Synthetic Linear Analysis: Improved Attacks on CubeHash and Rabbit

Yi Lu[1], Serge Vaudenay[2], Willi Meier[3],
Liping Ding[1], and Jianchun Jiang[1]

[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. It has been considered most important and difficult to analyze the bias and find a large bias regarding the security of crypto-systems, since the invention of linear cryptanalysis. The demonstration of a large bias will usually imply that the target crypto-system is not strong. Regarding the bias analysis, researchers often focus on 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. For CubeHash, the best linear attack on 11-round CubeHash with 2^{470} queries was proposed previously. 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^{513}, which is sharply close to the security parameter 2^{512} of CubeHash. Rabbit is a stream cipher among the finalists of ECRYPT Stream Cipher Project (eSTREAM). For Rabbit, the best linear attack with complexity 2^{141} was recently presented. Our synthetic bias analysis yields the improved attack with complexity 2^{136}. Moreover, it seems that our results might be further improved, according to our ongoing computations.
Keywords: bias, linear cryptanalysis, synthetic analysis, conditional dependence, CubeHash, Rabbit.

1 Introduction

It has been considered most important and difficult to analyze the bias and find a large bias regarding the security of crypto-systems, since the invention of linear cryptanalysis [6] 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. Unfortunately, it does not help much to analyze the bias for a broad class of problems.

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 "differ" from the terms occurring in the others. It is thus essential to conduct synthetic analysis to study these 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 an analysis to estimate the bias.

We apply our synthetic analysis to improve the most recent linear attack [1] on the hash function CubeHash [2]. CubeHash was selected to the second round of SHA-3 competition [8]. In [1], 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 [1]. 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^{513}, which is sharply close to the security parameter 2^{512} of CubeHash.

Meanwhile, our synthetic analysis is applied to the recent linear attack [5] on stream cipher Rabbit [3]. 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). In [5], the best linear attack with complexity 2^{141} was presented. As reference, Rabbit designers claim the security level 2^{128}. Our synthetic analysis applies to the main part of the bias analysis [5]. Our results yield the improved linear attack with complexity 2^{136}.

The rest of the paper is organized as follows. In Section 2, we give preliminary analysis on CubeHash. In Section 3, we introduce the synthetic approach to the bias analysis problem and discuss how to apply to CubeHash round function in details. In Section 4, we propose the synthetic bias analysis for the conditional dependent problem. In Section 5 and Section 6, we present our improved attacks on CubeHash and Rabbit. We conclude in Section 7.

2 Preliminary Analysis on CubeHash Round Function

The hash function CubeHash [2] was designed by Prof. Daniel J. Bernstein. It was one of the 14 candidates which were selected to the second round of SHA-3 competition [8]. 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 [1].

Each Round of CubeHash consists of two half rounds. Each half round consists of five steps, and only one step out of five introduces nonlinearity to the internal state by performing the modular addition operations. We will investigate the largest bias [1] 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. As reference, the security parameter is 2^{512}. We will improve the bias analysis of multiple linear approximations in [1]. Recall that the bias[4] of a binary random variable X is $\Pr[X = 0] - \Pr[X = 1]$. Our main focus is that, within each round, the linear approximations are *not* all independent pairwise. This can be justified by the fact that nonlinearity is introduced by two separate steps (Step 1 and Step 6) instead of one step within a round.

Let us start from a simple case of Round 7 first. We let 32 words $x_{00000}, x_{00001}, \ldots, x_{11111}$ to denote the internal states of 1024 bits (each word has 32 bits). The round transformation T can be described by the

[4] Our definition of bias is slightly different from [1], it was defined as $\Pr[X = 0] - 1/2$ in [1].

