Synthetic Linear Analysis with Applications to CubeHash and Rabbit*

Yi Lu[1], Serge Vaudenay[2], and Willi Meier[3]

[1] National Engineering Research Center of Fundamental Software,
Institute of Software, Chinese Academy of Sciences, Beijing, China
[2] EPFL, Lausanne, Switzerland
[3] FHNW, Windisch, Switzerland

Abstract. In linear cryptanalysis, it has been considered most important and difficult to analyze the bias and find a large bias. The demonstration of a large bias will usually imply that the target crypto-system is not strong. Regarding the bias analysis, researchers tend to look for a theoretical solution for a specific problem. In this paper, we take a first step towards the synthetic approach on bias analysis. We successfully apply our synthetic analysis to improve the most recent linear attacks on CubeHash and Rabbit respectively.

CubeHash was selected to the second round of SHA-3 competition. The best linear attack on 11-round CubeHash with 2^{470} queries was proposed in [2]. We present an improved attack for 11-round CubeHash with complexity $2^{414.2}$. Based on our 11-round attack, we give a new linear attack for 12-round CubeHash with complexity 2^{509}. It is the first known attack on 12 rounds with complexity below the security parameter 2^{512} of CubeHash.

Rabbit is a stream cipher among the finalists of ECRYPT Stream Cipher Project (eSTREAM). It has also been published as informational RFC 4503 with the Internet Engineering Task Force (IETF), which is the main standardization body for Internet technology. For Rabbit, the best linear attack with complexity 2^{141} was recently presented in [9]. Our synthetic bias analysis yields the improved attack with complexity 2^{136}.

Keywords: bias, linear cryptanalysis, synthetic analysis, conditional dependence, CubeHash, Rabbit.

1 Introduction

It is one of most important and difficult problems to analyze the bias and find a large bias regarding the security of crypto-systems, since the

* Preliminary version appeared in the proceedings [10] of ICISC 2011, the 14th Annual International Conference on Information Security and Cryptology. In this paper, the new advanced two-round bias analysis for CubeHash is added; further, we give an improved 12-round attack with reduced complexity 2^{509}, compared with 2^{513} in [10].

invention of linear cryptanalysis [11] almost 20 years ago. The demonstration of a large bias will usually imply that the target crypto-system is not as strong as expected. Regarding the bias analysis, researchers often focus on a theoretical solution for a specific problem. For a broad class of problems, this does not help much to analyze the bias unfortunately.

Most often, we need to study the combined bias of multiple Boolean functions (such as multiple linear approximations) with many input variables. Assuming that these Boolean functions are all independent pairwise, the problem reduces to the bias computation of each Boolean function separately. Apparently, if the terms involved in each Boolean function are statistically independent of the terms in the others, we are sure that all are independent pairwise and it is "safe" to concentrate on bias computation of each Boolean function. Further, it is worth pointing out that it is incorrect to conclude independence when the terms involved in each function appear to be different from the terms occurring in the others. It is thus essential to conduct synthetic analysis to study the bias problems. In this paper, we take a first step towards the synthetic approach on bias analysis. We also propose a conditional dependent bias problem and we give the analysis to estimate the bias.

We apply our synthetic analysis to improve the most recent linear attack [2] on the hash function CubeHash [4]. CubeHash was selected to the second round of SHA-3 competition [12]. In [2], based on the bias analysis for 11-round CubeHash, the best linear attack on 11-round CubeHash with 2^{470} queries was proposed. Our results improve the bias analysis of [2]. We show the largest bias $2^{-207.1}$ for 11-round CubeHash, and we present an improved linear attack for 11-round CubeHash with complexity $2^{414.2}$. Further, based on our 11-round attack, we give a new linear attack for 12-round CubeHash with complexity 2^{509}. This is the first known attack on 12 rounds with complexity below the critical security parameter[4] 2^{512} of CubeHash.

Meanwhile, our synthetic analysis is applied to the recent linear attack [9] on stream cipher Rabbit [5]. Rabbit is a stream cipher among the finalists of ECRYPT Stream Cipher Project (eSTREAM). It has also been published as informational RFC 4503 with the Internet Engineering Task Force (IETF), which is the main standardization body for Internet technology. In [9], the best linear attack with complexity 2^{141} was presented. As reference, Rabbit designers claim the security level of 2^{128} ([5, Page 18]). Our synthetic analysis applies to the main part of the bias

[4] As reference, we note that for the 1024-bit internal state size like CubeHash, the critical security parameter is 2^{512}, which implies a collision attack.

analysis [9]. Our results yield the improved linear attack with complexity 2^{136}.

The rest of the paper is organized as follows. In Section 2, we introduce the idea of synthetic approach to linear analysis; we also propose the synthetic bias analysis for a conditional dependent problem. In Section 3, we discuss step by step how to apply synthetic analysis to CubeHash round function and present our improved attacks on CubeHash. Our improved new result is the first known attack on 12-round CubeHash with complexity below the critical security parameter 2^{512}. In Section 4, we apply our conditional dependent problem to improve the best known linear attack on Rabbit. Finally, we conclude in Section 5.

2 Synthetic Bias Analysis

2.1 The Idea

Define the bias of a binary random variable A by

$$\mathrm{bias}(A) \overset{\mathrm{def}}{=} \Pr(A=0) - \Pr(A=1). \tag{1}$$

The range of bias is always between -1 and +1. Given multiple Boolean functions, e.g., linear approximations, $f_1(X_1), f_2(X_2), \ldots, f_k(X_k)$, for the fixed k, we often need to study the combined bias, i.e., the bias for $f_1(X_1) \oplus f_2(X_2) \oplus \cdots \oplus f_k(X_k)$. It is common and convenient to assume that the inputs $X_1, X_2, \ldots, X_k$ are all statistically independent pairwise. Thus, with our definition on bias in (1), the bias of the combination of independent Boolean functions (such as linear approximations) is equal to the product of the individual biases as asserted by Piling-up lemma [11]. Consequently, the bias problem reduces to the bias computation of each Boolean function $f_i(X_i)$ (for $i = 1, 2, \ldots, k$) separately.

When it comes to the bias computation of a single Boolean function (e.g., linear approximation), in general, it is a hard problem, although in certain cases it might be feasible to calculate the bias. For example, [6, 13] is applicable to analyze the bias of a single linear approximation in our CubeHash problem. Nevertheless, when the bias is large[5], we can always compute it empirically, as successfully showed with recent results on RC4 biases (e.g., [14]).

Henceforth, it leads us to resort to the well-known approach of Divide-and-Conquer method to the bias analysis involving multiple Boolean functions, when the bias is small and direct computing is infeasible. That is, we

[5] The bias computation when the bias is small is beyond the scope of this paper.

try to group multiple dependent Boolean functions together (e.g., linear approximations with regards to CubeHash). The aim is that the Boolean functions in each group are dependent jointly and the Boolean functions in different groups are all independent. For example, suppose we have three groups with functions $f_1(X_1)$, $f_2(X_2)$,..., $f_9(X_9)$. Group One contains $f_1(X_1)$, $f_2(X_2)$, $f_3(X_3)$, Group Two contains $f_4(X_4)$, $f_5(X_5)$, $f_6(X_6)$ and Group Three contains $f_7(X_7)$, $f_8(X_8)$, $f_9(X_9)$. Then, $f_1(X_1), f_2(X_2)$ (both from Group One) are not necessarily independent; $f_1(X_1)$, $f_5(X_5)$, $f_7(X_7)$ (each from a different group) are *all* independent; $f_2(X_2)$, $f_6(X_6)$, $f_7(X_7)$ are *all* independent, and so on.

When grouping, it is desirable to make each group size as small as possible, where the group size is referred to as the number of Boolean functions contained in the group. The rationale behind grouping is that, we are dividing originally one (big) group of a larger number of functions into multiple independent groups; once grouping is done, we just need to study the combined bias for each group of smaller size individually. This helps make the task of bias analysis easier by reducing the number of the functions, which have to be studied simultaneously. In our above example for instance, computing the combined bias for $f_1(X_1)$, $f_2(X_2)$, ..., $f_9(X_9)$ might turn out to be impossible. After we divide into three groups, our problem reduces to computing the combined bias for three groups individually. When the combined bias for each group is large enough, we can compute it empirically, or if the input space is not too large we can even compute it directly. In Section 3, we will apply this technique to analyze CubeHash.

