Cube Attack on Reduced-Round Quavium

Shiyong Zhang[1, a *], Gongliang Chen[1,b] and Jianhua Li[1,c]

[1]School of Information Security Engineering,

Shanghai Jiaotong University, China

[a]poetzhangzi@sjtu.edu.cn, [b]chengl@sjtu.edu.cn, [c]lijh888@sjtu.edu.cn

Keywords: Quavium, Trivium, Security, Cube Attack.

Abstract. Trivium is a notable light-weight synchronous stream cipher submitted to the European eSTREAM project in April 2005. Quavium is a Trivium-like algorithm which is almost as fast as Trivium. In this paper, the security of Quavium is concerned under cube attack, which is one of the best known attack on the reduced round Trivium proposed by Dinur and Shamir at EUROCRYPT 09. Trivium with 576 initialization rounds can be recovered in 2^{11}. We show that it is difficult to search the cubes of Quavium with the same rounds and after 288 rounds the attack complexity is reduced to 2^{59}. Therefore, comparing with Trivium, Quavium has a better performance under cube attack.

Introduction

Trivium is a notable light-weight synchronous stream cipher designed by Christophe De Canniere and Bart Preneel, which is submitted to the European eSTREAM project in April 2005 [1]. This algorithm is designed to be both efficient and secure. During 3 phases of eSTREAM evaluation on the stream cipher proposals, the performance of Trivium is outstanding compared with other stream ciphers such as A5. Trivium outperforms other eSTREAM candidates considered in the paper in terms of the two most important optimization criteria, minimum area and maximum throughput to area ratio, by a factor of at least two [2].

Quavium, proposed by Tian in 2012, is a 4-round Trivium-like algorithm [3]. It is designed based on the Trivium-like shift register. The experimental results on software using C++ code show that the speed of keystream generation of Quavium is nearly the same as that of Trivium and the performance of Quavium both on hardware and software is almost as good as Trivium [3]. However, the security of Quavium is not concerned by Tian.

Since now, there are many works about the security of Trivium. Raddum presents a technique to solve systems of equations associated with Trivium [4]. But his attack is very complex when applied to the full cipher and is no faster than exhaustive search. Borghoff presents a numerical attack [5]. However the estimated time complexity of this attack is about $2^{63.7}$ seconds. Maximov studies two attacks on Trivium [6], which are state recovering and statistical tests. The internal state of Trivium can be recovered in time around $2^{83.5}$, which is still too complex for application.

Cube attack proposed by Dinur and Shamir at EUROCRYPT 09 is one of the best known attack on the reduced round Trivium [7]. Vielhaber try to recover 47 bits of the key after 576 rounds using an algebraic method [8]. Later Dinur and Shamir described a full key recovery in less than 2^{30} queries to Trivium reduced to 735 rounds and also recovered 35 key bits after 767 rounds in about 2^{36} queries [7, 9]. Fouque and Vannet requires 2^{39} queries to break 784 round Trivium [10]. Srinivasan gives 69 equations after 576 rounds to reduce the complexity to 2^{11} [11].

In this paper, the structure of Quavium is studied and we study the security of Quavium under cube attack.

We analysis Quavium reduced to 288 rounds using cube attack gives 21 linear equations, which can recover 21 bits of the key and reduces the attack complexity to 2^{59}.

The following part of the paper is organized as follows. The algorithm of Quavium will be described in section 2. The method of cube attack will be shown in detail in Section 3. Section 4 will compare the security of Trivium and Quavium under cube attack. The conclusion will be given in section 5.