following ten steps of operations ('+' denotes modular addition):

$$\begin{aligned}
\text{Step 1:}\quad & x_{0n} + x_{1n} \to x_{1n} \text{ for all 4-bit } n\\
\text{Step 2:}\quad & x_{0n} \lll 7 \to x_{0n} \text{ for all 4-bit } n\\
\text{Step 3:}\quad & x_{00n} \leftrightarrow x_{01n} \text{ for all 3-bit } n\\
\text{Step 4:}\quad & x_{0n} \oplus x_{1n} \to x_{0n} \text{ for all 4-bit } n\\
\text{Step 5:}\quad & x_{1jk0m} \leftrightarrow x_{1jk1m} \text{ for all 1-bit } j,k,m\\
\text{Step 6:}\quad & x_{0n} + x_{1n} \to x_{1n} \text{ for all 4-bit } n\\
\text{Step 7:}\quad & x_{0n} \lll 11 \to x_{0n} \text{ for all 4-bit } n\\
\text{Step 8:}\quad & x_{0j0km} \leftrightarrow x_{0j1km} \text{ for all 1-bit } j,k,m\\
\text{Step 9:}\quad & x_{0n} \oplus x_{1n} \to x_{0n} \text{ for all 4-bit } n\\
\text{Step 10:}\quad & x_{1jkm0} \leftrightarrow x_{1jkm1} \text{ for all 1-bit } j,k,m
\end{aligned}$$

For our purpose, we use superscripts to represent the step number within the round. We let the states without superscripts to represent the states right at beginning of the round. The round number of the internal states which we study is clear from the context, and we omit it from the notations. At Step 1 of Round 7, the step operation allows us to deduce:

$$\begin{aligned}
& 0x300 \cdot x_{10100} \oplus 0x300 \cdot x_{10110} && (1)\\
= & 0x300 \cdot x_{10100} \oplus 0x300 \cdot x_{00100} \oplus 0x300 \cdot x_{00100} \oplus && (2)\\
& 0x300 \cdot x_{10110} \oplus 0x300 \cdot x_{00110} \oplus 0x300 \cdot x_{00110} && (3)\\
\approx & 0x300 \cdot x^1_{10100} \oplus 0x300 \cdot x^1_{10110} \oplus 0x300 \cdot x^1_{00100} \oplus 0x300 \cdot x^1_{00110} && (4)
\end{aligned}$$

We note that two linear approximations are introduced into (4):

$$\begin{aligned}
0x300 \cdot x^1_{10100} \oplus 0x300 \cdot x^1_{00100} \approx 0x300 \cdot (x^1_{10100} - x^1_{00100}) && (5)\\
0x300 \cdot x^1_{10110} \oplus 0x300 \cdot x^1_{00110} \approx 0x300 \cdot (x^1_{10110} - x^1_{00110}) && (6)
\end{aligned}$$

We continue on (4) from Step 2 through Step 5:

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

At Step 6, (7) can be rewritten as,

$$\begin{aligned}
&= 0x300 \cdot x^5_{10110} \oplus 0x300 \cdot x^5_{00110} \oplus 0x300 \cdot x^5_{00110} \oplus \\
&\quad 0x300 \cdot x^5_{10100} \oplus 0x300 \cdot x^5_{00100} \oplus 0x300 \cdot x^5_{00100} \oplus \\
&\quad 0x18000 \cdot x^5_{01100} \oplus 0x18000 \cdot x^5_{11100} \oplus 0x18000 \cdot x^5_{01110} \oplus 0x18000 \cdot x^5_{11110} \\
&\approx 0x300 \cdot x^6_{10110} \oplus 0x300 \cdot x^6_{00110} \oplus 0x300 \cdot x^6_{10100} \oplus 0x300 \cdot x^6_{00100} \oplus \\
&\quad 0x18000 \cdot x^6_{11100} \oplus 0x18000 \cdot x^6_{11110}
\end{aligned}$$

Four linear approximations are introduced in this step:

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

It is clear that the bias for the linear approximation at Round 7,

$$\begin{aligned}
&0x300 \cdot x_{10100} \oplus 0x300 \cdot x_{10110} \\
&\approx 0x180000 \cdot (x^{10}_{00000} \oplus x^{10}_{00010} \oplus x^{10}_{10001} \oplus x^{10}_{10011}) \oplus \\
&\quad 0x300 \cdot (x^{10}_{10101} \oplus x^{10}_{10111}) \oplus 0x18000 \cdot (x^{10}_{11101} \oplus x^{10}_{11111})
\end{aligned} \tag{12}$$

equals the combined bias of the six approximations (5), (6), (8), (9), (10), (11) holding simultaneously. Furthermore, if these approximations are independent, we apply Piling-up lemma [6] to deduce that the total bias is equal to the product of the six individual biases for each linear approximation. Unfortunately, as we demonstrate below, this independence assumption is not true.