2.2 Bias in the Conditional Dependent Problem

Let X, Y, Z be random variables. X, Y are statistically independent variables, but X, Z are dependent as well as Y, Z. We say that X, Y are conditional dependent[6] given Z. We are concerned with the combined bias of $f_1(X) \oplus f_2(Y) \oplus f_3(Z)$. For convenience, we say $f_1(X)$, $f_2(Y)$ are conditional dependent given $f_3(Z)$ rather than that X, Y are conditional dependent given Z. Formally speaking, we consider that u_0, u_1, u_2, v_1, v_2 are independent variables of binary strings (of fixed length). Three Boolean functions $f_A(u_0, u_1, u_2)$, $f_B(u_2, v_2)$, $f_C(u_1, v_1)$ are defined over those variables. They are denoted by A, B, C in short. We assume that we know the

[6] Note that this is in contrast to the concept of conditional independence in statistics. Recall that X, Y are conditional independent given Z, if X, Y, Z satisfy $\Pr(X = x, Y = y|Z) = \Pr(X = x|Z)\Pr(Y = y|Z)$ for all x, y.

bias for A, B, C respectively. We want to estimate the bias for $A \oplus B \oplus C$, and due to the dependence we do not want to use the Piling-up approximation. Our first solution is to obtain the bias for $A \oplus C$ (or $A \oplus B$). Then, estimate the bias for $A \oplus B \oplus C$ by taking either $\text{bias}(A \oplus C) \cdot \text{bias}(B)$, or $\text{bias}(A \oplus B) \cdot \text{bias}(C)$.

By considering the functions as black-boxes (of random functions), we propose to use the heuristics and make a more delicate estimate as follows. As u_1 affects both $A \oplus B$ and C, we make a simple assumption about the two distributions of the bias for $A \oplus B$ and for C over u_1: the absolute value of the bias is (almost) a constant and can only take values in a set of two elements. Thus, it leads us to compute the average p_+ (resp. p_-) of the positive (resp. negative) biases for $A \oplus B$ over randomly chosen u_1 and the percentage q of the positive biases for $A \oplus B$ over randomly chosen u_1. Similarly, we also compute the average of the positive p'_+ (resp. negative p'_-) biases for C over randomly chosen u_1 and the percentage q' of the positive biases. The distribution of the bias for $A \oplus B$ over u_1 is independent of the distribution of the bias for C over u_1, so we combine the results and give an estimate on the bias of $A \oplus B \oplus C$ by

$$qq'p_+p'_+ + (1-q)(1-q')p_-p'_- - q(1-q')p_+p'_- - (1-q)q'p_-p'_+ \tag{2}$$

3 Our Analysis on CubeHash

3.1 Preliminaries

The hash function CubeHash [4] was designed by D.J. Bernstein. It was one of the 14 candidates which were selected to the second round of SHA-3 competition [12]. SHA-3 was initiated by the U.S. National Institute of Standards and Technology to push forwards the development of a new hash standard, following the recent fruitful research work on the hash function cryptanalysis. CubeHash is a family of cryptographic hash functions, parameterized by the performance and security requirement. At the heart of it, CubeHash consists of an internal state of 1024 bits, round transformation T, round number r, between introduction of new message blocks. At the end, T is repeated $10r$ times before outputting h bits of its state as the final hash value. Security/performance tradeoffs are provided with different combinations h, r and the message block length b. The normal security parameters are $r = 16, b = 32$, according to [4]. Note that for CubeHash (regardless of r, b, h), we consider 2^{512} as one critical security parameter for reference, because CubeHash has a compression function with the 1024-bit internal state size, which implies that a generic collision attack is possible due to the birthday attacks.

We let CubeHash internal states of 1024 bits be represented by 32 words $x_{00000}, x_{00001}, \ldots, x_{11111}$ with each word being 32-bit long. CubeHash round transformation T can be described by the following 10 steps of operations:

Step 1: add (modulo 2^{32}) x_{0n} into x_{1n}, i.e.,
$x_{0n} + x_{1n} \to x_{1n}$ for all 4-bit n

Step 2: rotate x_{0n} left by 7 bits, i.e.,
$x_{0n} \lll 7 \to x_{0n}$ for all 4-bit n

Step 3: swap x_{00n} with x_{01n}, i.e.,
$x_{00n} \leftrightarrow x_{01n}$ for all 3-bit n

Step 4: xor x_{1n} into x_{0n}, i.e.,
$x_{0n} \oplus x_{1n} \to x_{0n}$ for all 4-bit n

Step 5: swap $x_{1jk0m}$with x_{1jk1m}, i.e.,
$x_{1jk0m} \leftrightarrow x_{1jk1m}$ for all 1-bit j, k, m

Step 6: add (modulo 2^{32}) x_{0n} into x_{1n}, i.e.,
$x_{0n} + x_{1n} \to x_{1n}$ for all 4-bit n

Step 7: rotate x_{0n} left by 11 bits, i.e.,
$x_{0n} \lll 11 \to x_{0n}$ for all 4-bit n

Step 8: swap x_{0j0km} with x_{0j1km}, i.e.,
$x_{0j0km} \leftrightarrow x_{0j1km}$ for all 1-bit j, k, m

Step 9: xor x_{1n} into x_{0n}, i.e.,
$x_{0n} \oplus x_{1n} \to x_{0n}$ for all 4-bit n

Step 10: swap x_{1jkm0} with x_{1jkm1}, i.e.,
$x_{1jkm0} \leftrightarrow x_{1jkm1}$ for all 1-bit j, k, m

Within the round, x_n will denote the initial state before the round begins, and x_n^i will denote the state x_n after Step i, for all $1 \leq i \leq 10$. The round number of the internal states which we study is clear from the context, and we omit it from the notations.

Recently, [1, 2] studied distinguishing attacks on CubeHash compression function. In [2], linear cryptanalysis technique [11] is used to construct the distinguisher; while in [1], the relatively new technique of rotational analysis is used. We refer to [7] for a survey of cryptanalysis results on CubeHash. In this section, we will investigate the largest bias [2] for

CubeHash. It was shown that due to this largest bias, a non-trivial linear attack on 11-round CubeHash with 2^{470} queries exists [2]. We will conduct synthetic analysis to improve the bias analysis of multiple linear approximations in [2]. Each round of CubeHash consists of two half rounds with five steps. For each half round, only one step introduces nonlinearity to the internal state by performing the modular addition operations. Our main focus is that, within each round, the multiple linear approximations are *not* necessarily all independent pairwise. This can be justified by the fact that nonlinearity is introduced by two separate steps (i.e., Step 1 and Step 6) instead of one single step within a round.

3.2 Our Detailed Analysis on CubeHash Round Function

Let us start from a simple case of Round 7 now. Note that a similar analysis is applicable to all other rounds. We give our analysis for CubeHash Round 8 in Appendix A and complete results in Section 3.3. We want to analyze the following bias for the linear approximation at Round 7,

$$\begin{aligned}&\text{0x300} \cdot (x_{10100} \oplus x_{10110}) \approx \text{0x180000} \cdot (x^{10}_{00000} \oplus x^{10}_{00010} \oplus x^{10}_{10001} \oplus x^{10}_{10011})\\ &\oplus \text{0x300} \cdot (x^{10}_{10101} \oplus x^{10}_{10111}) \oplus \text{0x18000} \cdot (x^{10}_{11101} \oplus x^{10}_{11111}) \qquad (3)\end{aligned}$$

We begin with a trivial equation

$$\begin{aligned}\text{0x300} \cdot (x_{10100} \oplus x_{10110}) = \text{0x300} \cdot (&x_{10100} \oplus x_{00100} \oplus x_{00100}\\ &\oplus x_{10110} \oplus x_{00110} \oplus x_{00110})\end{aligned}$$

We introduce the two linear approximations at Step 1,

$$\begin{aligned}&\text{0x300} \cdot (x_{10100} \oplus x_{00100})\\ &\approx \text{0x300} \cdot (x_{10100} + x_{00100}) = \text{0x300} \cdot x^{1}_{10100} \qquad (4)\\ &\text{0x300} \cdot (x_{10110} \oplus x_{00110})\\ &\approx \text{0x300} \cdot (x_{10110} + x_{00110}) = \text{0x300} \cdot x^{1}_{10110} \qquad (5)\end{aligned}$$

