IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 1, January 2011
ISSN (Online): 1694-0814
www.IJCSI.org

Distinguishing attack on bivium algorithm

Kimia Khodayari **Arash Dana**

Electrical Engineering Department, Islamic Azad University, Central Tehran Branch
Tehran, Iran

Abstract

Trivium is one of the candidates of eStream Project which was designed by C. de Cannier and B. Preneel and was promoted up to phase 3. Bivium is a truncated version of Trivium which follows the same structure and principles.

In this presenting the distinguishing attack on Brivium algorithm is the aim, which has been downgraded by the help of linear approximation and the outcomes of complication simulation to $O(^{230.79})$ which is more successful than the attacks done by Alex Biryukov and Maximov in the complication was $O(2^{32})$

Keywords: *Distinguishing attack, stream ciphers, key stream generation, NLFSR.*

1. Introduction

Social life, as a necessity, has made humans develop a tendency for personalizing and concealing private information and telecommunication in order to achieve security. The developments in the technology of information telecommunication in the 20th century and the need for common and at the same time scientific structures and the Shanon article [1] in 1949 turned the art of encryption into the science of encryption.

The ideas in the above article are still being used even in recently suggested structures. Moreover, the techniques of codes' analysis have enjoyed considerable progress. Today, people use the Worldwide Web in order to transmit huge amounts of data. Homogeneity and assurance of encryption authenticity have also developed vastly for the sake of confidentiality.

In today's coding, symmetrical key code is used to create data confidentiality. In this technique, two individuals can share a secret key and utilize symmetrical key code. The data sender and recipient share a common symmetrical key code and secret code. The sender encrypts the message with code and key system and the coded text is created, which is transferred via a secure channel. The data recipient decodes the encrypted text in order to achieve the original message.

Enemy may access the coded text. In order to avoid enemy access of data, a strong code and key system must be used. There exist two algorithms for symmetrical key codes: Stream and Block ciphers [2].

A typical protracted cipher has an update function and an outlet one. The condition of the cipher is updated in a daily manner. The outlet function produces the chain bits of the key in accordance with the situation and coding or decoding processes are done. If the initial state of the protracted cipher is not equal to the key, there will be a need for a key setup stage in which the initial state is achieved according to the key. If a key with various initial vectors is used for chain production of the key, there will be a need for initial key/IV setup stage in which the initial state is achieved according to the key and the initial vector.

Protracted ciphers are divided into two groups in accordance to the situation update: If the situation is independent from the message, they are called synchronous protracted cipher. In such a system, there is a need for the sender and the recipient to be synchronous. If the situation is dependent on N bits of text which have been encrypted, there will be a synchronous protracted cipher. [3]

In 2004, eSTREAM project [6] was planned by ECRYPT. This project aimed at design and analysis of protracted ciphers in order to present a trustworthy collection of such codes. This project was carried out in two separate profiles:

- Profile 1: protracted ciphers for software applications with high transmit rate.
- Profile 2: protracted ciphers for hardware applications with limited liabilities.

The required characteristics for protracted ciphers in eSTREAM are as follows:

1-Security

2-Functionality in comparison with AES from at least one aspect.

3-Functionality in comparison with other candidates.

4-Simplicity and flexibility.

Clarity and completeness. Until 29 Apr. 2005, the final deadlines for the presentation of the initial plans, 34 algorithms of protracted ciphers were registered. These algorithms were analyzed in three different phases by the coding society and ultimately in 2008 the final list of algorithms was announced by eSTREAM. [4, 5]

1-2)Distinguishing attack

One of the possible tools to analyse a stream cipher is the linear statistical distinguisher approach introduced by J. Goli [12].His work contains the basic notion, the basic mathematical results, a method for finding distinguishers called Linear Sequential Circuit Approximation (LSCA), The purpose of a distinguishing attack is to provide evidence that the generated keystream sequence is not completely random. A distinguishing attack is not as strong as a key-recovery attack, but can provide some undesired information leakage to the adversary. Note that a key-recovery attack is also a distinguishing attack, so if we want to make things simple we can simply state that a good stream cipher should be resistant to distinguishing attacks. The strength of this required resistance, i.e., the required computational complexity, memory, and length of the keystream sequence sequence for a successful distinguisher, is an issue of debate. [13]

1-3) triviume&bivium

Trivium is a stream cipher candidate of the eSTREAM project. Trivium is a synchronous stream cipher designed to generate up to 264 bits of key stream from an 80-bit secret key and an 80-bit initial value (IV).

