HIDE BEHIND THE DMCA IF SOMEONE NOTICES!!

Transpositional:

I OOYFDTP O EPW PRRENRLOTTSDWEHEAECERT T SAC U ANPIE LDHTSYEIHI NEMHBD DIM CMFENEC OOSASNT! OEI!

Substitution:

RU DV WL MLG KILKVIOB KILGVXG GSV FHVIH WZGZ DV XZM HRNKOB SRWV YVSRMW GSV WNXZ RU HLNVLMV MLGRXVH!!

Top 5 Unigraph Frequency counts:
Plaintext: E, O, T, I and D
Transpositional: E, O, T, I and D
Substitution: V, G, L, R and H

Even though the transpositional cipher is a small sample, it has the top 4 letters used in plaintext with E being the highest.
When dealing with a substitution cipher you should check the frequency of letters and their adjacent letters as well. A pair of letters together is referred to as a *Digraph*. The common digraphs in the English language are, TH, HE, EN, RE and ER. There are also *Trigraphs* that consist of frequency of three letters next to each other THE, ING, CON, ENT and ERE.

5. CRYPTANALYSIS TECHNIQUES

This section introduces fundamental cryptanalytic techniques which are used in cryptographic attacks, also referred to as cryptanalysis.

- Exhaustive Key Search Attack
- Side Channel Analysis Attack
- Time Memory Trade off Attacks
- Distinguishing Attacks
- Algebraic Attack
- Correlation attacks
- Guess and Determine attacks
- Linear Masking attacks
- Related Key Attack
- Divide and Conquer Attack

5.1. Exhaustive Key Search Attack

The exhaustive search attack is also called brute force. The method of this attack is to search through all possible states, checking for a match between the resulting and the observed keystream. Fortunately, Babbage [5] in 1995 improved the exhaustive search attack in stream ciphers. He defined two attacks in this area.

In the first attack, the attacker first produces a list of n-bit subsequences, sorted in lexographic (or numberic) order. Then the attacker select a random candidate state in this list and check, if the selected state produces the output of cipher, then the attacker found the initial state else he continues try to find the initial state [5].

The second attack was defined by Babbage [5] as:
" Let V be a vector space of dimension n over GF(2), with each possible KG (Keystream Generator) state an element of V. The initial state, which we wish to determine, is s0, and the state transition function is linear, and so can be represented by an n n matrix A, so that si =s0Ai. The output function h : V !GF(2), so that the ith keystream bit ki is equal to h(si)." [5].

5.2. Side Channel Analysis Attack

Generally there are two steps involved in developing any cryptographic primitive. First, it is defined as an abstract mathematical object. Thereafter this mathematical entity needs to be implemented in form of a program and in some cases these programs are further implemented in some specific hardware. These programs after implementation will be executed in a computing environment on processing units. These executions will present some specific characteristics.
Side channel Analysis (SCA) refers to the attacks based on the physically observable characteristics during execution. Some of the common physical characteristics that are used for

Side Channel Analysis are Power and Microprocessor time required for execution, electromagnetic radiation, heat dissipation and noise of the system etc.
On the basis of above characteristics; there are different Side Channel attacks on ciphers in general and on stream cipher in particular. Some of powerful techniques, that generally used for Side Channel Analysis attacks are Simple Power analysis attack, Differential Power Analysis attack [2, 3], Timing Analysis attack [4, 5], Electromagnetic Analysis attacks [6, 7,8] and Acoustic Cryptanalysis [9].

Though there is no general countermeasure to these attacks but some of the possible countermeasures maybe noise addition, buffering of the output sequence, Physical shielding, reduction of signal size, eliminating the branch processing in implemented algorithm that will make the encryption time equivalent and few more.

