2010 10th IEEE International Conference on Computer and Information Technology (CIT 2010)

Distinguishing Attack on Bivium

Zainab Noferesti [1] , Neda Rohani [1] , Javad Mohajeri [2] and Mohammad Reza Aref [1]
[1] *Information System and Security Lab (ISSL)*
[2] *Electronics Research Center*
Dept. of Electrical Engineering , Sharif University of Technology
P.O. Box 11365-11155, Tehran, Iran
E-mail:znoferesti@ee.sharif.edu, n_rohani@ee.sharif.edu, mohajer@sharif.ir, aref@sharif.edu

Abstract

Bivium is a simplified version of Trivium, a hardware profile finalist of eSTREAM project. Bivium has an internal state of size 177 bits and a key length of 80 bits. In this paper we introduce a distinguishing attack on this cipher. In this method we first find the best linear approximation for the updating function. Then by using this approximation, and optimizing the time delay, we find the distinguisher. The complexity of the attack is $O(2^{30.79})$, which is an improvement to the previous distinguishing attack with a complexity of order $O(2^{32})$.

Keywords

eSTREAM, stream cipher, distinguishing attack, Bivium

1. Introduction

Cryptography can be defined as the conversion of data into a confidential code that can be decrypted and sent across a public or private channel. Cryptography uses two main styles of encryption: symmetrical and asymmetrical. Symmetric algorithms, use the same key for encryption as they do for decryption.

Symmetric cryptography schemes are generally categorized as being either stream ciphers or block ciphers. Stream ciphers generate a very long keystream and use it to encrypt a single bit, byte or word at a time. A block cipher encrypts one block of data at a time using the same key on each block. In general, the same plaintext block will always encrypt to the same ciphertext when using the same key in a block cipher, whereas the same plaintext will encrypt to different ciphertext in a stream cipher.

Stream ciphers are designed in two modes: self - synchronizing and synchronous. Self-synchronizing stream ciphers calculate each bit in the keystream as a function of the previous *n* bits in the keystream. Synchronous stream ciphers generate the keystream independent of the message stream by using the same keystream generation function at sender and receiver. While stream ciphers do not propagate transmission errors, they are, by their nature, periodic so that the keystream will eventually repeat.

In 2004, European Network of Excellence in Cryptology (ECRYPT) launched a project on stream ciphers, eSTREAM [1], with a focus on introducing practical stream ciphers with an acceptable level of security. The main evaluation criteria were likely to be long-term security, efficiency (performance), flexibility and market requirements. All of the stream ciphers introduced before this project have been cryptanalysed. So the finalists of eSTREAM project are the only present candidates for secure stream ciphers.

The project was launched in two profiles: software and hardware. Software profile candidates should have been stream ciphers for software applications with high throughput requirements and support key lengths of at least 128 bits. Hardware profile candidates should have been stream ciphers for hardware applications with restricted resources such as limited storage, gate count, or power consumption and have security level of at least 80 bits.

Bivium [2] is a simplified version of Trivium [3], a synchronous stream cipher submitted to this project as a hardware profile candidate. Trivium has been selected as one of the portfolio finalists. Bivium has less internal state variables. Security, speed and simplicity are three important characteristics of its design. The previous distinguishing attack on Bivium was performed by Maximov and Biryukov [4], with a compelxity of order $O(2^{32})$.

Due to Bivium's low nonlinearity and existence of linear approximations with good bias in this cipher, we were able to perform a distinguishing attack on this cipher with a complexity of order $O(2^{30.79})$.

978-0-7695-4108-2/10 $26.00 © 2010 IEEE
DOI 10.1109/CIT.2010.196

Our attack is a combination of theoretical reasoning and simulations. We performed the simulations in C++ by running the cipher for a subset of secret keys and initialization vector (IV)s.

This paper is organized as follows. In the next Section a description of Bivium stream cipher is given. In Section 3 our distinguishing attack on Bivium is described and Section 4 provides a comparison between related work. Section 5 concludes this paper.

2. Cipher Specification

Bivium is a bit-oriented stream cipher with an internal state of 177 bits, initialized by an 80-bit key and an 80-bit IV during an initialization phase. IV is used to increase the entropy of the cipher[5]. In every step two bits are updated according to a nonlinear update function and the others are updated as in a linear shift register. Throughout this document '⊕' and '.' operators will denote addition and multiplication over GF(2), respectively.

2.1. Keystream Generation

Denoting the state variables in clock time *t* by $(s_t^1, s_t^2, \ldots, s_t^{177})$ the keystream generation is described according to the following pseudo-code as shown in Figure 1.