As for most stream ciphers, this process consists of two phases: first the internal state of the cipher is initialized using the key and the IV, then the state is repeatedly updated and used to generate key stream bit It has an internal state of 288 bits and the key of 80 bits. Though the cipher was designed for hardware implementation it is also very fast in software, which makes it one of the most attractive candidates of the competition. [8]

Bivium are simplified versions of Trivium that are built on the same design principles. Their design serves as a tool for investigating Trivium-like ciphers with a reduced complexity Bivium consists of a non-linear feedback shift register (NLFSR) coupled with a linear filter function (LF). The NLFSR operates on a 177-bit state; denoted by (s_1,…s_{177}) The LF takes a linear combination of the state to produce the keystream. Each clock of the cipher involves updating two bits of the state and out putting one bit of keystream, denoted by z(t) The cipher continues to run until the required number of keystream bits are produced. Like Trivium, Bivium incorporates a Key and IV setup stage, where the cipher is clocked a number of times to initialize the state. [11]

The sections of this article are as follows: The 2nd section scrutinizes the structure of trivium algorithm and its truncated version of bivium. In the 3rd one, with the aid of proper linear approximation, the distinguishing attack on bivium has been exerted and the respective complication has been calculated. In the 4th section, the outcomes of this attack and its comparison with other exerted attacks on this algorithm have been explicated. The 5th section includes the conclusion.

2)Algorithm structure

2-1) key stream generation

.

The suggested design contains a 177-bit internal state marked by (s_1,…s_{177}). Each clock of the cipher encompasses updating two bits of the state and outputting one bit of keystream, highlighted by z (t) i. The cipher carries on until the requisite number of keystream bits is produced. The following algorithm is a full explication of the keystream generation: [11]

for i = 1 to N do

$t_1 \leftarrow s_{66} + s_{93}$

$t_2 \leftarrow s_{162} + s_{177}$

$z_i \leftarrow t_1+t_2$

$t_1 \leftarrow t_1+s_{91}.s_{92}+s_{171}$

$t_2 \leftarrow t_2+s_{175}.s_{176}+s_{69}$

$(s_1,s_2,\ldots,s_{93}) \leftarrow (t_2,s_1,\ldots,s_{92})$

$(s_{94},s_{95},\ldots,s_{177}) \leftarrow (t_1,s_{94},\ldots,s_{176})$

end for

Like Trivium, Bivium includes a Key and IV setup stage, where the cipher is clocked a number of times to initialise the state. In our analysis we have not used the initialisation process and this matter will not be discussed further.

Then these vectors replace together as follow as:

$(s_1,s_2,\ldots,s_{93}) \rightarrow (a_1,a_2,\ldots a_{93})$

$(s_{94},s_{95},\ldots,s_{177}) \rightarrow (b_1,b_2,\ldots,b_{84})$

$$a_{t+1}^1 = a_t^{69} \oplus b_t^{69} \oplus b_t^{84} \oplus b_t^{82}.b_t^{83}$$

$$b_{t+1}^1 = a_t^{66} \oplus a_t^{93} \oplus b_t^{78} \oplus a_t^{91}.a_t^{92}$$

for i=92:-1:1

$$\mathbf{a}_{t+1}^{i+1} = \mathbf{a}_t^i$$

end

for i=83:-1:1

$$\mathbf{b}_{t+1}^{i+1} = \mathbf{b}_t^i$$

End

The rest of the outlet key is achieved via the following:

$$z_t = a_t^{66} \oplus a_t^{93} \oplus b_t^{69} \oplus b_t^{84}$$

Two classes of stream ciphers are shown in figure 1, namely Bivium and Trivium. [9]The number of basic components is two or three, respectively. Each basic component (a register) consists of three blocks, each of size divisible by 3. An instance of this class is a specification vector with the blocks' sizes specified.