5.3. Time Memory Tradeoff Attacks

A time memory tradeoff attack is a method of cryptanalysis that aims to attack a cryptographic primitive with lower complexity than look up table and an online complexity lower than exhaustive key search. TMTO is an improvement to the exhaustive key search attack that trades off computational time against memory complexity [12].
This attack can be divided into two phases; an offline phase or pre-computation phase and online phase. In offline phase a table is constructed in like lookup table method by selecting different random keys and generating the output for each chosen key. These pairs of output strings and keys are stored in an indexed table by the output strings. In the second phase or online phase, the attacker observes the output generated by unknown keys. Then these outputs are matched with the outputs of the table generated in the offline phase. If a match is found then corresponding key will be the key off the matched output.
Amirazizi and Hellmen were the first to propose Time memory processor trade-off attack [12] on block ciphers and in case of stream ciphers, TMTO was proposed by Babbage [13] in 1995 and Golic [14] in 1997 independently. Later on Biryukov and Samir combined Babbage and Golic scheme with Hellmen attack [15]. This attack was further refined by Birykov, Shamir and Wagner and applied on A5/1 [16].
To avoid TMTO on stream ciphers, Hong and Sarkar suggested that state size should be equal to or greater than sum of key size and size of IV and it should be random. Babbabge [13] and Golic [14] suggested that state size should be at least double the size of key.

5.4. Distinguishing Attack

The most important criterion for a good stream cipher design is that keystream generation should be random. A distinguishing attack tries to identify that if a given keystream is a random sequence or a cipher or generator has created it. Distinguishing attack tries to identify the relations between internal state variables and output keystream. The internal structure of a cipher has to be analyzed extensively for distinguishing attack. Distinguishing attack is a known keystream attack.

5.5. Algebraic Attack

The algebraic attack is used in stream ciphers based in LFSRs. These attacks try to find the initial state given some keystream bits. The algebraic attacks has two steps. In the first step, the attack tries to find a system of equations in the bits of the secret key K and the output bits Z_t [22]. If it has enough low degree equations and known key bits stream, then the secret key K can be recovered by solving this system of equations in a second step. This system could be solved using Groebner bases [23] , XL , XSL and others.

For Courtois [26] the algebraic attack can be defined in a synchronous stream cipher, which has a state $S \in GF(2)^n$. At each clock t the state s is updated by a "connection function" $s \rightarrow L(s)$ that is assumed to be linear over GF(2). Then a combine *f* is applied to *s*, to produce the output bit $b_t = f(s)$. The goal for the attack is to find the initial state of s [26] [25].
Flori et al [48] approach how to avoid the algebraic attacks using a good binary strings distribution. Unfortunately, they just had a conjecture and do not have a theorem. However, Wang and Johansson [28] proved that is capable to have a Boolean function [29] with fast algebraic immunity and higher order nonlinearity. To determinate the computation of immunity against algebriac and fast algebriac attack you can consult the Armknecht et al work.
Using this attack Orumiehchiha et al [30] recovered both initial state and secret key, from WG-7 cipher [31], with the time complexity 2^{27}.

5.6. Correlation Attacks

The correlation attack was proposed by Siegenthaler in 1985 [37]. An important work in this area was elaborated by Meier and Staffelbach [60]. After them, Mihaljevi and Goli [34] was one of the promising work. Other important work is from Anderson [35], he started the search for the optimum correlation attack. They opened the world of cryptanalysis to correlation attack.

The correlation attack is defined as:

"The correlation attack exploits the existence of a statistical dependence between the keystream and the output of a single constituent LFSR." [36].

In the Figure 5, we can see how works a stream cipher based on LFSR. The random noise in the Figure 5 is the keystream (LFSRs), the function h(x) is to expand the secret key and the output Ki is the secret key.

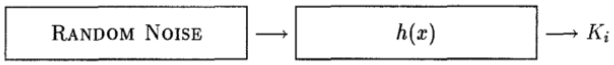

Figure 5: The idea of a stream cipher with LFSR

In the Figure 5 we can see the idea of the correlation attack. Using this attack Mihaljevi et al [33] recovered the internal state of LILI-128 [39] in a complexity time of the order 235.
In the work of Wei et al [38], they presented a new correlation attack on nonlinear combining generators. In the moment, we have a good review about correlation attacks in Meier work [40] and in the work of Canteaut [41].

5.7. Guess-and-Determine Attacks

Guess and determine attacks are general attacks on stream ciphers. As it is clear from the name, in Guess and determine attacks, an attacker guess a part of the internal state and try to recover the full value of internal states by observing the keystream using the guessed part and small amount of known keystream. In the end a part of keystream is generated using the guessed values and then it is compared with the known keystream to check the correctness of the guessed values. In [11] guess and determine attack was given against Polar Bear. Guess and determine attacks were also presented in [12] against SNOW. By irregular clocking, resistance against guess and determine attacks can be increased. Guess and determine attacks are more effective against word oriented stream ciphers [13].