Keystream Generation Pseudocode

for i = 1 to N do

$r_1 \leftarrow s_{66} \oplus s_{93}$

$r_2 \leftarrow s_{162} \oplus s_{177}$

$z_i \leftarrow r_1 \oplus r_2$

$r_1 \leftarrow r_1 \oplus s_{91}.s_{92} \oplus s_{171}$

$r_2 \leftarrow r_2 \oplus s_{175}.s_{176} \oplus s_{69}$

$(s_1, s_2, \ldots, s_{93}) \leftarrow (r_2, s_1, \ldots, s_{92})$

$(s_{94}, s_{95}, \ldots, s_{177}) \leftarrow (r_1, s_{94}, \ldots, s_{176})$

end for

Figure 1. The algorithm of keystream generation

We will denote $(s_1, s_2, \ldots, s_{93})$ by $(a_1, a_2, \ldots, a_{93})$ and $(s_{94}, s_{95}, \ldots, s_{177})$ by $(b_1, b_2, \ldots, b_{84})$. So the above relations will take the following form:

$$a_{t+1}^1 \leftarrow b_t^{69} \oplus b_t^{84} \oplus a_t^{69} \oplus b_t^{82}.b_t^{83} \quad (1)$$

$$b_{t+1}^1 \leftarrow a_t^{66} \oplus a_t^{93} \oplus b_t^{78} \oplus a_t^{91}.a_t^{92} \quad (2)$$

for i=92:-1:1

$a_{t+1}^{i+1} = a_t^i$

end

for i=83:-1:1

$b_{t+1}^{i+1} = b_t^i$

end

$$z_t \leftarrow a_t^{66} \oplus a_t^{93} \oplus b_t^{69} \oplus b_t^{84} \quad (3)$$

2.2. Initialization

The initialization process is done by first loading the internal state bits with key and IV as shown in Figure 2:

Loading the Internal State Bits

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

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

Figure 2. Loading the internal state bits

and then running the cipher 4×177 times without generating any output. The key generation process is illustrated in Figure 3[4], where Bivium is illustrated as a truncated version of Trivium cipher.

In the next Section the distinguishing attack on Bivium is described.

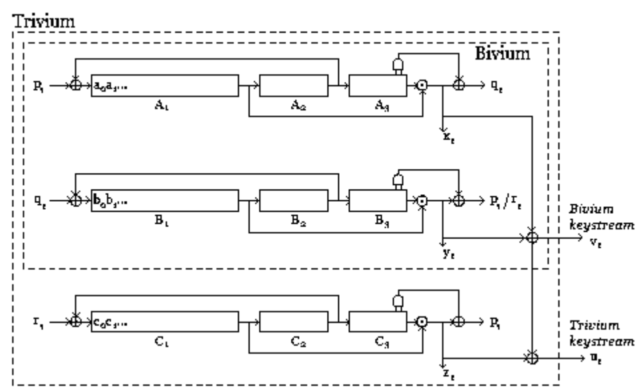

Figure 3. Bivium and Trivium stream ciphers [4]

3. The New Distinguishing Attack on Bivium

Distinguishing attacks, are attacks in which the attacker tries to distinguish the output sequence of a cipher from a random one [6]. As some examples of this type of attack we can mention distinguishing attacks on Py [7], NLS [8] and Grain [9].

Our method consists of three steps as follows:

Step 1. In the first step we try to find a linear approximation for the nonlinear updating function. As the truth table in Figure 4 verifies, the multiplication of two bits can be substituted with zero, with a probability of 3/4.

a	b	a.b
0	0	0
0	1	0
1	0	0
1	1	1

Figure 4. Truth Table for multiplication of two bits

Therefore reducing the nonlinear updating relations (1) and (2) to:

$$a_{t+1}^{1} = b_t^{69} \oplus b_t^{84} \oplus a_t^{69} \quad (4)$$
$$b_{t+1}^{1} = a_t^{66} \oplus a_t^{93} \oplus b_t^{78} \quad (5)$$

According to (3) the sum of the two first columns on the right hand side of each of equations (4) and (5) results in z_t.

Step 2. Then we try to find a time delay for which the sum of the remaining terms of (4) and (5) would be equal to the delayed sum $b_{t+\tau}^{69} \oplus b_{t+\tau}^{84} \oplus a_{t+\tau}^{66} \oplus a_{t+\tau}^{93}$, namely:

$$a_{t+1}^{1} \oplus a_t^{69} \oplus b_{t+1}^{1} \oplus b_t^{78} = b_{t+\tau}^{69} \oplus b_{t+\tau}^{84} \oplus a_{t+\tau}^{66} \oplus a_{t+\tau}^{93}\ , \quad (6)$$

with highest possible bias.

We examined several τ values by running the cipher for a subset of secret keys and 2^{15} IVs, to find the best time delay. We found out that $\tau = 46$ gives the best bias.

Step 3. Next we define the following events:

A : $a_{t+1}^{1} \oplus a_t^{69} = b_t^{69} \oplus b_t^{84}$
B : $b_{t+1}^{1} \oplus b_t^{78} = a_t^{66} \oplus a_t^{93}$
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 : $z_t \oplus z_{t+\tau} = 0$

According to the relation between the events A, B and C, and considering the binary nature of the variables, we have:

$$P(C) = P(A \cap B) + P(A' \cap B'). \quad (7)$$

By simplifying (7), relation (8) is resulted:

$$P(C) = 1 - P(A) - P(B) + 2 \times P(A \cap B). \quad (8)$$

So in the next step we should try to find the last probability. Our simulation shows that $P(A \cap B) = 0.5624$, so that P(C) will be equal to 0.6258.

