Study of NLFSR and Reasonable Security Improvement on Trivium Cipher

Subhrajyoti Deb, Bhaskar Biswas and Nirmalya Kar

Abstract The decision regarding widely acceptable stream cipher for hardware security is one of the challenging assignment. Trivium cipher is ironed out most of the weakness of the other stream ciphers. Beside this, our paper presents the security analysis of Trivium and some modification of cipher which gives better security level. Our primary focus is finding some particular biases of output and reasonable algebraic attack with some guessing approach. More specifically, we show some modifications which can increase its security level without changing its internal structure. For this cipher finally we obtain some new cryptanalytic result and the next part of the paper we study some algebraic analysis of Trivium cipher and we try to show that it is vulnerable to different types of attacks and try to recover the overall internal state of the stream cipher.

Keywords Cryptanalysis · Trivium cipher · Stream cipher · Algebraic attack · Non-linear feedback shift register · Linear equation

1 Introduction

In the literature review of symmetric key cryptology, the stream ciphers are observe that to be one of the most important primitives. So far, stream cipher that has some kind of internal state which is regularly updated [1]. For instance there length is n bits, then the function [2] f can be defined as

S. Deb (✉) · B. Biswas · N. Kar
Computer Science and Engineering Department,
National Institute of Technology, Agartala, India
e-mail: subhrajyoti.cse@nita.ac.in

B. Biswas
e-mail: bhaskar.cse@nita.ac.in

N. Kar
e-mail: nirmalya@nita.ac.in

J.K. Mandal et al. (eds.), *Information Systems Design and Intelligent Applications*,
Advances in Intelligent Systems and Computing 339,
DOI 10.1007/978-81-322-2250-7_73

$$f : \mathbb{GF}(2)^n \to \mathbb{GF}(2)$$

The Trivium stream cipher was originally proposed by De Canniere and Preneel [1]. In data encryption primitives Trivium stream cipher is widely applicable both in hardware and in software level. This stream cipher is a hardware-oriented synchronous stream cipher, which claimed a key size of 80 bits and an initialization vector (IV) size of 80 bits and it remain consistent when it was submitted. In that approach, it presented in [3], Trivium stream cipher is algebraic in nature. Attacking on Trivium are propose by Babbage in [3], that evidence is given that internal state bits of Trivium can be protect with a time complexity $c^{283.5}$. The study and analysis of Trivium literature proposed mainly the algebraic structure of the cipher which has the propensity basis on guess and determine types of attack or it may recover the internal state bits of the Trivium cipher which can be mounted and resist by algebraic attack [4]. We try to modify the literature structure of Trivium so that its appropriate structure can be used to provide a improve security. This modified version of Trivium relay the better security to the internal state recovery of that cipher attack.

1.1 Description of the Cipher

Trivium consists two processes that concentrate the values of 15 specific state bits and uses them both to update 3 bits of the internal state. Let s_1, s_2, …, s_{288} be the 288 internal bits and the key stream generated at a particular time *i* (*i = 0, 1, …, n*). This process is repeated for 4 * 288 = 1,152 times. So it can be summarized by the following generation of pseudo code [5] represent in Algorithm 1.

Algorithm 1 Key and IV Setup Generation Process of Trivium

```
for $t = 1$ to $n$ do
    $(s_1, s_2,...,s_{93}) := (K_1, K_2,...,K_{80}, 0,..., 0)$
    $(s_{94}, s_{95},...,s_{177}) := (IV_1, IV_2,...,IV_{80}, 0, 0, 0, 0)$
    $(s_{178}, s_{179},...,s_{288}) := (0,...,0, 1, 1, 1)$
    $t_1 := s_{66} \oplus s_{93}$
    $t_2 := s_{162} \oplus s_{177}$
    $t_3 := s_{243} \oplus s_{288}$
    $z_i := t_1 \oplus t_2 \oplus t_3$
    $t_1 := t_1 \oplus s_{91} . s_{92} \oplus s_{171}$
    $t_2 := t_2 \oplus s_{175} . s_{176} \oplus s_{264}$
    $t_3 := t_3 \oplus s_{286} . s_{287} \oplus s_{69}$
    $(s_1, s_2,...,s_{93}) := (t_3, s_1,...,s_{92})$
    $(s_{94}, s_{95},...,s_{177}) := (t_1, s_{94},...,s_{176})$
    $(s_{178}, s_{179},...,s_{288}) := (t_2, s_{178},...,s_{176})$
end for
```

In each round of Trivium cipher a single bit include linear function of six state bits. Generally this key-stream process repeated until N $\leq 2^{64}$ bits of key-stream generated. Each state bits are rotated, and as well as the process is repeats in the same manner. The generation of key stream is described by the following pseudo-code [5] represent in Algorithm 2.