5.8. Linear Masking Attacks

Linear masking attacks can be applied to those ciphers where some non-linear process resembling block cipher design exist and in which linear masking is used to hide this process. In these attacks first of all a non-linear characteristic is distinguished that exhibits some bias. Then we look at linear process and get some missing linear combinations. The same linear combinations are applied to the cipher output and we try to find the traces of distinguishing property. Coppersmith et al. in [14] described a generic attack on stream ciphers using linear masking. Watanabe et al. proposed linear masking attack on SNOW [15]. It is a form of Guess and Determine attack.

5.9. Related Key Attack

To provide a little bit of extra safety or security some of the cryptographic protocol limits the amount of data, which can be encrypted using a single key. In such cases either the new key is generated with using the IV (initialization vector) and with master key or to change the IV which in turns change the cipher key.

In such type of ciphers if the rekeying strategy relates the inputs to the internal states without sufficient non-linearity then cipher may become prone to a related key attack. These types of weaknesses are not very common in case of stream ciphers but there are some examples of related key attacks. Fluhrer et al. shown the related key attack on RC4 in [16] by exploiting a weakness of invariance in the key initialization algorithm. This weakness of RC4 was used by Stubblefield et al. to break the WEP protocol with practical complexity [17]. Sekar et al. presented a related key attack on Py-family of stream ciphers [18].

5.10. Divide and Conquer Attack

Divide and conquer is a common technique to divide the problem into small problems and try to solve the problem step by step. The same strategy is applied in case of divide and conquer attack where a cipher is partitioned into components and only a few key bits are determined in each stage. First the most vulnerable components are attacked. Siegenthaler [19] originally pointed out this concept. The attack can be termed as successful only if complexities of all the stages are smaller than the exhaustive key search. Some examples of attack are [20]. High correlation immunity decreases the vulnerability to divide and conquer attack [21].

6. CONCLUSION

In this paper, we have tried to describe the existing cryptanalytic attacks on stream ciphers and counter measures to these attacks have been suggested with different examples. These attacks are generally tried against any new cryptographic primitive at first. In order to develop a new secure stream cipher, it is very much necessary that these attacks should be taken into consideration during development and countermeasures of these attacks should be applied in the design, so that the new design is not vulnerable to these attacks. Though these are the available techniques in literature

for cryptanalysis of the stream ciphers but generally combinations and variants of these attacks can be used in future and just by overcoming these attacks any cryptographic primitive cannot be assumed secure. We are working in the field of cryptanalysis for further enhancement of available attacks and their applications on available stream ciphers.