Obviously, we can easily justify that Approximations (5), (6) are independent, because the involved states x_{10100}, x_{00100} in (5) are independent of the involved states x_{10110}, x_{00110} in (6); 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. The internal states are invertible with the CubeHash round function T, as each step operation is invertible. Thus, we can rewrite Approximations (8), (9), (10), (11) in terms of states right after step one as follows respectively,

$$\begin{aligned}
0x300 \cdot (x^1_{10100} + (x^1_{01110} \lll 7 \oplus x^1_{10110})) &\approx 0x300 \cdot (x^1_{10100} \\
&\quad \oplus x^1_{01110} \lll 7 \oplus x^1_{10110}) \quad (13) \\
0x300 \cdot (x^1_{10110} + (x^1_{01100} \lll 7 \oplus x^1_{10100})) &\approx 0x300 \cdot (x^1_{10110} \\
&\quad \oplus x^1_{01100} \lll 7 \oplus x^1_{10100}) \quad (14) \\
0x18000 \cdot (x^1_{11110} + (x^1_{00100} \lll 7 \oplus x^1_{11100})) &\approx 0x18000 \cdot (x^1_{11110} \\
&\quad \oplus x^1_{00100} \lll 7 \oplus x^1_{11100}) \quad (15) \\
0x18000 \cdot (x^1_{11100} + (x^1_{00110} \lll 7 \oplus x^1_{11110})) &\approx 0x18000 \cdot (x^1_{11100} \\
&\quad \oplus x^1_{00110} \lll 7 \oplus x^1_{11110}) \quad (16)
\end{aligned}$$

In the next section, we will discuss the synthetic bias analysis and apply it for CubeHash Round 7 to analyze (5), (6), (13), (14), (15) and (16).

3 The Synthetic Approach

When we study the combined bias of multiple Boolean functions, such as multiple linear approximations, it is common to assume that they are all independent pairwise. This way, the problem reduces to the bias computation of each Boolean function separately. Apparently, if the terms involved in each function are statistically independent of the terms in the other functions, we are sure that all function outputs 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 "differ" from the terms occurring in the other functions. For example, one might take it for granted that (5), (6), (8), (9), (10), (11) are all independent, as the terms occurring in any linear approximation never occurs in other approximations. As a matter of fact, as we will see later, after re-writing (8), (9), (10), (11) equivalently by (13), (14), (15), (16) respectively, they are *not* all independent.

It thus leads us naturally to the "Divide-and-Conquer" method to the bias analysis involving multiple Boolean functions. That is, we try to group multiple possibly dependent Boolean functions (eg. linear approximations with regards to CubeHash). The aim is that the functions in each group are dependent and the functions in different groups are independent. When grouping, it is desirable to make each group size as small as possible. The group size is referred to the number of 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 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. We will explain in details next on CubeHash.

3.1 Our Analysis on CubeHash Round Function

We will first see how to group the six approximations (5), (6), (13), (14), (15) and (16) for CubeHash Round 7. We look at (13) and (14) first. At first glance, it seems that they are dependent as both have x^1_{10110} and x^1_{10100}. However, we note that x^1_{01110}, x^1_{01100} only occurs once in (5), (6), (13), (14), (15), (16), i.e., neither occurs in (5), (6), (15), (16). From the fact that x^1_{01110}, x^1_{01100} are independent, we deduce that $x^1_{10100}, x^1_{01110} \lll 7$ are independent of $x^1_{10110}, x^1_{01100} \lll 7$. Thus, $x^1_{10100}, x^1_{01110} \lll 7 \oplus x^1_{10110}$ are independent of $x^1_{10110}, x^1_{01100} \lll 7$; and $x^1_{10100}, x^1_{01110} \lll 7 \oplus x^1_{10110}$ are independent of $x^1_{10110}, x^1_{01100} \lll 7 \oplus x^1_{10100}$. Consequently, we know that (13) and (14) are independent. As x^1_{10100} occurs in both (5) and (13), we group (5) and (13) together. Likewise, as x^1_{10110} occurs in both (6) and (14), we group (6) and (14) together.