Algorithm 2 Key Stream Generation Process of Trivium

```
for $t = 1$ to $n$ do
    $t_1 := s_{66} \oplus s_{93}$
    $t_2 := s_{162} \oplus s_{177}$
    $t_3 := s_{243} \oplus s_{288}$
    $z_i := t_1 \oplus t_2 \oplus t_3$
    $t_1 := t_1 \oplus s_{91} \,.\, s_{92} \oplus s_{171}$
    $t_2 := t_2 \oplus s_{175} \,.\, s_{176} \oplus s_{264}$
    $t_3 := t_3 \oplus s_{286} \,.\, s_{287} \oplus s_{69}$
    $(s_1, s_2, \ldots, s_{93}) := (t_3, s_1, \ldots, s_{92})$
    $(s_{94}, s_{95}, \ldots, s_{177}) := (t_1, s_{94}, \ldots, s_{176})$
    $(s_{178}, s_{179}, \ldots, s_{288}) := (t_2, s_{178}, \ldots, s_{176})$
end for
```

2 Study of Non-linear Feedback Shift Register

2.1 Structure of Non-linear Feedback Shift Register

A non-linear feedback shift register (NLFSR) works as same as a linear feedback shift register (LFSR) but there is some difference between them, in linear feedback shift register the feedback function is linear, that is it is only the combination of contents of L stages for a L length linear feedback shift register [4]. In non-linear feedback shift register feedback function is not linear, a non-linear feedback shift register is a combination of L stages of a L-length NLFSR [2, 6–8]. In non-linear contents are produced by adding a stage content to feedback (Fig. 1).

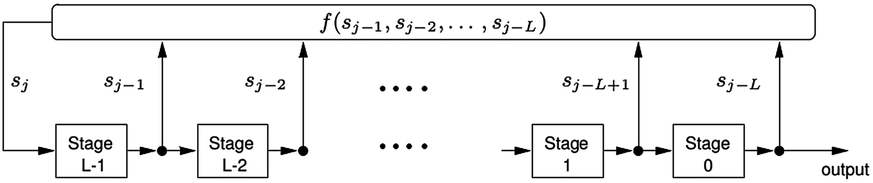

Fig. 1 A NLFSR structure of length L

The feedback function of NLFSR can be defined as

$$s_j = f(s_{L-1}, s_{L-2}, \ldots, s_0) \text{ where j} = (L, L+1, L+2\ldots).$$

2.2 Non-linear Combination Generator

Non-linear combination generator is another type of a linear feedback shift register. More then one linear feedback shift register is used and each linear feedback shift register where output is a key bit. Output of these linear feedback shift register are combined for producing a final output bit [4, 7, 8]. Each output bit of each linear feedback shift register are combined non-linearly for producing a non-linear output bit.

A Non-linear feedback shift register state variables can be represented as $x = (x_0, x_1, \ldots x_{l-1})$ which denote the state variable $s = (s_0, s_1, \ldots s_{l-1})\ \varepsilon(0, 1)^l$.

3 Implementation Details on Trivium Cipher

The linear correlations between key stream bits and internal state recovery bits are simple to compute due to the reason of $\mathbb{Z}_i$ is simply describe to be equal to $s_{66} \oplus s_{93} \oplus s_{162} \oplus s_{177} \oplus s_{243} \oplus s_{288}$. Trivium stream cipher state evolves in a non linear way i.e. not easy for an attacker to add these equations in order to efficiently recover the internal state [9]. 288-bit nonlinear feedback shift register are uses in Trivium. Trivium cipher uses nonlinear state-update three bits of the internal state and keystream bits are represented as a linear combination of the contents of six stages of its internal state.

This paper we try to describe the algebraic representation of Trivium. In that section we briefly explain some differential fault analysis of Trivium stream cipher and try to generate some the polynomial equation system that represent the inner state of Trivium cipher. Our attack description and the results are presented in next section respectively.

3.1 Proposed Work

In this paper we analyse the biases of Trivium cipher. As a case-study of Trivium cipher, we present a new method of finding linear feedback bits and the linear combination of keystream bits of the cipher. Our approach is to change some

Table 1 Testing key and IV setup speed

No. of keys	Total time	Key setup speed
9,000	202.67 usec	45.10 cycles/byte
700	223.14 usec	638.50 cycles/byte

Table 2 Testing stream encryption speed

No. of encrypted blocks	Total time	Encryption speed
22 blocks	242.15 usec	5.38 cycles/byte

particular state bits without changing the elegant structure of stream cipher and compare the output with the original one. So far, our proposed method confined that some modification of statebits increases security level without changing internal structure of the cipher. We are able to show that statebit stream and generated new bit stream variables has a significant impact on the success of our algebraic attack on Trivium cipher.