REFERENCES

[1]. Russell, Matthew. "CRANK – CryptANalysis toolKit". 21 Aug062E0401. URL: http://crank.sourceforge.net/about.html (24 Nov 2001).
[2]. Brown, Lawrie. "Classic Cryptography". 22 Feb 1996. URL:http://www.geocities.com/SiliconValley/Network/2811/classic/classical.htm (24 Nov 2001)
[3]. Teitelbaum, Jeremy T., "Classic Ciphers". 1995. URL: http://raphael.math.uic.edu/~jeremy/crypt/intro.html (24 Nov 2001)
[4]. SANS Institute. "SANS GIAC Training and Certification". URL: http: //www. sans. Org / giactc /GIAC_certs.htm (24 Nov 2001).
[5]. Agrawal, D., Archambeault, B., Rao and J.R., Rohatgi, P.: "The EM Side-Channel(s): Attacks and Assessment Methodologies". In: Cryptographic Hardware and Embedded Systems – CHES 2002 (2002).
[6]. W. Fischer, B. M. Gammel, O. Kniffler and J. Velton, "Differential Power Analysis of Stream Ciphers," Topics in Cryptology-CT-RSA 2007, Springer-Verlag, LNCS, Vol. 4377, pp. 257–270, 2007.
[7]. P. Kocher, J. Jaffe and B. Jun, "Differential Power Analysis", in the Proceedings of Crypto 1999, LNCS, vol 1666, pp 398–412, Santa-Barbara, CA, USA, August 1999.
[8]. J.-F. Dhem, F. Koeune, P.-A. Leroux, P. Mestré, J.-J. Quisquater, and J.-L. Willems, "A practical implementation of the timing attack", Proc. CARDIS 1998, Smart Card Research and Advanced Applications (J.-J. Quisquater and B. Schneier, eds.), LNCS, Springer, 1998.
[9]. P. Kocher, "Timing attacks on implementations of Diffe-Hellman, RSA, DSS, and other systems", Advances in Cryptology - CRYPTO '96, Sant Barbara, California (N. Koblitz, ed.), LNCS, vol. 1109, Springer, 1996, pp. 104-113.
[10]. S. Fluhrer and D. McGrew, "Statistical Analysis of the Alleged RC4 Keystream Generator", proceedings of FSE 2000, Lecture Notes in Computer Science 1978, pp. 19-30, Springer-Verlag, 2001.
[11]. J. Mattsson. "A Guess and Determine Attack on the Stream Cipher Polar Bear". eSTREAM, ECRYPT Stream Cipher Project, Report 2006/017, 2006. http://www.ecrypt.eu.org/stream.
[12]. P. Hawkes and G. G. Rose. "Guess-and-Determine Attacks on SNOW". In Selected Areas in Cryptography, pages 37–46, 2002
[13]. Philip Hawkes and Gregory G. Rose. "Exploiting Multiples of the Connection Polynomial in Word-Oriented Stream Ciphers", Proceedings of the 6th International Conference on the Theory and Application of Cryptology and Information Security: Advances in Cryptology, p.303-316, December 03-07, 2000.
[14]. D. Coppersmith, S. Halevi, and C. Jutla. "Cryptanalysis of stream ciphers with linear masking". In Advances in Cryptology - CRYPTO 2002, volume 2442 of Lecture Notes in Computer Science, pages 515 – 532, January 2002.
[15]. D.Watanabe, A. Biryukov, and C. De Canniere. "A Distinguishing Attack of SNOW 2.0 with Linear Masking Method". In Selected Areas in Cryptography (SAC 2003), LNCS 3006, pp. 222{233, Springer-Verlag, 2004.
[16]. Scott Fluhrer, Itsik Mantin, and Adi Shamir. "Weaknesses in the key scheduling algorithm of RC4". In Mitsuru Matsui, editor, Proceedings of the 8th International Workshop on Fast Software Encryption, volume 2355 of Lecture Notes in Computer Science, pages 1–24. Springer-Verlag, 2001.
[17]. Adam Stubblefield, John Ioannidis, and Avi Rubin. "Using the Fluhrer, Mantin, and Shamir Attack to break WEP". Technical report, TD-4ZCPZZ AT&T Labs Technical Report, 2001.
[18]. Sekar, G., Paul, S., & Preneel, B., "Related-key attacks on the Py-family of ciphers and an approach to repair the weaknesses". In LNCS Vol. 4859. Indocrypt'07 (pp. 58–72). Berlin: Springer, 2007.
[19]. T. Siegenthaler, "Decrypting a class of stream ciphers using ciphertext only," IEEE Trans. Computers, vol. C-34, no. 1, pp. 81–84, 1985.
[20]. Kevin Chen, Matt Henricksen, Leonie Simpson, William Millan and Ed Dawson. "A Complete Divide and conquer attack on the Alpha1 stream cipher". ICISC 2003, 6th International Conference, Seoul, November 27- 28, 2003, Revised papers, volume 2971, of Lecture Notes In Computer Science page 418-431. Springer 2004.
[21]. T. Siegenthaler, "Design of Combiners to Prevent Divide and Conquer Attacks", Advances in Cryptology-CRYPTO'85, H. C. Williams (Ed.), LNCS 218, Springer-verlag, 1986, pp. 273-279.
[22]. Frederik Armknecht. Improving fast algebraic attacks. In Bimal Roy and Willi Meier, editors, Fast Software