Note that the bias for linear approximation (4), (5) is 2^{-1} respectively by [6, Corollary 1, Page 232]. We have

$$\text{0x300} \cdot (x_{10100} \oplus x_{10110}) \approx \text{0x300} \cdot (x^{1}_{10100} \oplus x^{1}_{10110} \oplus x^{1}_{00100} \oplus x^{1}_{00110}) \quad (6)$$

Going from Step 2 through Step 5, we continue on (6)

$$\begin{aligned}
&= \text{0x300} \cdot x^2_{10100} \oplus \text{0x300} \cdot x^2_{10110} \oplus \text{0x18000} \cdot x^2_{00100} \oplus \text{0x18000} \cdot x^2_{00110} \\
&= \text{0x300} \cdot x^3_{10100} \oplus \text{0x300} \cdot x^3_{10110} \oplus \text{0x18000} \cdot x^3_{01100} \oplus \text{0x18000} \cdot x^3_{01110} \\
&= \text{0x300} \cdot x^4_{10100} \oplus \text{0x300} \cdot x^4_{10110} \oplus \text{0x18000} \cdot x^4_{01100} \oplus \text{0x18000} \cdot x^4_{11100} \oplus \\
&\quad \text{0x18000} \cdot x^4_{01110} \oplus \text{0x18000} \cdot x^4_{11110} \\
&= \text{0x300} \cdot x^5_{10110} \oplus \text{0x300} \cdot x^5_{10100} \oplus \text{0x18000} \cdot x^5_{01100} \oplus \text{0x18000} \cdot x^5_{11110} \oplus \\
&\quad \text{0x18000} \cdot x^5_{01110} \oplus \text{0x18000} \cdot x^5_{11100} \\
&= \text{0x300} \cdot x^5_{10110} \oplus \text{0x300} \cdot x^5_{00110} \oplus \text{0x300} \cdot x^5_{00110} \oplus \text{0x300} \cdot x^5_{10100} \oplus \\
&\quad \text{0x300} \cdot x^5_{00100} \oplus \text{0x300} \cdot x^5_{00100} \oplus \text{0x18000} \cdot x^5_{01100} \oplus \\
&\quad \text{0x18000} \cdot x^5_{11100} \oplus \text{0x18000} \cdot x^5_{01110} \oplus \text{0x18000} \cdot x^5_{11110} \qquad (7)
\end{aligned}$$

At Step 6, four linear approximations are introduced:

$$\begin{aligned}
\text{0x300} \cdot (x^5_{10110} \oplus x^5_{00110}) &\approx \text{0x300} \cdot (x^5_{10110} + x^5_{00110}) && (8) \\
\text{0x300} \cdot (x^5_{10100} \oplus x^5_{00100}) &\approx \text{0x300} \cdot (x^5_{10100} + x^5_{00100}) && (9) \\
\text{0x18000} \cdot (x^5_{01100} \oplus x^5_{11100}) &\approx \text{0x18000} \cdot (x^5_{01100} + x^5_{11100}) && (10) \\
\text{0x18000} \cdot (x^5_{01110} \oplus x^5_{11110}) &\approx \text{0x18000} \cdot (x^5_{01110} + x^5_{11110}) && (11)
\end{aligned}$$

Note that the bias for linear approximation (8), (9), (10), (11) is 2^{-1} respectively by [6, Corollary 1, Page 232]. Because of the step operation, the right hand-sides of (8), (9), (10), (11) are equal to $\text{0x300} \cdot x^6_{10110}$, $\text{0x300} \cdot x^6_{10100}$, $\text{0x18000} \cdot x^6_{11100}$ and $\text{0x18000} \cdot x^6_{11110}$ respectively. So, (7) can be approximated by

$$\text{0x300} \cdot (x^6_{10110} \oplus x^6_{00110} \oplus x^6_{10100} \oplus x^6_{00100}) \oplus \text{0x18000} \cdot (x^6_{11100} \oplus x^6_{11110}).$$

It is equal to

$$\begin{aligned}
&\text{0x300} \cdot x^7_{10110} \oplus \text{0x180000} \cdot x^7_{00110} \oplus \text{0x300} \cdot x^7_{10100} \oplus \text{0x180000} \cdot x^7_{00100} \\
&\oplus \text{0x18000} \cdot x^7_{11100} \oplus \text{0x18000} \cdot x^7_{11110} \\
= {} &\text{0x300} \cdot x^8_{10110} \oplus \text{0x180000} \cdot x^8_{00010} \oplus \text{0x300} \cdot x^8_{10100} \oplus \text{0x180000} \cdot x^8_{00000} \\
&\oplus \text{0x18000} \cdot x^8_{11100} \oplus \text{0x18000} \cdot x^8_{11110} \\
= {} &\text{0x300} \cdot x^9_{10110} \oplus \text{0x180000} \cdot x^9_{00010} \oplus \text{0x180000} \cdot x^9_{10010} \oplus \text{0x300} \cdot x^9_{10100} \\
&\oplus \text{0x180000} \cdot x^9_{00000} \oplus \text{0x180000} \cdot x^9_{10000} \oplus \text{0x18000} \cdot x^9_{11100} \\
&\oplus \text{0x18000} \cdot x^9_{11110} \\
= {} &\text{0x300} \cdot x^{10}_{10111} \oplus \text{0x180000} \cdot x^{10}_{00010} \oplus \text{0x180000} \cdot x^{10}_{10011} \oplus \text{0x300} \cdot x^{10}_{10101} \\
&\oplus \text{0x180000} \cdot x^{10}_{00000} \oplus \text{0x180000} \cdot x^{10}_{10001} \oplus \text{0x18000} \cdot x^{10}_{11101} \\
&\oplus \text{0x18000} \cdot x^{10}_{11111}
\end{aligned}$$

which is just the right-hand side of (3). From our calculations, it is clear that the bias for the linear approximation (3) at Round 7 equals the combined bias of the six approximations (4), (5), (8), (9), (10), (11) holding simultaneously. We observe that the terms occurring in any linear approximation never occur in the other approximations. If we *assume* these approximations to be independent from that observation, we can deduce the total bias as $(2^{-1})^6 = 2^{-6}$ by Piling-up lemma [11], since we know (4), (5), (8), (9), (10), (11) has bias 2^{-1} each as mentioned before. Unfortunately, as we demonstrate later, this independence assumption is not true. The following results will be useful for us before we move on.

Theorem 1. *Suppose that the 32-bit random variables A, B, A', B' are independent. Then, for any 32-bit m, m', the two bits $m \cdot (A + B) \oplus m \cdot (A \oplus B)$ and $m' \cdot (A' + B') \oplus m' \cdot (A' \oplus B')$ are independent.*

Proof. First, we let $U = (A+B) \oplus A \oplus B$ and $U' = (A'+B') \oplus A' \oplus B'$. As A, B, A', B' are independent, we know U, U' are independent. Given 32-bit m and 1-bit v, we define the set $E_{<m,v>} = \{X \in GF(2)^{32} : m \cdot X = v\}$. We let $|E_{<m,v>}|$ denote the cardinality of the set $E_{<m,v>}$, and let $E^i_{<m,v>}$ denote the element of $E_{<m,v>}$ for $i = 1, 2, \ldots, |E_{<m,v>}|$. Hence, for any $v, v' \in \{0, 1\}$, we have

$$\begin{aligned}&\Pr(m \cdot U = v, m' \cdot U' = v')\\ =&\sum_{i,j} \Pr(U = E^i_{<m,v>}, U' = E^j_{<m',v'>})\end{aligned}$$

Since U, U' are independent, we have

$$\begin{aligned}&\Pr(m \cdot U = v, m' \cdot U' = v')\\ =&\sum_{i} \Pr(U = E^i_{<m,v>}) \sum_{j} \Pr(U' = E^j_{<m',v'>})\\ =&\Pr(m \cdot U = v) \Pr(m' \cdot U' = v')\end{aligned}$$

Thus, we see that the two bits $m \cdot U$ and $m' \cdot U'$ are independent. □

We further note that assuming the 32-bit random variables A, B are independent with uniform probability distribution, the bias for the linear approximation $m \cdot (A \oplus B) \approx m \cdot (A + B)$, i.e., the bias for the bit $m \cdot (A \oplus B) \oplus m \cdot (A + B)$, for any[7] 32-bit m can be efficiently evaluated by [13, Theorem 1, Page 156]. With regards to CubeHash round function, we have the following results.