3.2 *Performance Analysis*

The hardware testing eStream framework determined five conditions of performance based on power consumption, area compactness, throughput, simplicity and flexibility. Trivium cipher feedback can be allowing more than one bit to be per cycle processed. Our implemented testing key and IV performance result are described in Table 1 and testing stream encryption speed performance result described in Table 2.

4 Analysis of Different Attacks on Trivium Cipher

Out of two variants of attacks state recovering attack is the first variant which tries to guess some state bits value. In that case it reduces the linear equations which can be implemented based on the elimination process. Another form of attack is known as distinguishing attack. This attack procedure collects information on keystream generation process and creates a distinguisher. Main focus stays only on state recovery attack [2]. As a polynomial expression F(K, IV) = Y, where K is the secret key of n bits (x_1, x_2 …., x_n) and IV is known publicly initialization vector (IV) of m bits (v_1, v_2 ….., v_m).

We use the expression F_i(K, IV) = Y_i is to represent the polynomial expression for the ith output bit of Trivium cipher [10]. Our main approach of this attack is to combine the equations in K but randomly representation of IV in such a manner which is low degree equations in the variable K [11]. The chosen IV s can be chosen in such a way that linear equations of unknowing bits in K can be obtained. In fact internal statebits of Trivium consists of 288 bits which are set in 3 shift registers respectively [11]. In generally some primitive polynomials can be determined to use to create an LFSR. This LFSR has the degree of the polynomial that will cycle over all non-zero states in Trivium [10, 12, 13]. The pseudo number generation in n bit shift registers are randomly scrolls between (2n − 1) values. This process quickly computes using less number of combinational logic. So basically when it reaches its final state, it tries to traverse the sequence exactly as before [9]. This External LFSR is one way of implementing i.e. all XOR gates are sequentially into one another and ends up as the input to the least or most significant bit of the LFSR [11]. So XORs are external form of the shift register and in internal structure of LFSR XOR gates are not sequential [11]. Recall that if f(X) ε $\mathbb{F}_p[X]$ is a polynomial of n degree such that f(0) $\neq$ 0, then there is positive integer $\phi \leq q^n - 1$ such that f(X) divides $X^{\phi} - 1$. The least such ϕ is called the order of f(X) and is denoted by f(X). If f(X) is any nonzero polynomial in $\mathbb{F}_p[X]$, then we can write f(X) = X^{ς} g (X), where $\varsigma \geq 0$ and g(X) ε $\mathbb{F}_p[X]$.

In Raddum's approach the key stream generation procedure of Trivium takes one linear combination of 6 sequence bits and 2 sequence bits from A, B and C registers. The key stream bits are represented in following way [14, 15].

$$z_{66} = A_{93} \oplus A_{66} \oplus B_{81} \oplus B_{66} \oplus C_{111} \oplus C_{66}$$
$$z_{69} = A_{96} \oplus A_{69} \oplus B_{84} \oplus B_{69} \oplus C_{113} \oplus C_{68}$$

4.1 Reasonable Degree Changes in Trivium Cipher

In our paper, Trivium reasonable modification is easy and elegant structure. The proposed modification is minimizing the number of linear equations that appears during the previous clock bits [12, 13]. In Trivium, our proposed modified equations forms a higher degree than the original structure. In our paper we try to compare with original algebraic structure of Trivium and modified structure of Trivium. In linear equations and other lower degree polynomial equations can be guessed and finally the remaining secret bits can be recovered. The proposed idea of modification tries to discard this simple recovery of bits even some bits which are guessing bit position [10–12]. So this approach may be determined if feedback of the product term comes in the output bits in prior manner. This modify key generation process is described by the following Algorithm 3.