Both (15), (16) involve x^1_{11110}, x^1_{11100}, and it thus seems that (15), (16) are dependent. We can use the fact that x^1_{00100}, x^1_{00110} are independent to show that $x^1_{11110}, x^1_{00100} \lll 7 \oplus x^1_{11100}$ are independent of $x^1_{11100}, x^1_{00110} \lll 7 \oplus x^1_{11110}$. So, we deduce (15) and (16) are independent. As (15), (16) relates to x^1_{00100}, x^1_{00110} respectively, x^1_{00100} is related to (5), (13), and x^1_{00110} is related to (6) and (14). Therefore, we are able to make two groups. Group One contains (5), (13), (15). Group Two contains (6), (14), (16). These two groups are independent as we have just explained above.

When it comes to the joint bias computation of a group of dependent linear approximations, in general, it is a computationally hard problem, although in certain cases it might be feasible to calculate the bias for a single linear approximation. For example, [4] is applicable to analyze the bias of a single linear approximation in our above CubeHash problem. Nevertheless, when the bias is large, we can always compute it empirically, as successfully showed with recent results on RC4 biases (eg. [9]). The direct bias computation when the bias is small is beyond the scope of this paper.

Our computations show that the joint bias for the group of approximations (5), (13), (15) holding simultaneously is around $2^{-2.5}$ and the joint bias for (6), (14), (16) is around $2^{-2.5}$.

Consequently, the total bias for the linear approximation (12) at Round 7, is calculated as $2^{-2.5} \times 2^{-2.5} = 2^{-5}$. In contrast, if the dependency within the round is ignored, we would have a smaller bias 2^{-6} at Round 7.

For CubeHash Round 8, we can show that six pairwise independent approximations arise at Step 1:

$$0x180000 \cdot (x^1_{00001} \oplus x^1_{10001}) \approx 0x180000 \cdot (x^1_{00001} - x^1_{10001}) \tag{17}$$
$$0x180000 \cdot (x^1_{00011} \oplus x^1_{10011}) \approx 0x180000 \cdot (x^1_{00011} - x^1_{10011}) \tag{18}$$
$$0x300 \cdot (x^1_{00101} \oplus x^1_{10101}) \approx 0x300 \cdot (x^1_{00101} - x^1_{10101}) \tag{19}$$
$$0x300 \cdot (x^1_{00111} \oplus x^1_{10111}) \approx 0x300 \cdot (x^1_{00111} - x^1_{10111}) \tag{20}$$
$$0x18000 \cdot (x^1_{01101} \oplus x^1_{11101}) \approx 0x18000 \cdot (x^1_{01101} - x^1_{11101}) \tag{21}$$
$$0x18000 \cdot (x^1_{01111} \oplus x^1_{11111}) \approx 0x18000 \cdot (x^1_{01111} - x^1_{11111}) \tag{22}$$

Eight pairwise independent approximations arise at Step 6. They are presented in terms of states right after step one of the round:

$$\begin{aligned} 0x180000 \cdot (x^1_{10011} + (x^1_{01001} \lll 7 \oplus x^1_{10001})) \approx \\ 0x180000 \cdot (x^1_{10011} \oplus x^1_{01001} \lll 7 \oplus x^1_{10001}) \end{aligned} \tag{23}$$
$$\begin{aligned} 0x180000 \cdot (x^1_{10001} + (x^1_{01011} \lll 7 \oplus x^1_{10011})) \approx \\ 0x180000 \cdot (x^1_{10001} \oplus x^1_{01011} \lll 7 \oplus x^1_{10011}) \end{aligned} \tag{24}$$
$$\begin{aligned} 0xc00300 \cdot (x^1_{10111} + (x^1_{01101} \lll 7 \oplus x^1_{10101})) \approx \\ 0xc00300 \cdot (x^1_{10111} \oplus x^1_{01101} \lll 7 \oplus x^1_{10101}) \end{aligned} \tag{25}$$
$$\begin{aligned} 0xc00300 \cdot (x^1_{10101} + (x^1_{01111} \lll 7 \oplus x^1_{10111})) \approx \\ 0xc00300 \cdot (x^1_{10101} \oplus x^1_{01111} \lll 7 \oplus x^1_{10111}) \end{aligned} \tag{26}$$
$$\begin{aligned} 0xc000000 \cdot (x^1_{11010} + (x^1_{00000} \lll 7 \oplus x^1_{11000})) \approx \\ 0xc000000 \cdot (x^1_{11010} \oplus x^1_{00000} \lll 7 \oplus x^1_{11000}) \end{aligned} \tag{27}$$
$$\begin{aligned} 0xc000000 \cdot (x^1_{11011} + (x^1_{00001} \lll 7 \oplus x^1_{11001})) \approx \\ 0xc000000 \cdot (x^1_{11011} \oplus x^1_{00001} \lll 7 \oplus x^1_{11001}) \end{aligned} \tag{28}$$
$$\begin{aligned} 0xc000000 \cdot (x^1_{11000} + (x^1_{00010} \lll 7 \oplus x^1_{11010})) \approx \\ 0xc000000 \cdot (x^1_{11000} \oplus x^1_{00010} \lll 7 \oplus x^1_{11010}) \end{aligned} \tag{29}$$
$$\begin{aligned} 0xc000000 \cdot (x^1_{11001} + (x^1_{00011} \lll 7 \oplus x^1_{11011})) \approx \\ 0xc000000 \cdot (x^1_{11001} \oplus x^1_{00011} \lll 7 \oplus x^1_{11011}) \end{aligned} \tag{30}$$

Thus, we have 6+8=14 linear approximations involved in this round. As was done for Round 7, we can demonstrate that these 14 approximations fall into four independent groups.

Group One: (17), (18), (23), (24), (28), (30).

Group Two: (19), (20), (21), (22), (25), (26).

Group Three: (27).

Group Four: (29).

As Group Three and Group Four each contains only one approximation, we easily know the bias is 2^{-1} for each group directly from [4]. 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^{-16}.

4 Synthetic Bias Analysis on the Conditional Dependent Problem

When analyzing CubeHash round function, we note a new bias problem, which we shall call conditional dependence from now on. This is in contrast to the well-known concept of conditional independence in statistics. Let X, Y, Z be random variables. 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. In our problem, X, Y are statistically independent variables, but X, Z are dependent as well as Y, Z. We say that X, Y are conditional dependent given Z. We are concerned with the bias of $f_1(X) \oplus f_2(Y) \oplus f_3(Z)$ for Boolean functions f_1, f_2, f_3. 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. For simplicity, they are denoted by A, B, C in shorthand. We assume that we already know the bias for A, B, C respectively. The main question is, 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. We assume it infeasible to compute directly. Our first solution is to obtain the bias for $A \oplus C$ (or $A \oplus B$) first. Then, estimate the bias for $A \oplus B \oplus C$ by taking either of the two

$$\text{Bias}(A \oplus C) \cdot \text{Bias}(B),\ \ \text{Bias}(A \oplus B) \cdot \text{Bias}(C).$$

Here, we consider only one dependence relation and ignore another dependence relation.

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

$$qq'p_+p'_+ + (1-q)(1-q')p_-p'_- - q(1-q')p_+p'_- - (1-q)q'p_-p'_+ \quad (31)$$

5 Improved Attacks on CubeHash

Using our synthetic analysis, we analyzed all the 11 rounds for CubeHash. We give our results in Table 1. Note that we can show that all the linear approximations for Round 5 are independent and for Round 6, so no bias improvement is possible for Round 5 as well as Round 6. Due to the dependence within each round, we are able to improve the bias estimate for 11-round CubeHash from 2^{-234} in [1] to $2^{-207.1}$. This gives an improved attack for 11-round CubeHash with complexity $2^{414.2}$.