[7] For special patterns of m such as those we have mentioned before, we have quick results on the bias (see [6, Corollary 1, Page 232] for details).

Theorem 2. *Assume that the 1024-bit initial state of CubeHash compression function is random and uniformly distributed before Round 1 begins, then, the 1024-bit state of CubeHash is random and uniformly distributed at any step of any round.*

Proof. Because the ten step operations are all invertible, the state transition function is a permutation over $GF(2)^{1024}$ from Step i to Step $i+1$ within the round. By induction, we know that the 1024-bit state of CubeHash compression function is random and uniformly distributed at any step of any round, assuming that the 1024-bit initial state of CubeHash is random and uniformly distributed before Round 1 begins. □

We can immediately justify that Approximations (4), (5) are independent as follows. By Theorem 2, the involved states x_{10100}, x_{00100} in (4) and the involved states x_{10110}, x_{00110} in (5) are all independent. And we apply Theorem 1 with

$$m = m' = 0\text{x}300, A = x_{10100}, B = x_{00100}, A' = x_{10110}, B' = x_{00110},$$

to see that Approximations (4), (5) are independent. Similarly, Approximations (8), (9), (10), (11) are independent pairwise.

Our main focus here is to show below that these two groups of approximations are, however, *not* independent. Note that the internal states are invertible with the CubeHash round function T, as each step operation is invertible. Thus, Step 1 operation allows to have the equalities hold true always, $x_{0m} = x^1_{0m}$ and $x_{1m} = x^1_{1m} - x^1_{0m}$ for all 4-bit m within the round. So, we can substitute all those state variables of x_n's in Approximations (4) and (5) and get the equivalent expression in terms of x^1_n's (i.e., states right after Step 1) only. We obtain

$$0\text{x}300 \cdot x^1_{10100} \oplus 0\text{x}300 \cdot x^1_{00100} \approx 0\text{x}300 \cdot (x^1_{10100} - x^1_{00100}) \qquad (12)$$
$$0\text{x}300 \cdot x^1_{10110} \oplus 0\text{x}300 \cdot x^1_{00110} \approx 0\text{x}300 \cdot (x^1_{10110} - x^1_{00110}) \qquad (13)$$

Here, note that rewriting the Approximations as shown above does not change the bias[8], because the state variables are always replaced by their equivalents variables. Similarly, we rewrite Approximations (8), (9), (10),

[8] That is, (4) and (12) have exactly the same bias, and (5) and (13) have exactly the same bias.

(11) in terms of states right after Step 1 as follows respectively,

$$0\text{x}300 \cdot (x^1_{10100} + (x^1_{01110} \lll 7 \oplus x^1_{10110})) \approx 0\text{x}300 \cdot (x^1_{10100} \oplus x^1_{01110} \lll 7 \oplus x^1_{10110}) \quad (14)$$

$$0\text{x}300 \cdot (x^1_{10110} + (x^1_{01100} \lll 7 \oplus x^1_{10100})) \approx 0\text{x}300 \cdot (x^1_{10110} \oplus x^1_{01100} \lll 7 \oplus x^1_{10100}) \quad (15)$$

$$0\text{x}18000 \cdot (x^1_{11110} + (x^1_{00100} \lll 7 \oplus x^1_{11100})) \approx 0\text{x}18000 \cdot (x^1_{11110} \oplus x^1_{00100} \lll 7 \oplus x^1_{11100}) \quad (16)$$

$$0\text{x}18000 \cdot (x^1_{11100} + (x^1_{00110} \lll 7 \oplus x^1_{11110})) \approx 0\text{x}18000 \cdot (x^1_{11100} \oplus x^1_{00110} \lll 7 \oplus x^1_{11110}) \quad (17)$$

Next, we will apply the synthetic bias analysis for CubeHash Round 7 to analyze (12), (13), (14), (15), (16) and (17). We will group the six approximations.

Both (14) and (15) arise at Step 6, and they are independent. As x^1_{10100} occurs in both (12) and (14), we group (12) and (14) together. Likewise, as x^1_{10110} occurs in both (13) and (15), we group (13) and (15) together. Both (16) and (17) arise at Step 6, and they are independent. As (16), (17) relates to x^1_{00100}, x^1_{00110} respectively, x^1_{00100} is related to (12), (14), and x^1_{00110} is related to (13) and (15). Therefore, we are able to make the two groups. Group One contains (12), (14), (16). Group Two contains (13), (15), (17).

To show these two groups are independent, we need[9] to show the four pairs [(12), (15)], [(12), (17)], [(13), (14)] and [(13), (16)] are independent respectively. By Theorem 1, we can immediately verify independence for the two pairs [(12), (17)], [(13), (16)] as follows. We verify independence for [(12), (17)] by applying Theorem 1, with $m = 0\text{x}300$, $A = x^1_{00100}$, $B = x^1_{10100} - x^1_{00100}$, $m' = 0\text{x}18000$, $A' = x^1_{11100}$, $B' = x^1_{00110} \lll 7 \oplus x^1_{11110}$; and the assumption that A, B, A', B' are independent is justified by applying Theorem 2. And we verify independence for [(13), (16)] with $m = 0\text{x}300$, $A = x^1_{00110}$, $B = x^1_{10110} - x^1_{00110}$, $m' = 0\text{x}18000$, $A' = x^1_{11110}$, $B' = x^1_{00100} \lll 7 \oplus x^1_{11100}$. We can similarly check the two pairs [(12), (15)], [(13), (14)] are independent respectively. Hence, we have justified that the two groups are independent.

With regards to the joint bias for each group, our computations show that the joint bias for the group of approximations (12), (14), (16) holding simultaneously is around $2^{-2.5}$ and the joint bias for (13), (15), (17) is

[9] Note that these five pairs [(12), (13)], [(14), (15)], [(14), (17)], [(16, (15)] and [(16), (17)] are independent respectively, because each pair arises from the same step.

around $2^{-2.5}$. Consequently, the total bias for the linear approximation (3) at Round 7, is calculated as $2^{-2.5} \times 2^{-2.5} = 2^{-5}$. In contrast, recall that if the dependency within the individual groups of functions of the round is ignored, that is, all the linear approximations of the round were assumed to be independent, we would have a smaller bias 2^{-6} at Round 7, as we have explained earlier.

3.3 Improved Attack on 11-round CubeHash

Using our synthetic analysis technique in Section 3.2, we analyzed all the 11 rounds for CubeHash. We give our results[10] in Table 1. Note that in Table 1, we see no bias improvement for Round 5. This is because we can actually show all the linear approximations for Round 5 are independent. Similar arguments hold true for Round 6. We now briefly explain. For Round 5, we can apply the same technique to demonstrate that the four linear approximations are involved:

$$0\text{x}c00000 \cdot (x^1_{01001} \oplus x^1_{11001}) \approx 0\text{x}c00000 \ \cdot \ (x^1_{11001} - x^1_{01001}) \tag{18}$$

$$0\text{x}c00000 \cdot (x^1_{01011} \oplus x^1_{11011}) \approx 0\text{x}c00000 \ \cdot \ (x^1_{11011} - x^1_{01011}) \tag{19}$$

$$0\text{x}6 \cdot (x^1_{11111} + (x^1_{00101} \lll 7 \oplus x^1_{11101})) \approx 0\text{x}6 \cdot (x^1_{11111} \oplus x^1_{00101} \lll 7 \oplus x^1_{11101}) \tag{20}$$

$$0\text{x}6 \cdot (x^1_{11101} + (x^1_{00111} \lll 7 \oplus x^1_{11111})) \approx 0\text{x}6 \cdot (x^1_{11101} \oplus x^1_{00111} \lll 7 \oplus x^1_{11111}) \tag{21}$$

And these four approximations are pairwise independent. For Round 6, two independent linear approximations arise at Step 6:

$$0\text{x}300 \cdot (x^5_{00101} \oplus x^5_{10101}) \approx 0\text{x}300 \cdot (x^5_{00101} + x^5_{10101}) \tag{22}$$

$$0\text{x}300 \cdot (x^5_{00111} \oplus x^5_{10111}) \approx 0\text{x}300 \cdot (x^5_{00111} + x^5_{10111}) \tag{23}$$

Due to the dependence within each round (except Round 5 and Round 6), we are able to improve the bias estimate for 11-round CubeHash from 2^{-234} in [2] to $2^{-207.1}$. Recall a well-known fact in coding theory, that is, if the bias of the bit A is ϵ, then we can successfully distinguish the distribution of randomly and uniformly chosen $(\frac{1}{\epsilon})^2$ samples of A from uniform distribution with probability of success higher than $\frac{1}{2}$. This gives an improved attack for 11-round CubeHash with complexity $2^{414.2}$.