Quavium Algorithm

Quavium is designed to generate up to 2^{64} bits of key stream from an 80-bit secret key Key and an 80-bit initial value IV [3]. The process consists of two phases: first the internal state of the cipher is initialized using Key and IV, then the state is repeatedly updated and used to generate key stream bits. There are 288 bits in the internal state, which is denoted as $\boldsymbol{s} = (s_1, s_2, \mathbf{L}, s_{288})$. Quavium has four rounds with similar structure. Denote the intermediate variable as t_1, t_2, t_3, t_4 and the output stream as $\boldsymbol{z} = (z_1, z_2, \mathbf{L}, z_N)$, with N standing for the number of output bits. The process of Quavium is shown as Algorithm 1:

Algorithm 1 Quavium Algorithm

for i=1 to N do

$t_1 \leftarrow s_3 + s_{51}$

$t_2 \leftarrow s_{57} + s_{108}$

$t_3 \leftarrow s_{126} + s_{195}$

$t_4 \leftarrow s_{204} + s_{288}$

$z_i \leftarrow t_1 + t_2 + t_3 + t_4$

$t_1 \leftarrow t_1 + s_{49} \cdot s_{50} + s_{96}$

$t_2 \leftarrow t_2 + s_{106} \cdot s_{107} + s_{135}$

$t_3 \leftarrow t_3 + s_{193} \cdot s_{194} + s_{228}$

$t_4 \leftarrow t_4 + s_{286} \cdot s_{287} + s_{33}$

$(s_1, s_2, \mathbf{L}, s_{51}) \leftarrow (t_4, s_1, \mathbf{L}, s_{50})$

$(s_{52}, s_{53}, \mathbf{L}, s_{108}) \leftarrow (t_1, s_{52}, \mathbf{L}, s_{107})$

$(s_{109}, s_{110}, \mathbf{L}, s_{195}) \leftarrow (t_2, s_{109}, \mathbf{L}, s_{194})$

$(s_{196}, s_{197}, \mathbf{L}, s_{288}) \leftarrow (t_3, s_{196}, \mathbf{L}, s_{287})$

end for

Key and IV are loaded as follows:

$$s(t) = \begin{cases} (s_1, s_2, \mathbf{L}, s_{51}) \leftarrow (K_1, K_2, \mathbf{L}, K_{51}) \\ (s_{52}, s_{53}, \mathbf{L}, s_{108}) \leftarrow (K_{52}, \mathbf{L}, K_{80}, IV_1, IV_2, \mathbf{L}, IV_{28}) \\ (s_{109}, s_{110}, \mathbf{L}, s_{195}) \leftarrow (IV_{29}, \mathbf{L}, IV_{80}) \\ (s_{196}, s_{197}, \mathbf{L}, s_{288}) \leftarrow (0, \mathbf{L}, 0,1,1,1) \end{cases} \tag{1}$$

Evaluation on implementation of Quavium is given by Tian with comparison to Trivium [3, 14, 15]. The comparison is based on the gate equivalent (GE) count which are shown in Table I. The results show that Quavium extends Trivium to 4 rounds and only increases 8 NAND gates [3].

Table I: Resource Consumption of Trivium and Quavium

Algorithm	Flip-flops	AND gates	XOR gates	total
Trivium	288	3	11	3488
Quavium	288	5	13	3496

Cube Attack

Cube attack is introduced in EUROCRYPT 09 by Dinur and Shamir as chosen IV attack on symmetric primitives [7, 12]. The attack allows one to find linear relations between key bits. Then using simple linear algebra techniques, it is possible to recover the bit values. The process can be described as follows:

In the ring $R = F_2\left[K_1, K_2, \mathbf{L}, K_n, IV_1, IV_2, \mathbf{L}, IV_p\right]$, we consider the polynomial representation of the first output bit of the cipher as the polynomial $P\left(K_1, K_2, \mathbf{L}, K_n, IV_1, IV_2, \mathbf{L}, IV_p\right)$ in R. Given a cube of the public variables $C = \left\{IV_{c_1}, IV_{c_2}, \mathbf{L}, IV_{c_k}\right\}$ of size k, P can be expressed as:

$$P = \prod_{i=1}^{k} IV_{c_i} P_C + P_R \quad (2)$$

where $P_C, P_R \in R$, no monomial of P_R is divisible by $\prod_{i=1}^{k} IV_{c_i}$.

Then we can compute P_C as follows:

$$P_C = \sum_{C} P \quad (3)$$

P_C is called the superpoly yielded by C and $\prod_{i=1}^{k} IV_{c_i}$ is called a maxterm if the superpoly yielded by C is linear. For Quavium, n=p=80.