Algorithm 3 Reasonable modification of Key and IV Set up Generation Process of Trivium

```
for $t = 1$ to $n$ do
    $t_1 := s_{33} \oplus s_{93}$
    $t_2 := s_{173} \oplus s_{177}$
    $t_3 := s_{209} \oplus s_{288}$
    $z_i := t_1 \oplus t_2 \oplus t_3$
    $t_1 := t_1 \oplus s_{91} \,.\, s_{92} \oplus s_{143}$
    $t_2 := t_2 \oplus s_{175} \,.\, s_{176} \oplus s_{243}$
    $t_3 := t_3 \oplus s_{286} \,.\, s_{287} \oplus s_{67}$
    $(s_1, s_2, \ldots, s_{93}) := (t_3, s_1, \ldots, s_{92})$
    $(s_{94}, s_{95}, \ldots, s_{177}) := (t_1, s_{94}, \ldots, s_{176})$
    $(s_{178}, s_{179}, \ldots, s_{288}) := (t_2, s_{178}, \ldots, s_{287})$
end for
```

In that section we try to get linear equations comparison with of the original structure and of Trivium cipher [9]. As we mentioned previously, the feedback product of the cipher comes first in output prior to the original structure [11, 13]. Consequently s_1, s_{94} and s_{178} are replaced by t_3, t_1, and t_2 respectively. After 65 clocks with update of s_1, s_{94} or s_{178} in such a manner. The Trivium cipher modification approach does not induced any extra logic gate except a reasonable change in the position of the feedback function. This procedure is to reduce the number of linear equations of the cipher. The difference between the degrees of algebraic structure of cipher with the original version of the cipher is described in Fig. 2.

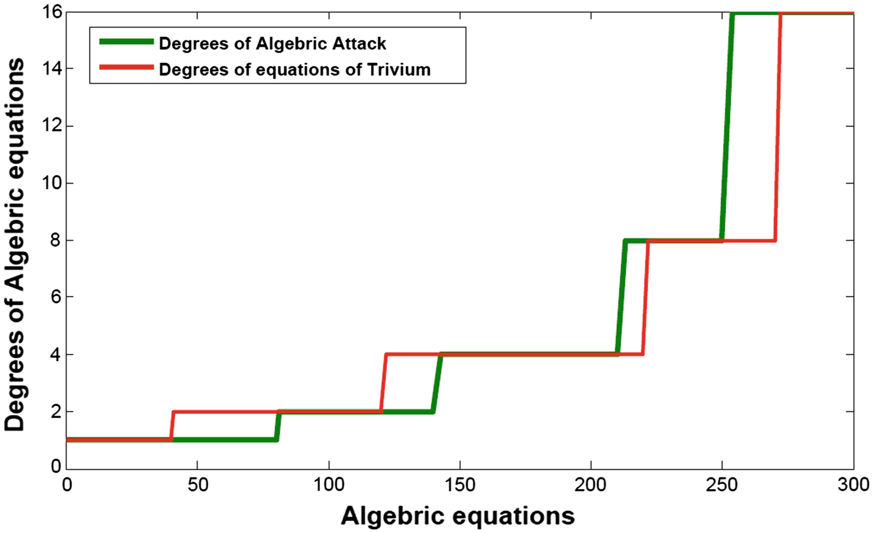

Fig. 2 Comparison of degree of trivium equation with algebraic attack

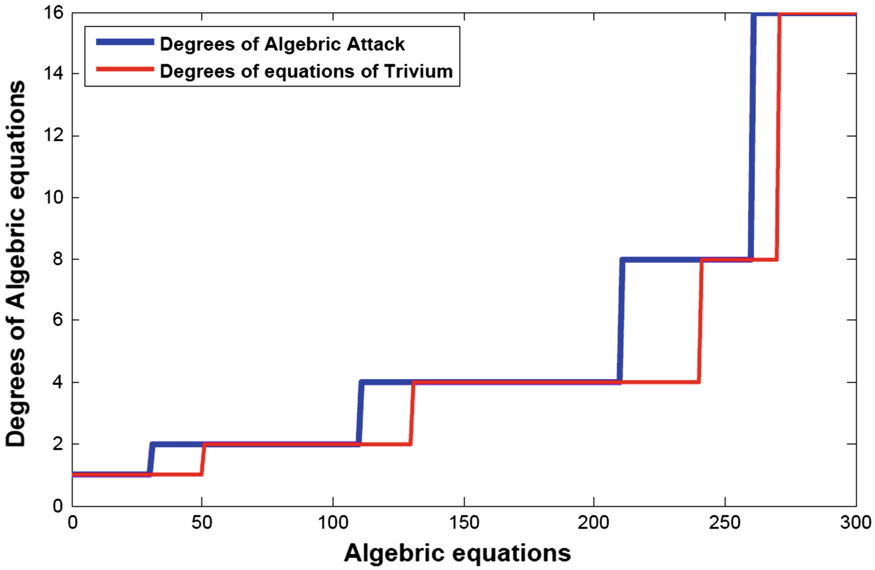

Fig. 3 Comparison between modified structure of trivium with algebraic attack

After the reasonable changes we get first 33 linear equations which is same as original cipher. As we compare with 66 linear equations of the original cipher, some degrees are reasonably changed step by step. Finally we compare our version with the original one, we get some reasonable high degrees rather than the original version is described in Fig. 3. The sequential guessing bits are not ideal only the reason of product terms of sequential bits [13, 16]. Trivium structure has three registers which is updating the feedback variables of the three registers and it becomes mutually dependent based on the degrees of equations.