Bivium →Bi(A_1,A_2,A_3;B_1,B_2,B_3),

Trivium →Tri(A_1,A_2,A_3;B_1,B_2,B_3;C_1, C_2, C_3)

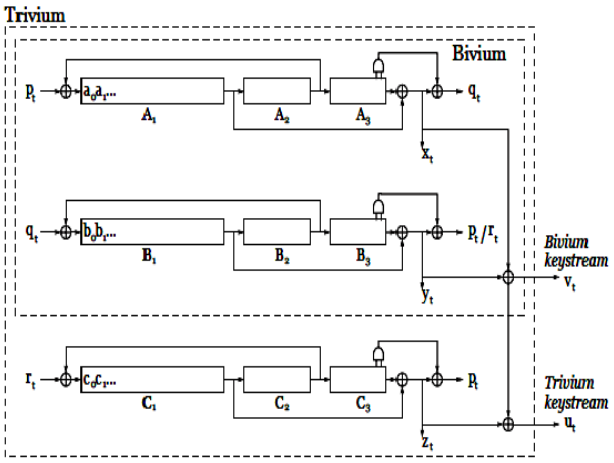

Figure1:Bivium and Trivium classes of stream ciphers

2-2)key update:

The algorithm is initialized by loading an 80-bit key and an 80-bit IV into the 288-bit initial state, and setting all remaining bits to 0. Then, the state is rotated over 4 full cycles, in the same way as explained above, but without generating key stream bits. This is summarized in the pseudo-code below: [8]

$(s_1,s_2,\ldots,s_{93})\leftarrow(K_1,\ldots K_{80},0,\ldots,0)$

$(s_{94},s_{95},\ldots,s_{177})\leftarrow(IV_1,\ldots,IV_{80},0,\ldots,0)$

3.Distinguishing attack on bivium

In the first stage we try to figure out a linear approximation for non-linear updating section. Then we approximate the multiplication of the two bits. The probability of this approximation is 3/4. Thus we will have:

$$a_{t+1}^1 = a_t^{69} \oplus b_t^{69} \oplus b_t^{84}$$

$$b_{t+1}^1 = b_t^{78} \oplus a_t^{66} \oplus a_t^{93}$$

The addition of the first two columns on the right will be Zt.We are after the time delay in which the following equation exists most probably:

$a_{t+1}^1 \oplus a_t^{69} \oplus b_{t+1}^1 \oplus b_t^{78} = a_{t+\tau}^{66} \oplus a_{t+\tau}^{93} \oplus b_{t+\tau}^{69} \oplus b_{t+\tau}^{84}$

With the aid of simulation, the best delay will be $\tau = 46$.

A : $b_{t+1}^1 \oplus b_t^{78} = a_t^{66} \oplus a_t^{93}$

B : $a_{t+1}^1 \oplus a_t^{69} = b_t^{69} \oplus b_t^{84}$

C : $z_t = a_{t+1}^1 \oplus a_t^{69} \oplus b_{t+1}^1 \oplus b_t^{78}$

D : $z_{t+\tau} = a_{t+1}^1 \oplus a_t^{69} \oplus b_{t+1}^1 \oplus b_t^{78}$

E : $z_t \oplus z_{t+\tau} = 0$

P(C) = P(A∩B) + P(A'∩B')= 1-P(A)-P(B)+2×P (A∩B).

Our goal for the distinguisher is the calculation of p(E).

P(A∩B) =0.5624

P(C)= 0.6258

P(E) = P(C∩D) + P(C'∩D')

P(E) = P(D|C)×P(C) + P(D'|C')×P(C')

P(D|C) = 0.4016

P(D'|C') = 0.6644

P(E) = 0.49999

The bias of the distinguisher is obtained via the following:

$\varepsilon = |1-2\times P|$

$\varepsilon = 2.32083\times10^{-5}$

$O(\varepsilon^{-2}) = O(2^{30.79})$

4.simulation result

In [7] Julia Borghoff, Lars R. Knudsen and Mathias Stolpe propose a new approach to solve the system of equations for internal state recovery of Bivium using combinatorial optimization with an estimated time complexity of $O(2^{64.5})$ seconds. In [10] Havard Raddum proposes an algebraic attack on Bivium using Minisat for solving the system of equations with a total complexity of order $O(2^{56})$. Cameron McDonald, Chris Charnes and Josef Pieprzyk [11] introduce a type of guess and determine attack on Bivium with a complexity of approximately $O(2^{52.3})$. In [9] Alexander Maximov and Alex Biryukov perform a state recovery attack on Bivium with a complexity of order $O(2^{51})$ and a distinguishing one with a complexity of order $O(2^{32})$. The present attack is a distinguishing attack with a complexity of order $O(2^{30.79})$.

This is the best among all. A summary is given in Table1.

Table 1: the comparision of attacks complexities

Analyzers	Type ofAttack	Complexity
Borghof,aKnudse, Stolpe	State recovery attack	$O(2^{64.5})$
Raddum	Algebraic Attack	$O(2^{56})$
McDonald, Charnes, Pieprzyk	Guess and Determine Attack	$O(2^{52.3})$
Maximov, Biryukov	State Recovery Attack	$O(2^{51})$
Maximov, Biryukov	Distinguishing Attack	$O(2^{32})$
Our attack	Distinguishing Attack	$O(2^{30.79})$

5.conclusion:

In this article, the trivium algorithm has been scrutinized and its downgraded copy of bivium which is one of the algorithms of eSTREAM project. This project aims at design and analysis of protracted ciphers in order to present a trustworthy collection. After a brief explanation of this algorithm and its structure which encompasses key update, state update and key generation, and with the help of the idea which has been presented in the article plus utilizing appropriate linear approximation and the outcomes of the simulation, I could set the attack

complication at $O(2^{30.79})$ which is more successful and with a lower rate of complication than the best complication set at $O(2^{32})$ which relates to Maximov and Biryukov attack.

References

[1] Claude Shannon. “Communication theory of secrecy systems”. Bell System Technical Journal , 28-4:656-715, 1949.
[2] Alexander Maximov. “Some Words on Cryptanalysis of Stream Ciphers”. Lund University,PH.D. Thesis, 2006.
[3] Golomb, S.W., “Shift Register Sequences”. Holden-Day, San Fransisco, 1967
[4]ECRYPT.theeStream project http://www.ecrypt.eu.org/timetable
[5] Matthew, Robshaw Olivier Billet. “New Stream Cipher Designs: The eSTREAM Finalists”, Lecture Notes in Computer Science, volume 4986, Springer, 2008.
[6] ECRYPT.theeStreamproject : http://www. ecrypt.eu.org/stream
[7] Borghoff, J., Knudsen, L. R., and Stolpe, M., "Bivium as a Mixed-Integer Linear Programming Problem", Lecture Notes in Computer Science (LNCS), Vol. 5921, 2009, pp 133-152.
[8] De Canniere, C., Preneel, B., "TRIVIUM–a stream cipher construction inspired by block cipher design principles", eSTREAM, ECRYPT Stream Cipher Project, Report 2005/030,2005 http://www.ecrypt.eu.org/stream/trivium.html
[9] Maximov, A., Biryukov, A., "Two Trivial Attacks on Trivium", Selected Areas in Cryptography 2007, 2007, pp. 36-55.
[10] Raddum, H.,"Cryptanalytic Results on Trivium", eSTREAM, ECRYPT Stream Cipher Project, Report 2006/039, 2006. http://www .ecrypt.eu.org/stream
[11] McDonald, C., Charnes, C., Pieprzyk, J., " AttackingBivium with MiniSat", Cryptology ePrint Archive, Report 2007/040, 2007
[12]J. Dj. Goli´c. Intrinsic statistical weakness of keystream generators. In J. Pieprzyk and R. Safavi-Naini, editors, Advances in Cryptology — ASIACRYPT’94, volume 917 of Lecture Notes in Computer Science, pages 91–103. Springer-Verlag, 1994.
[13] J. Dj. Goli´c. Linear models for a time-variant permutation generator.IEEE Transactions on Information Theory, 45(7):2374–2382,1999