In the real attack, to test the constant and linearity, the most common linearity test for polynomials is the BLR(Blum Luby Rubinfeld) test [13]. Given a black-box polynomial P on n variables one wants to test for linearity, the BLR test requires the computation on random inputs X and Y, on the 0 vector and on X+Y. One then simply checks whether P(0)+P(X)+P(Y)+P(X+Y)=0.

The algorithm of searching the maxterm on Quavium is given in Algorithm 2.

Algorithm 1 Cube Attack on Quavium

Select the Cube C randomly.

Select $X, Y \in \{0,1\}^n$, compute P(0), P(X), P(Y),P(X+Y)$ and check whether P(0)+P(X)+P(Y)+P(X+Y)=0. Test for more than 200 times.

for i=1 to n do

Denote $e_i = (0, 0, \mathbf{L}, 0, 1, 0, \mathbf{L}\ 0)$, where all the variants are zero expect the ith bit.

Compute $P(e_i)$

end for

The maxterm can be expressed as $P(0) + \sum_{i=1}^{n} P(e_i) K_i$

Attack Results

We first try to search the maxterm on Quavium after 576 rounds. However, we fail to find any superpoly. In fact, if P has a low-enough degree, even though it has a large number of variables, it is possible to find the linear maxterms. However, if P is a uniformly random polynomial of high degree, then it is extremely unlikely that there exists maxterms. For Trivium, the polynomials are expected to retain a low degree even after hundreds of initialization rounds. However for Quavium, the

polynomials increases to a very high degree only in one hundred rounds. Therefore, we search the maxterm on Quavium after 288 rounds. The maxterms and the cube indexes are listed in Table II.

Table II: Linear superpolys for Quavium with 288 Initialization rounds

Cube Indexes	Maxterm	Cube Indexes	Maxterm
{5,14,19,20,21,27,29,53,42,43}	K_1	{3,4,5,9,14,20,21,27,42,79}	$1+K_{21}$
{6,7,17,27,28,48,60,69,74,75}	$1+K_4+K_{19}+K_{54}$	{3,4,7,9,14,20,21,42,43,65}	$1+K_{22}+K_{52}$
{4,5,7,19,20,37,53,74,78,79}	$1+K_5+K_{11}+K_{54}$	{3,7,19,20,21,27,42,43,74,79}	$1+K_{23}$
{3,4,5,19,20,21,27,29,42,53}	$1+K_8$	{3,4,5,19,20,27,29,42,53,79}	$1+K_{31}$
{7,10,11,36,41,62,63,69,70,72}	$1+K_{13}$	{9,14,19,20,21,27,35,42,43,65}	$1+K_{32}$
{4,5,9,14,19,20,35,42,43,53,79}	$1+K_{14}$	{3,4,5,9,19,20,27,35,43,79}	$1+K_{33}$
{2,15,16,27,28,36,48,72,74,75}	$1+K_{15}$	{9,14,19,20,21,27,29,35,53,79}	K_1+K_{52}
{14,19,20,21,27,29,35,37,42,53}	K_1+K_{16}	{3,4,5,14,19,20,27,29,35,53}	K_1+K_{57}
{3,4,5,7,20,21,29,42,78,79}	$1+K_{18}+K_{21}$	{4,5,9,14,19,20,21,27,35,43}	$K_1+K_{19}+K_{52}+K_{65}$
{6,7,15,16,27,31,48,60,74,75}	$1+K_{19}$	{3,9,19,20,21,27,29,42,43,79}	$1+K_3+K_{21}+K_{54}+K_{57}$
{3,7,9,14,19,20,35,42,43,57}	$1+K_{20}$		

We compare Trivium and Quavium under cube attack [10, 11]. The result is shown in Table III.

Table III: Comparison of Two Algorithm Under Cube Attack

Algorithm	Initialization rounds	Breaking Complexity
Trivium	576	2^{11}
Trivium	799	2^{62}
Quavium	288	2^{59}

From the result, it can be seen that compared to the 3-round Trivium, Quavium have better performance of security due to more internal rounds. Furthermore, the degree of equations of Quavium increases more faster than the degree of Trivium. It is difficult to attack Quavium after 576 rounds. After 288 rounds the breaking complexity is still 2^{59} which is higher than Trivium after 576 rounds. Therefore, Quavium has a better performance under cube attack than Trivium.