5 Conclusion and Future Work

In this paper we try to present some weakness of Trivium stream cipher which provide random sequence generators through non-linear shift registers. Several research problems on Trivium is till remain unanswered like patterns of behavior, algebraic attack, lengths, weak keys etc. So questions arises basis on its bits security level which can be rapidly increase with equal number of statebits. Under this design paradigm, we present reduced variant sized modifications of that stream generator, which increases the bit security level of Trivium cipher. We speculate the properties identified in the reduced sized model would remain invariant in the original ones and it would be worthwhile to innovate a new framework of Trivium cipher.

References

1. De Canniere, C., Preneel, B.: Trivium Specifications. eSTREAM, ECRYPT Stream Cipher Project, Report (2008)
2. Khoo, K., Gong, G., Lee, H.K.: The rainbow attack on stream ciphers based on Maiorana-Mcfarland functions :ACNS 2006, LNCS 3989, pp. 194–209 (2006)
3. Babbage, S.: Some thoughts on Trivium. http://www.ecrypt.eu.org/stream/papersdir/2007/007.pdf
4. Berbain, C., Gilbert, H., Joux, A.: Algebraic and correlation attacks against linearly filtered non linear feedback shift registers. In: Avanzi, R.M., Keliher, L., Sica, F. (eds.) Selected Areas in Cryptography (SAC 2008). Lecture Notes in Computer Science, vol. 5381, pp. 184–198. Springer, Berlin (2009)
5. Turan, M.S., Kara, O.: Linear Approximations for 2-round Trivium. http://www.ecrypt.eu.org/stream/papersdir/2007/008.pdf
6. Dubrova, E.: A List of Maximum-Period NLFSRs, Cryptology ePrint Archive, Report 2012/166, March 2012, http://eprint.iacr.org/2012/166 (2012)
7. Dubrova, E.: A scalable method for constructing Galois NLFSRs with period 2^n-1 using cross-join pairs. Technical Report 2011/632, Cryptology ePrint Archive, November 2011. http://eprint.iacr.org/2011/632 (2011)
8. Dutta, A., Paul, G.: Deterministic hard fault attack on trivium. In: Advances in Information and Computer Security. Lecture Notes in Computer Science, vol. 8639, pp 134–145. Springer, Berlin (2014)
9. Maximov, A., Biryukov, A.: Two trivial attacks on trivium. https://eprint.iacr.org/2007/021.pdf (2007)
10. Teo, S.G., Wong, K.K.H., Bartlett, H., Simpson, L., Dawson, E.: Algebraic analysis of Trivium-like ciphers http://www.eprint.iacr.org/2013/240.pdf
11. Gierlichs, B. et al.: Susceptibility of eSTREAM Candidates towards Side Channel Analysis. SASC 2008—The State of the Art of Stream Ciphers, Workshop Record, pp. 123–150. http://www.ecrypt.eu.org/stream (2008)
12. Mohamed, M.S.E., Bulygin, S., Buchmann, J.: Improved differential fault analysis of trivium. COSADE 2011—Second International Workshop on Constructive Side-Channel Analysis and Secure Design. http://cosade2011.cased.de/files/2011/cosade2011_talk15_paper.pdf (2011)
13. Modifications in the Design of Trivium to Increase its Security Level Afzal, M., Masood, A. https://eprint.iacr.org/2009/250
14. Raddum, H.: Cryptanalytic results on trivium, eSTREAM, ECRYPT Stream Cipher Project, Report 2006/039 (2006)
15. Schilling, T.E., Raddum, H.: Solving compressed right hand side equation systems with linear absorption. In: Helleseth, T., Jedwab, J. (eds.) Sequences and Their Applications (SETA 2012). Lecture Notes in Computer Science, vol. 7280, pp. 291–302. Springer, Berlin (2012)
16. Borgho, J., Knudsen, L.R., Matusiewicz, K.: Hill climbing algorithms and trivium. In: Biryukov, A., Gong, G., Stinson, D.R. (eds.) Selected Areas in Cryptography (SAC 2010). Lecture Notes in Computer Science, vol. 6544, pp. 57–73. Springer, Berlin (2011)