[10] Note that our definition of bias in (1) is twice in quantity of that in [2], which was defined as $\Pr(X = 0) - 1/2$. In Table 1, the biases of [2] were adjusted using our definition, and thus were twice of those in [2, Table 6, Page 472]; and the total bias for 11-round CubeHash were calculated as 2^{-234} in Table 1 rather than 2^{-235} in [2].

Table 1. Our analysis results on 11-round linear approximations of CubeHash

round	1	2	3	4	5	6
our bias	2^{-29}	$2^{-35.7}$	$2^{-16.9}$	2^{-13}	2^{-4}	2^{-2}
paper [2]	2^{-34}	2^{-40}	2^{-18}	2^{-14}	2^{-4}	2^{-2}

round	7	8	9	10	11	total
our bias	2^{-5}	$2^{-13.8}$	$2^{-18.7}$	$2^{-36.5}$	$2^{-32.5}$	$2^{-207.1}$
paper [2]	2^{-6}	2^{-16}	2^{-22}	2^{-42}	2^{-36}	2^{-234}

3.4 Improved Attack on 12-round CubeHash

We can extend our above results to attack 12-round CubeHash. Our analysis shows that by choosing the same output masks from the set

$$\{0\text{x}600, 0\text{x}18000, 0\text{x}180000, 0\text{x}c000000, 0\text{x}c0000000\}$$

for x_{01101} and x_{01111} at the end of Round 5, going backwards 6 rounds, forwards 6 rounds, we get[11] five new linear approximations (given in Appendix B) on 12-round CubeHash. They all have the same bias of around $2^{-261.1}$. In particular, with this construction, the last 11 rounds all have the same bias as our 11-round CubeHash above; moreover, the bias for its first round, if we assume all linear approximations are independent, is 2^{-54}. With our synthetic analysis in Section 3.2, we can improve the bias for its first round to 2^{-51}, which leads to the total improved bias $2^{-258.1}$ for 12-round CubeHash.

Additionally, from each of the five linear approximations on 12-round CubeHash, we can deduce 8 linear approximations of equal bias as follows. Observe that the above construction is derived by letting the pair (x_{01101}, x_{01111}) at the end of Round 5 going backwards and forwards. We further found that we can similarly derive five linear approximations from each of the other seven pairs:

$$(x_{00000}, x_{00010}), (x_{00001}, x_{00011}),\ (x_{00100}, x_{00110}), (x_{00101}, x_{00111}),$$
$$(x_{01000}, x_{01010}), (x_{01001}, x_{01011}),\ (x_{01100}, x_{01110}).$$

[11] Note that in above analysis on 11-round CubeHash, the 11-round linear approximation can be obtained by going backwards 5 rounds and forwards 6 rounds with mask 0x6 for x_{01101} and x_{01111} at the end of Round 5.

Therefore, for 12-round CubeHash, we have $8 \times 5 = 40$ linear approximations of equal bias.

Advanced Two-round Analysis In above analysis, the analysis is focused within each round. According to the specification of CubeHash, no randomization is introduced between consecutive rounds, and the biases of consecutive rounds of CubeHash are likely to be dependent. Here, we illustrate the two-round analysis on Round 6 and Round 7. Note that a similar approach works for all consecutive two rounds.

To apply our synthetic analysis on Round 6 and Round 7 jointly, we rewrite (22), (23) in terms of states after Step 1 at Round 7:

$$\begin{aligned}
0\text{x}300 \cdot (x^1_{10100} - x^1_{00100}) \approx\ & 0\text{x}300 \cdot ((x^1_{10100} - x^1_{00100}) - \\
& (x^1_{00001} \oplus (x^1_{10000} - x^1_{00000})) \ggg 11 \oplus \\
& (x^1_{00001} \oplus (x^1_{10000} - x^1_{00000})) \ggg 11) \qquad (24) \\
0\text{x}300 \cdot (x^1_{10110} - x^1_{00110}) \approx\ & 0\text{x}300 \cdot ((x^1_{10110} - x^1_{00110}) - \\
& (x^1_{00011} \oplus (x^1_{10010} - x^1_{00010})) \ggg 11 \oplus \\
& (x^1_{00011} \oplus (x^1_{10010} - x^1_{00010})) \ggg 11) \qquad (25)
\end{aligned}$$

Now, we apply our synthetic analysis on the eight linear approximations (24), (25), (12), (13), (14), (15), (16) and (17), which are involved at Round 6 and Round 7 and all expressed by the states at the same step. It is easy to see that (25) is independent of (24), (12), (14), (16); likewise, (24) is independent of (25), (13), (15), (17). Thus, based on our previous results within one round, we have two independent groups for Round 6 and Round 7 together. Group one contains (24), (12), (14), (16). Group two contains (25), (13), (15), (17). Our computations show that both Group one and Group two has bias 2^{-3}. So, Round 6 and Round 7 yield the combined improved bias 2^{-6}.

Finally, we calculate the overall bias for 12-round CubeHash. Recall that before the two-round advanced analysis, the bias for each of the 40 linear approximations for 12-round CubeHash was $2^{-258.1}$; in particular, the combined bias of Round 6 and Round 7 was $2^{-2} \times 2^{-5} = 2^{-7}$, if we assume Round 6 and Round 7 to be independent. With our above advanced two-round analysis, the combined bias of Round 6 and Round 7 improves to 2^{-6}. Thus, we have the improved bias estimate $2^{-257.1}$ for 12-round CubeHash. By using the aforementioned 40 equal biases, we have an improved attack with complexity

$$\frac{1}{40} \times (\frac{1}{2^{-257.1}})^2 = 2^{509}.$$

4 Improved Analysis on Stream Cipher Rabbit

4.1 Preliminaries

Rabbit [5] is a stream cipher among the finalists of EU-funded ECRYPT Stream Cipher Project (eSTREAM). Rabbit encryption algorithm has been published as informational RFC 4503 with the Internet Engineering Task Force (IETF), the standardization body for Internet technology. We give a short description on Rabbit here. We refer to [5, 9] for full description. Rabbit outputs the 128-bit keystream block s_i from the eight state variables x's of 32 bits at each iteration i,

$$\begin{array}{llll}
s_i^{[15..0]} & = x_{0,i}^{[15..0]} \oplus x_{5,i}^{[31..16]} & s_i^{[31..16]} & = x_{0,i}^{[31..16]} \oplus x_{3,i}^{[15..0]} \\
s_i^{[47..32]} & = x_{2,i}^{[15..0]} \oplus x_{7,i}^{[31..16]} & s_i^{[63..48]} & = x_{2,i}^{[31..16]} \oplus x_{5,i}^{[15..0]} \\
s_i^{[79..64]} & = x_{4,i}^{[15..0]} \oplus x_{1,i}^{[31..16]} & s_i^{[95..80]} & = x_{4,i}^{[31..16]} \oplus x_{7,i}^{[15..0]} \\
s_i^{[111..96]} & = x_{6,i}^{[15..0]} \oplus x_{3,i}^{[31..16]} & s_i^{[127..112]} & = x_{6,i}^{[31..16]} \oplus x_{1,i}^{[15..0]}
\end{array}$$

The state variables x's are computed from intermediate variables g's of 32 bits,

$$x_{0,i+1} = g_{0,i} + (g_{7,i} \lll 16) + (g_{6,i} \lll 16) \tag{26}$$

$$x_{1,i+1} = g_{1,i} + (g_{0,i} \lll 8) + g_{7,i} \tag{27}$$

$$x_{2,i+1} = g_{2,i} + (g_{1,i} \lll 16) + (g_{0,i} \lll 16) \tag{28}$$

$$x_{3,i+1} = g_{3,i} + (g_{2,i} \lll 8) + g_{1,i} \tag{29}$$

$$x_{4,i+1} = g_{4,i} + (g_{3,i} \lll 16) + (g_{2,i} \lll 16) \tag{30}$$

$$x_{5,i+1} = g_{5,i} + (g_{4,i} \lll 8) + g_{3,i} \tag{31}$$

$$x_{6,i+1} = g_{6,i} + (g_{5,i} \lll 16) + (g_{4,i} \lll 16) \tag{32}$$

$$x_{7,i+1} = g_{7,i} + (g_{6,i} \lll 8) + g_{5,i} \tag{33}$$

where $\lll$ denotes left bit-wise rotation and all additions are computed modulo 2^{32}. The description of computing g's (see [5, 9]) is not relevant for us and we omit it here.