Conclusion

In this paper, we study the internal structure of Quavium and the security of Quavium under cube attack. We try to recover the secret key of Quavium with reduced round, given a piece of a known keystream. We show that Quavium after 288 rounds can be recovered in time around 2^{59}, while for Trivium after 576 rounds the complexity is 2^{11}. Therefore, comparing with Trivium, Quavium has a better performance under cube attack.

Acknowledgment

This work was supported in part by International Researcher Exchange Project of National Science Foundation of China and Centre national de la recherche scientifique de France (NSFC-CNRS) under Grant No. 61211130104 and National Science Foundation of China under Grants No. 61271220.

References

[1] C. De Cannire and B. Preneel. TRIVIUM Specifications. eSTREAM, ECRYPT Stream Cipher Project (http://www.ecrypt.eu.org/stream), Report 2005/030, April 2005.

[2] K. Gaj, G. Southern and R. Bachimanchi, "Comparison of hardware performance of selected phase II eSTREAM candidates", http://www.ecrypt.eu.org/stream/papersdir/200 7/026.pdf, 2007.

[3] Tian, Y., Chen, G., Li, J. "QUAVIUM - a new stream cipher" TRIVIUM. J. Comput, pp 1278-1283 (2012).

[4] H. Raddum, "Cryptanalytic results on Trivium", http://www.ecrypt.eu.org/stream/papersdir/2006/039.ps, 2007.

[5] J. Borghoff, L. R. Knudsen and M. Stolpe, "Bivium as a mixed-integer linear programming problem", in LNCS vol.5921, M. G. Parker Eds. Heidelberg: Springer, 2009, pp. 133-152, 2009.

[6] A. Maximov, A. Biryukov. "Two trivial attacks on TRIVIUM", in SASC2007: The State of the Art of Stream Ciphers, pp. 1-16, 2007.

[7] Dinur,I.,Shamir,A. "Cube Attacks on weakable Black Box Polynomials." in Joux,A. (ed.) EUROCRYPT 2009. LNCS, Springer, Heidelberg, vol. 5479, pp. 278-299, 2009.

[8] M.Vielhaber "Breaking one.fivium by aida an algebraic iv differential attack", Cryptology ePrint Archive, Report 2007/413, 2007, http://eprint.iacr.org/.

[9] J.-P. aumasson, I. Dinur, W. Meier, and a. Shamir, "Cube testers and key recovery attacks on reduced-round MD6 and trivium," in Fast Software Encryption, 2009, pp. 1-22.

[10] Fouque, P.A., Vannet, T, "Improving Key Recovery to 784 and 799 rounds of Trivium using Optimized Cube Attacks." inMoriai, S. (ed.) FSE 2013. LNCS, Springer, Heidelberg, vol. 8424, pp. 502-517, 2014.

[11] Chungath Srinivasana, Utkarsh Umesan Pillaia, K.V. Lakshmya and M. Sethumadhavan "Cube Attack on Stream Ciphers using a Modified Linearity Test" in Journal of Discrete Mathematical Sciences and Cryptography, 2015, pp 301-311.

[12] Dinur,I.,Shamir,A. "applying cube attacks to stream ciphers in realistic scenarios," Cryptography and Communications, vol. 4, pp. 217-232, 2012.

[13] M. Blum, M. Luby et R. Rubinfeld - "Self-testing/correcting with applications to numerical problems", Proceedings of the twenty-second annual ACM symposium on Theory of computing (New York, NY, USA), STOC 90, ACM, 1990, pp 73-83.

[14] M. Feldhofer and J. Wolkerstorfer. "Hardware Implementation of Symmetric Algorithms for RFID Security". in RFID Security: Techniques, Protocols and System-on-Chip Design, pp. 373-415. Springer, September 2008.

[15] M. Feldhofer. "Comparison of Low-Power Implementations of Trivium and Grain". \textit{Workshop on The State of the Art of Stream Ciphers (SASC2007)} pp 236-246, 2007.