Encryption, volume 3017 of Lecture Notesin Computer Science, pages 65–82. Springer Berlin Heidelberg, 2004.
[23]. W. Boege, R. Gebauer, and H. Kredel. Some examples for solving systems of algebraic equations by calculating groebner bases. J. Symb. Comput., 2(1):83–98, January 1986.
[24]. W. Boege, R. Gebauer, and H. Kredel. Some examples for solving systems of algebraic equations by calculating groebner bases. J. Symb. Comput., 2(1):83–98, January 1986.
[25]. Nicolas T. Courtois and Willi Meier. Algebraic attacks on stream ciphers with linear feedback. In Proceedings of the 22Nd International Conference on Theory and Applications of Cryptographic Techniques, EUROCRYPT'03 pages 345–359, Berlin, Heidelberg, 2003. Springer-Verlag.
[26]. NicolasT. Courtois. Fast algebraic attacks on stream ciphers with linear feedback. In Dan Boneh, editor, Advances in Cryptology - CRYPTO 2003, volume 2729 of Lecture Notes in Computer Science, pages 176–194. Springer Berlin Heidelberg, 2003.
[27]. Jean-Pierre Flori, Hugues Randriam, Grard Cohen, and Sihem Mesnager. On a conjecture about binary strings distribution. In Claude Carlet and Alexander Pott, editors, Sequences and Their Applications SETA 2010, volume 6338 of Lecture Notes in Computer Science, pages 346–358. Springer Berlin Heidelberg, 2010.
[28]. Qichun Wang and Thomas Johansson. A note on fast algebraic attacks and higher order nonlinearities. In Xuejia Lai, Moti Yung, and Dongdai Lin, editors, Information Security and Cryptology, volume 6584 of Lecture Notes in Computer Science, pages 404–414. Springer Berlin Heidelberg, 2011.
[29]. A. Canteaut and M. Videau. Symmetric boolean functions. Information Theory, IEEE Transactions on, 51(8):2791–2811, Aug 2005.
[30]. Mohammad Ali Orumiehchiha, Josef Pieprzyk, and Ron Steinfeld. Cryptanalysis of wg-7: a lightweight stream cipher. Cryptography and Communications, 4(3-4):277–285, 2012.
[31]. Yiyuan Luo, Qi Chai, Guang Gong, and Xuejia Lai. A lightweight stream cipher wg-7 for rfid encryption and authentication. In Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE, pages 1–6, Dec 2010.
[32]. Willi Meier and Othmar Staffelbach. Fast correlation attacks on certain stream ciphers. Journal of Cryptology, 1(3):159–176, 1989.
[33]. Miodrag J. Mihaljevi, Sugata Gangopadhyay, Goutam Paul, and Hideki Imai Internal state recovery of keystream generator lili-128 based on a novel weakness of the employed boolean function. Information Processing Letters,112(21):805 – 810, 2012.
[34]. MiodragJ. Mihaljevi and JovanDj. Goli. Convergence of a bayesian iterative error-correction procedure on a noisy shift register sequence. In RainerA. Rueppel, editor, Advances in Cryptology EUROCRYPT 92, volume 658 of Lecture Notes in Computer Science, pages 124–137. Springer Berlin Heidelberg,1993.
[35]. Ross Anderson. Searching for the optimum correlation attack. In Bart Preneel, editor, Fast Software Encryption, volume 1008 of Lecture Notes in Computer Science, pages 137–143. Springer Berlin Heidelberg, 1995.
[36]. Anne Canteaut. Correlation attack for stream ciphers. In HenkC.A. van Tilborg and Sushil Jajodia, editors, Encyclopedia of Cryptography and Security, pages 261–262. Springer US, 2011.
[37]. T. Siegenthaler. Decrypting a class of stream ciphers using ciphertext only. Computers, IEEE Transactions on, C-34(1):81–85, Jan 1985.
[38]. Yongzhuang Wei, E. Pasalic, and Yupu Hu. A new correlation attack on nonlinear combining generators. Information Theory, IEEE Transactions on,57(9):6321–6331, Sept 2011.
[39]. Andrew Clark, Ed Dawson, J. Fuller, Jovan Dj. Golic, H.-J. Lee, William Millan, S.-J. Moon, and Leone Simpson. The lili-ii keystream generator. In Proceedings of the 7th Australian Conference on Information Security and Privacy, ACISP '02, pages 25–39, London, UK, UK, 2002. Springer-Verlag.
[40]. Willi Meier. Fast correlation attacks: Methods and countermeasures. In Antoine Joux, editor, Fast Software Encryption, volume 6733 of Lecture Notes in Computer Science, pages 55–67. Springer Berlin Heidelberg, 2011.
[41]. Anne Canteaut. Correlation attack for stream ciphers. In HenkC.A. van Tilborg and Sushil Jajodia, editors, Encyclopedia of Cryptography and Security, pages 261–262. Springer US, 2011.

ABOUT THE AUTHOR:

1. Mr. A.Mahendar is research scholar, department of computer science & engineering, Acharya Nagarjuna University. mahi.adapa@gmail.com.

2. Dr. E. Srinivasa Reddy is working as a professor in the department of computer science & engineering, at Acharya Nagarjuna University. edara_67@yahoo.com.