Prior to [9], the work of [3, 8] focused on the bias analysis within one keystream sub-block of 16 bits, and the best distinguishing attack [8] has complexity 2^{158}. Recently, [9] studied the bias of Rabbit involving multiple sub-blocks of one keystream block. More formally, [9] proposed to estimate the bias for Rabbit keystream outputs of the form

$$Mask_0 \cdot s_{i+1}^{[15..0]} \oplus Mask_1 \cdot s_{i+1}^{[31..16]} \oplus \cdots \oplus Mask_7 \cdot s_{i+1}^{[127..112]}, \tag{34}$$

(with 16-bit $Mask_i$'s) as two parts separately, i.e., the linear part $\alpha_0 \cdot g_{0,i} \oplus \alpha_1 \cdot g_{1,i} \oplus \cdots \oplus \alpha_7 \cdot g_{7,i}$ (with appropriate 32-bit α_i's) and the nonlinear part $Mask_0 \cdot s_{i+1}^{[15..0]} \oplus \cdots \oplus Mask_7 \cdot s_{i+1}^{[127..112]} \oplus \alpha_0 \cdot g_{0,i} \oplus \cdots \oplus \alpha_7 \cdot g_{7,i}$. In particular, [9] showed that for

$$0\text{x}606 \cdot s_{i+1}^{[47..32]} \oplus 0\text{x}606 \cdot s_{i+1}^{[79..64]} \oplus 0\text{x}606 \cdot s_{i+1}^{[111..96]}, \tag{35}$$

the bias[12] for the linear part and the nonlinear part is $2^{-50.5}$, 2^{-20} respectively. It thus makes the total bias of around $2^{-70.5}$ and yields the best distinguishing attack with complexity 2^{141}, which is still above the claimed security level 2^{128} ([5, Page 18]).

4.2 Our Synthetic Analysis on Rabbit

In this section, we apply our new conditional dependent problem in Section 2.2 to analyze the nonlinear part[13] of the bias analysis, i.e., the total combined bias of the six linear approximations below for $m = 0\text{x}606, m' = 0\text{x}6060000$ (for simplicity we omit the irrelevant subscripts i from the variables g):

$$m \cdot (g_2 + g_1 \lll 16 + g_0 \lll 16) \approx m \cdot (g_2 \oplus g_1 \lll 16 \oplus g_0 \lll 16) \tag{36}$$
$$m \cdot (g_4 + g_3 \lll 16 + g_2 \lll 16) \approx m \cdot (g_4 \oplus g_3 \lll 16 \oplus g_2 \lll 16) \tag{37}$$
$$m \cdot (g_6 + g_5 \lll 16 + g_4 \lll 16) \approx m \cdot (g_6 \oplus g_5 \lll 16 \oplus g_4 \lll 16) \tag{38}$$
$$m' \cdot (g_1 + g_0 \lll 8 + g_7) \approx m' \cdot (g_1 \oplus g_0 \lll 8 \oplus g_7) \tag{39}$$
$$m' \cdot (g_3 + g_2 \lll 8 + g_1) \approx m' \cdot (g_3 \oplus g_2 \lll 8 \oplus g_1) \tag{40}$$
$$m' \cdot (g_7 + g_6 \lll 8 + g_5) \approx m' \cdot (g_7 \oplus g_6 \lll 8 \oplus g_5) \tag{41}$$

Let Group One contain (36), (37), (40) and Group Two contain (38), (41). We can demonstrate that the linear approximations in Group One are independent from those in Group Two. Nonetheless, given (39), the two groups are *not* independent. We let A denote the corresponding Boolean function of (39) by replacing '$\approx$' with '$\oplus$' in (39), and let B, C denote the corresponding Boolean function for Group One, Group Two respectively by replacing '$\approx$' with '$\oplus$' in all the linear approximations in the group and XORing them together. Obviously, this is a conditional dependent bias problem as we proposed in Section 2.2.

Using our first solution in Section 2.2, we compute the bias for $A \oplus B, C$ respectively and get the results $2^{-11.4}$, 2^{-6}. We estimate the combined bias

[12] with our definition (1) on bias

[13] Note that the bias of the linear part cannot be improved.

for above six linear approximations by

$$2^{-11.4} \times 2^{-6} = 2^{-17.4}. \tag{42}$$

Now, we want to apply our black-box solution (2) in Section 2.2. In our case, we have $u_1 = g_7^{[31..16]}$. For $A \oplus B$, we compute with 2^{26} random samples for each randomly chosen u_1 and we run it 2^{14} times. We obtain the results in hexadecimal form,

$$q = \frac{\text{0x}24a6}{\text{0x}4000}, \quad p_+ = \frac{\text{0x}1e0e6fc1}{\text{0x}24a6 \times 2^{25}}, \quad p_- = \frac{\text{0x}123210ab}{\text{0x}1b5a \times 2^{25}}.$$

They correspond to the percentage of positive bias 57.3%, the average of positive bias $+2^{-9.3}$, the average of negative bias $-2^{-9.6}$, and the average bias $+2^{-11.43}$ of all. For the function C, we compute with 2^{22} random samples for each randomly chosen u_1 and we run it 2^{16} times. We obtain the results in hexadecimal form,

$$q' = \frac{\text{0x}afed}{\text{0x}10000}, \quad p'_+ = \frac{\text{0x}d4698c87}{\text{0x}afed \times 2^{21}}, \quad p'_- = \frac{\text{0x}4ea00ceb}{\text{0x}5013 \times 2^{21}}.$$

They correspond to the percentage of positive bias 68.7%, the average of positive bias $+2^{-4.73}$, the average of negative bias $-2^{-5.03}$, and the average bias $+2^{-5.94}$ of all[14] .

By (2), we estimate the bias $2^{-17.5}$ for $A \oplus B \oplus C$. This result agrees with our first estimation (42). Note that based on the naive independence assumption, this combined bias was estimated in [9] to be smaller, i.e., 2^{-20}. Consequently, based on [9], we have an improved attack on Rabbit with complexity

$$(2^{-50.5} \times 2^{-17.5})^{-2} = 2^{136}.$$

5 Conclusion

In this paper, we take a first step towards the synthetic approach on bias analysis. We apply the "Divide-and-Conquer" method to our synthetic bias analysis. When multiple (possibly dependent) Boolean functions are involved, we propose to group dependent ones together and make independent groups. This allows to split the original complicated bias problem into smaller ones separately. Thus, for each split problem, the input space is reduced and the combined bias is enlarged. The former might make exhaustive computation possible now, while the latter might make

[14] The computations were run $3 \sim 5$ times and we always got these same statistics.

computing empirically possible. Obviously, with our synthetic approach, the task of bias analysis becomes easier. In the meantime, we also propose a conditional dependent bias problem. Based on naive heuristics and certain ideal assumptions, we give the bias analysis to estimate the bias.

Our synthetic approach is successfully applied to improve the best linear attacks [2, 9] on CubeHash and Rabbit respectively. We present an improved attack on 11-round CubeHash with complexity $2^{414.2}$. Based on our 11-round attack, we give a new linear attack for 12-round CubeHash with complexity 2^{509}. It is the first known attack on 12 rounds with complexity below the security parameter 2^{512} of CubeHash. We also give an improved attack on Rabbit with complexity 2^{136}.