In the next step we try to find P(E). Since all of the variables are binary:

$$P(E) = P(C \cap D) + P(C' \cap D'). \quad (9)$$

According to Bayes' theorem, (9) is equal to:

$$P(E) = P(D|C) \times P(C) + P(D'|C') \times P(C'). \quad (10)$$

Our simulations show that $P(D|C) = 0.4016$ and $P(D'|C') = 0.6644$. Replacing these values in (10) results in $P(E) = 0.49999$.

The bias of the distinguisher is given by $\varepsilon = |1-2\times P|$, where P is the probability of the relation. In this case $\varepsilon = 2.32083\times10^{-5}$. In a distinguishing attack usually $O(\varepsilon^{-2})$ samples of the keystream are needed to distinguish the keystream from a random sequence with a high success rate[10].

So the complexity of our attack is $O(2^{30.79})$. Since the base 2 logarithm of this complexity is less than the key length, our attack can be claimed to be a successful one and as will be shown in the next section, an improvement to all previous attacks on Bivium.

Finally we run the cipher to find the real correlation between z_t and z_{t+46} for 10^{10} bits of the keystream, in order to verify the previous results. The correlation is found to be 2.46168×10^{-5} which is very close to the bias found by our method.

4. Related Works

In [2] 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 $2^{64.5}$ seconds. In [11] 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 [12] introduce a type of guess and determine attack on Bivium with a complexity of approximately $O(2^{52.3})$. In [5] 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 latter is performed by finding a way of sampling from the keystream such that their distribution is biased. The present attack is a distinguishing attack with a complexity of order $O(2^{30.79})$, which is the best among all. A summary is given in Table 1.

5. Conclusion

In this paper we concentrated on a distinguishing attack on Bivium stream cipher, a simplified version of Trivium, one of the hardware profile finalists of eSTREAM project. The attack is based on approximating the nonlinear update function of the cipher with a linear relation. Then by using this approximation, and optimizing the time delay, we find the distinguisher. The final distinguisher has bias 2.32083×10^{-5} and the complexity of the attack is $O(2^{30.79})$. This result is the best among all previous attacks on this cipher.

Table1-Attacks on Bivium

Analyzers	Type of Attack	Complexity
Borghof,Knudse, 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	$O(2^{30.79})$

6. Acknowledgements

This work was partially supported by Iran National Science Foundation (INSF) - cryptography chair and by Iran Telecommunications Research Center (ITRC).

7. References

[1] eSTREAM: eSTREAM – The ECRYPT Stream Cipher Project: *http://www. ecrypt.eu.org/stream*

[2] Borghoff, J., Knudsen, L. R., and Stolpe, M., "Bivium as a Mixed-Integer Linear Programming Problem", *Lecture Notes in Computer Science (LNCS),* Proceedings of the 12th IMA International Conference on Cryptography and Coding Vol. 5921, 2009, pp 133-152.

[3] 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*

[4] Maximov, A., Biryukov, A., "Two Trivial Attacks on Trivium", *Selected Areas in Cryptography*, 14th International Workshop, SAC 2007, Ottawa, Canada, 2007, pp. 36-55.

[5] Hong, J., Sarkar, P., "New applications of time memory data tradeoffs", *Advances in Cryptology - ASIACRYPT 2005*, Vol. 3788 in *Lecture Notes in Computer Science(LNCS)*, pp. 353-372. Springer-Verlag, 2005.

[6] Hakala, R., Linear Cryptanalysis of Two Stream Ciphers, Helsinki University of Technology, Dec. 2007.

[7] Paul, S., Preneel, B., and G. Sekar, "Distinguishing Attack on Stream Cipher Py", eSTREAM – The ECRYPT Stream Cipher Project , Report 2005/081 , *http://www.ecrypt.eu.org /stream/081.html*

[8] Cho, J. Y., Pieprzyk, J., "Linear Distinguishing Attack on NLS", eSTREAM – The ECRYPT Stream Cipher Project, Report 2006/044 , *http://www.ecrypt.eu.org/stream/018.html*

[9] Noferesti, Z., Rohani, N., Mohajeri, J., and Aref, M. R., "Distinguishing Attack on Grain", submitted to *the 7th Conference on Security and Cryptography for Networks (SCN 2010)*, Italy.

[10] Ekdahl P., On LFSR based Stream Ciphers Analysis and Design, Ph.D Thesis, Lund University, 2003.

[11] Raddum, H., "Cryptanalytic Results on Trivium", eSTREAM, ECRYPT Stream Cipher Project, Report 2006/039, 2006. *http://www.ecrypt.eu.org/stream*

[12] McDonald, C., Charnes, C., Pieprzyk, J., "Attacking Bivium with MiniSat", *Cryptology ePrint Archive*, Report 2007/040, 2007.