As for the future work, we note that our current analysis on grouping is done manually. When too many variables are involved in the combined bias problem for multiple Boolean functions, a tool to make groups automatically is essential for the synthetic analysis. We believe that with the tool, we can perform three-round (or even more rounds) analysis and improve our bias analysis for 12-round CubeHash.

Acknowledgments

We gratefully thank the anonymous reviewers for many helpful and valuable comments. The first author is supported by the National Science and Technology Major Project No. 2010ZX01036-001-002 and the Knowledge Innovation Key Directional Program of Chinese Academy of Sciences under Grant No. KGCX2-YW-125, and the National Natural Science Foundation of China under Grant No. 90818012.

References

1. J. Alizadeh, A. Mirghadri, *A new distinguisher for CubeHash-8/b and CubeHash-15/b compression functions*, IACR eprint, `http://eprint.iacr.org/2011/550`, 2011.
2. T. Ashur, O. Dunkelman, *Linear analysis of reduced-round CubeHash*, ACNS 2011, LNCS vol. 6715, pp. 462-478, Springer-Verlag, 2011.
3. J. P. Aumasson, *On a bias of Rabbit*, SASC 2007, `http://www.ecrypt.eu.org/stream/papersdir/2007/033.pdf`, 2007.
4. D. J. Bernstein, *CubeHash specification (2.B.1)*, submission to National Institute of Standards and Technology (NIST), 2009.
5. M. Boesgaard, M. Vesterager, T. Christensen and E. Zenner, *The stream cipher Rabbit (version 1.1)*, the ECRYPT stream cipher project, `http://www.ecrypt.eu.org/stream/`.
6. J. Y. Cho, J. Pieprzyk, *Multiple modular additions and crossword puzzle attack on NLSv2*, ISC 2007, LNCS vol. 4779, pp. 230-248, 2007.

7. ECRYPT II SHA-3 Zoo, *CubeHash Profile*, `http://ehash.iaik.tugraz.at/wiki/The_SHA-3_Zoo`.

8. Y. Lu, H. Wang, S. Ling, *Cryptanalysis of Rabbit*, ISC 2008, LNCS vol. 5222, pp. 204-214, Springer-Verlag, 2008.

9. Y. Lu, Y. Desmedt, *Improved distinguishing attack on Rabbit*, ISC 2010, LNCS vol. 6531, pp. 17-23, Springer-Verlag, 2011.

10. Y. Lu, S. Vaudenay, W. Meier, L. Ding, J. Jiang, *Synthetic linear analysis: Improved attacks on CubeHash and Rabbit*, to appear in the proceedings of the 14th Annual International Conference on Information Security and Cryptology (ICISC 2011), LNCS vol. 7259, Springer-Verlag, 2012.

11. M. Matsui, *Linear cryptanalysis method for DES cipher*, EUROCRYPT 1993, LNCS vol. 765, pp. 386-397, 1994.

12. National Institute of Standards and Technology (NIST), *Cryptographic hash algorithm competition*, `http://www.nist.gov/hash-competition`.

13. K. Nyberg, J. Wallen, *Improved linear distinguishers for SNOW 2.0*, FSE 2006, LNCS vol. 4047, pp. 144-162, 2006.

14. P. Sepehrdad, S. Vaudenay, M. Vuagnoux, *Discovery and exploitation of new biases in RC4*, SAC 2010, LNCS vol. 6544, pp. 74-91, 2011.

Appendix A: Analysis on CubeHash Round 8

For CubeHash Round 8, six pairwise independent approximations arise at Step 1. They are presented in terms of states right after Step 1.

$$
\begin{aligned}
0\text{x}180000 \cdot (x^1_{00001} \oplus x^1_{10001}) &\approx 0\text{x}180000 \cdot (x^1_{00001} - x^1_{10001}) \qquad (43)\\
0\text{x}180000 \cdot (x^1_{00011} \oplus x^1_{10011}) &\approx 0\text{x}180000 \cdot (x^1_{00011} - x^1_{10011}) \qquad (44)\\
0\text{x}300 \cdot (x^1_{00101} \oplus x^1_{10101}) &\approx 0\text{x}300 \cdot (x^1_{00101} - x^1_{10101}) \qquad (45)\\
0\text{x}300 \cdot (x^1_{00111} \oplus x^1_{10111}) &\approx 0\text{x}300 \cdot (x^1_{00111} - x^1_{10111}) \qquad (46)\\
0\text{x}18000 \cdot (x^1_{01101} \oplus x^1_{11101}) &\approx 0\text{x}18000 \cdot (x^1_{01101} - x^1_{11101}) \qquad (47)\\
0\text{x}18000 \cdot (x^1_{01111} \oplus x^1_{11111}) &\approx 0\text{x}18000 \cdot (x^1_{01111} - x^1_{11111}) \qquad (48)
\end{aligned}
$$

Note that each of above six approximation has bias 2^{-1} by [6]. Eight pairwise independent approximations arise at Step 6. They are presented in terms of states right after Step 1.

$$\begin{aligned}
\text{0x180000} \cdot (x^1_{10011} + (x^1_{01001} \lll 7 \oplus x^1_{10001})) &\approx \\
&\text{0x180000} \cdot (x^1_{10011} \oplus x^1_{01001} \lll 7 \oplus x^1_{10001}) \quad (49)\\
\text{0x180000} \cdot (x^1_{10001} + (x^1_{01011} \lll 7 \oplus x^1_{10011})) &\approx \\
&\text{0x180000} \cdot (x^1_{10001} \oplus x^1_{01011} \lll 7 \oplus x^1_{10011}) \quad (50)\\
\text{0x}c\text{00300} \cdot (x^1_{10111} + (x^1_{01101} \lll 7 \oplus x^1_{10101})) &\approx \\
&\text{0x}c\text{00300} \cdot (x^1_{10111} \oplus x^1_{01101} \lll 7 \oplus x^1_{10101}) \quad (51)\\
\text{0x}c\text{00300} \cdot (x^1_{10101} + (x^1_{01111} \lll 7 \oplus x^1_{10111})) &\approx \\
&\text{0x}c\text{00300} \cdot (x^1_{10101} \oplus x^1_{01111} \lll 7 \oplus x^1_{10111}) \quad (52)\\
\text{0x}c\text{000000} \cdot (x^1_{11010} + (x^1_{00000} \lll 7 \oplus x^1_{11000})) &\approx \\
&\text{0x}c\text{000000} \cdot (x^1_{11010} \oplus x^1_{00000} \lll 7 \oplus x^1_{11000}) \quad (53)\\
\text{0x}c\text{000000} \cdot (x^1_{11011} + (x^1_{00001} \lll 7 \oplus x^1_{11001})) &\approx \\
&\text{0x}c\text{000000} \cdot (x^1_{11011} \oplus x^1_{00001} \lll 7 \oplus x^1_{11001}) \quad (54)\\
\text{0x}c\text{000000} \cdot (x^1_{11000} + (x^1_{00010} \lll 7 \oplus x^1_{11010})) &\approx \\
&\text{0x}c\text{000000} \cdot (x^1_{11000} \oplus x^1_{00010} \lll 7 \oplus x^1_{11010}) \quad (55)\\
\text{0x}c\text{000000} \cdot (x^1_{11001} + (x^1_{00011} \lll 7 \oplus x^1_{11011})) &\approx \\
&\text{0x}c\text{000000} \cdot (x^1_{11001} \oplus x^1_{00011} \lll 7 \oplus x^1_{11011}) \quad (56)
\end{aligned}$$

Thus, we have 6+8=14 linear approximations involved in this round. Note that (49), (50), (53), (54), (55), (56) has bias 2^{-1} each; both (51) and (52) has bias 2^{-2} by [13]. As was done for Round 7 in Section 3.2, we can demonstrate that these 14 approximations fall into four independent groups. Group One contains (43), (44), (49), (50), (54), (56). Group Two contains (45), (46), (47), (48), (51), (52). Group Three contains (53). Group Four contains (55).

As Group Three and Group Four each contains only one approximation, we know the bias is 2^{-1} for each group. Group One and Group Two each contains six approximations. We compute the total bias for each group separately. Our results show that the bias for Group One is 2^{-5} and the bias for Group Two is $2^{-6.8}$. Note that the independence assumption would yield a smaller bias $2^{-6}, 2^{-8}$ for Group One, Group Two respectively. Consequently, we deduce the total bias $2^{-5} \times 2^{-6.8} \times 2^{-1} \times 2^{-1} = 2^{-13.8}$ for Round 8, by considering the dependence within the round. Note that, if the dependency within the round is ignored, we would have a smaller bias $(2^{-1})^{12} \times (2^{-2})^2 = 2^{-16}$.

Appendix B: New Linear Approximations on 12-round CubeHash

The five new linear approximations on 12-round CubeHash, which we used in Section 3.4, are given below (x, x' denote the inputs, outputs respectively):

$$
\begin{aligned}
&0\mathrm{x}18199800 \cdot x_{00000} \oplus 0\mathrm{x}18199800 \cdot x_{00010} \oplus 0\mathrm{x}e7999f81 \cdot x_{01101} \\
&\oplus 0\mathrm{x}e7999f81 \cdot x_{01111} \oplus 0\mathrm{x}18199800 \cdot x_{10001} \oplus 0\mathrm{x}18199800 \cdot x_{10011} \\
&\oplus 0\mathrm{x}30333 \cdot x_{10101} \oplus 0\mathrm{x}30333 \cdot x_{10111} \oplus 0\mathrm{x}1819980 \cdot x_{11101} \\
&\oplus 0\mathrm{x}1819980 \cdot x_{11111} \approx 0\mathrm{x}99800181 \cdot x'_{00000} \oplus 0\mathrm{x}99800181 \cdot x'_{00010} \\
&\oplus 0\mathrm{x}18006018 \cdot x'_{01101} \oplus 0\mathrm{x}18006018 \cdot x'_{01111} \oplus 0\mathrm{x}99800181 \cdot x'_{10001} \\
&\oplus 0\mathrm{x}99800181 \cdot x'_{10011} \oplus 0\mathrm{x}30333000 \cdot x'_{10101} \oplus 0\mathrm{x}30333000 \cdot x'_{10111} \\
&\oplus 0\mathrm{x}19980018 \cdot x'_{11101} \oplus 0\mathrm{x}19980018 \cdot x'_{11111}
\end{aligned}
$$

$$
\begin{aligned}
&0\mathrm{x}6660006 \cdot x_{00000} \oplus 0\mathrm{x}6660006 \cdot x_{00010} \oplus 0\mathrm{x}e667e079 \cdot x_{01101} \\
&\oplus 0\mathrm{x}e667e079 \cdot x_{01111} \oplus 0\mathrm{x}6660006 \cdot x_{10001} \oplus 0\mathrm{x}6660006 \cdot x_{10011} \\
&\oplus 0\mathrm{x}c0ccc0 \cdot x_{10101} \oplus 0\mathrm{x}c0ccc0 \cdot x_{10111} \oplus 0\mathrm{x}60666000 \cdot x_{11101} \\
&\oplus 0\mathrm{x}60666000 \cdot x_{11111} \approx 0\mathrm{x}60006066 \cdot x'_{00000} \oplus 0\mathrm{x}60006066 \cdot x'_{00010} \\
&\oplus 0\mathrm{x}180606 \cdot x'_{01101} \oplus 0\mathrm{x}180606 \cdot x'_{01111} \oplus 0\mathrm{x}60006066 \cdot x'_{10001} \\
&\oplus 0\mathrm{x}60006066 \cdot x'_{10011} \oplus 0\mathrm{x}ccc000c \cdot x'_{10101} \oplus 0\mathrm{x}ccc000c \cdot x'_{10111} \\
&\oplus 0\mathrm{x}66000606 \cdot x'_{11101} \oplus 0\mathrm{x}66000606 \cdot x'_{11111}
\end{aligned}
$$

$$
\begin{aligned}
&0\mathrm{x}66600060 \cdot x_{00000} \oplus 0\mathrm{x}66600060 \cdot x_{00010} \oplus 0\mathrm{x}667e079e \cdot x_{01101} \\
&\oplus 0\mathrm{x}667e079e \cdot x_{01111} \oplus 0\mathrm{x}66600060 \cdot x_{10001} \oplus 0\mathrm{x}66600060 \cdot x_{10011} \\
&\oplus 0\mathrm{x}c0ccc00 \cdot x_{10101} \oplus 0\mathrm{x}c0ccc00 \cdot x_{10111} \oplus 0\mathrm{x}6660006 \cdot x_{11101} \\
&\oplus 0\mathrm{x}6660006 \cdot x_{11111} \approx 0\mathrm{x}60666 \cdot x'_{00000} \oplus 0\mathrm{x}60666 \cdot x'_{00010} \\
&\oplus 0\mathrm{x}1806060 \cdot x'_{01101} \oplus 0\mathrm{x}1806060 \cdot x'_{01111} \oplus 0\mathrm{x}60666 \cdot x'_{10001} \\
&\oplus 0\mathrm{x}60666 \cdot x'_{10011} \oplus 0\mathrm{x}ccc000c0 \cdot x'_{10101} \oplus 0\mathrm{x}ccc000c0 \cdot x'_{10111} \\
&\oplus 0\mathrm{x}60006066 \cdot x'_{11101} \oplus 0\mathrm{x}60006066 \cdot x'_{11111}
\end{aligned}
$$

$0\mathrm{x}30003033 \cdot x_{00000} \oplus 0\mathrm{x}30003033 \cdot x_{00010} \oplus 0\mathrm{x}3f03cf33 \cdot x_{01101} \oplus 0\mathrm{x}3f03cf33 \cdot x_{01111} \oplus 0\mathrm{x}30003033 \cdot x_{10001} \oplus 0\mathrm{x}30003033 \cdot x_{10011} \oplus 0\mathrm{x}6660006 \cdot x_{10101} \oplus 0\mathrm{x}6660006 \cdot x_{10111} \oplus 0\mathrm{x}33000303 \cdot x_{11101} \oplus 0\mathrm{x}33000303 \cdot x_{11111} \approx 0\mathrm{x}3033300 \cdot x'_{00000} \oplus 0\mathrm{x}3033300 \cdot x'_{00010} \oplus 0\mathrm{x}c0303000 \cdot x'_{01101} \oplus 0\mathrm{x}c0303000 \cdot x'_{01111} \oplus 0\mathrm{x}3033300 \cdot x'_{10001} \oplus 0\mathrm{x}3033300 \cdot x'_{10011} \oplus 0\mathrm{x}60006066 \cdot x'_{10101} \oplus 0\mathrm{x}60006066 \cdot x'_{10111} \oplus 0\mathrm{x}303330 \cdot x'_{11101} \oplus 0\mathrm{x}303330 \cdot x'_{11111}$

$0\mathrm{x}30333 \cdot x_{00000} \oplus 0\mathrm{x}30333 \cdot x_{00010} \oplus 0\mathrm{x}f03cf333 \cdot x_{01101} \oplus 0\mathrm{x}f03cf333 \cdot x_{01111} \oplus 0\mathrm{x}30333 \cdot x_{10001} \oplus 0\mathrm{x}30333 \cdot x_{10011} \oplus 0\mathrm{x}66600060 \cdot x_{10101} \oplus 0\mathrm{x}66600060 \cdot x_{10111} \oplus 0\mathrm{x}30003033 \cdot x_{11101} \oplus 0\mathrm{x}30003033 \cdot x_{11111} \approx 0\mathrm{x}30333000 \cdot x'_{00000} \oplus 0\mathrm{x}30333000 \cdot x'_{00010} \oplus 0\mathrm{x}303000c \cdot x'_{01101} \oplus 0\mathrm{x}303000c \cdot x'_{01111} \oplus 0\mathrm{x}30333000 \cdot x'_{10001} \oplus 0\mathrm{x}30333000 \cdot x'_{10011} \oplus 0\mathrm{x}60666 \cdot x'_{10101} \oplus 0\mathrm{x}60666 \cdot x'_{10111} \oplus 0\mathrm{x}3033300 \cdot x'_{11101} \oplus 0\mathrm{x}3033300 \cdot x'_{11111}$