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

round	1	2	3	4	5	6	7	8	9	10	11	total
our bias	2^{-29}	$2^{-35.7}$	$2^{-16.9}$	2^{-13}	2^{-4}	2^{-2}	2^{-5}	$2^{-13.8}$	$2^{-18.7}$	$2^{-36.5}$	$2^{-32.5}$	$2^{-207.1}$
paper [1]	2^{-34}	2^{-40}	2^{-18}	2^{-14}	2^{-4}	2^{-2}	2^{-6}	2^{-16}	2^{-22}	2^{-42}	2^{-36}	2^{-234}

We can extend our above results to attack 12-round CubeHash. Our analysis shows that by choosing the same output masks from the set $\{0x600, 0x18000, 0x180000, 0xc000000, 0xc0000000\}$ for x_{01101} and x_{01111} at the end of Round 5, going backwards 6 rounds, forwards 6 rounds, we get[5] five new linear approximations (given in Appendix A) 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. The bias for its first round is 2^{-54}, assuming all linear approximations are independent[6].

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. Our current quick results show that the bias for round 6 and round 7 can be improved to 2^{-6}, and the bias for round 4 and round 5 can be improved to 2^{-16}. Thus, we have the improved bias estimate $2^{-259.1}$ for 12-round CubeHash. By using the five equal biases, we have an attack complexity $2^{(-256.5)\times(-2)}$, ie. $O(2^{513})$.

6 Our Improved Analysis on Stream Cipher Rabbit

Rabbit [3] 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 brief description on Rabbit in Appendix B. Recently, the bias for Rabbit keystream outputs, $0x606 \cdot s_{i+1}^{[47..32]} \oplus 0x606 \cdot s_{i+1}^{[79..64]} \oplus 0x606 \cdot s_{i+1}^{[111..96]}$ was estimated to be $2^{-70.5}$ in [5]. It yields the best distinguishing attack with complexity 2^{141}, which is still above the claimed security level 2^{128}.

In this section, we apply our synthetic approach to analyze the main part of the bias analysis, i.e., the total combined bias of the six linear approximations below for $m = 0x606, m' = 0x6060000$ (for simplicity we omit the irrelevant subscripts i from the variables g):

$$m \cdot (g_2 + g_1 \lll 16 + g_0 \lll 16) \approx m \cdot (g_2 \oplus g_1 \lll 16 \oplus g_0 \lll 16) \quad (32)$$

$$m \cdot (g_4 + g_3 \lll 16 + g_2 \lll 16) \approx m \cdot (g_4 \oplus g_3 \lll 16 \oplus g_2 \lll 16) \quad (33)$$

$$m \cdot (g_6 + g_5 \lll 16 + g_4 \lll 16) \approx m \cdot (g_6 \oplus g_5 \lll 16 \oplus g_4 \lll 16) \quad (34)$$

$$m' \cdot (g_1 + g_0 \lll 8 + g_7) \approx m' \cdot (g_1 \oplus g_0 \lll 8 \oplus g_7) \quad (35)$$

$$m' \cdot (g_3 + g_2 \lll 8 + g_1) \approx m' \cdot (g_3 \oplus g_2 \lll 8 \oplus g_1) \quad (36)$$

$$m' \cdot (g_7 + g_6 \lll 8 + g_5) \approx m' \cdot (g_7 \oplus g_6 \lll 8 \oplus g_5) \quad (37)$$

[5] 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.

[6] As our computation is going on, we expect that our previous analysis on the internal round dependence would further improve it.

Let Group One contain (32), (33), (36) and Group Two contain (34), (37). Following Sect. 3, we can demonstrate that the linear approximations in Group One are independent from those in Group Two. Nonetheless, given (35), the two groups are *not* independent. We let A denote the corresponding[7] Boolean function of (35), and let B, C denote the corresponding[8] Boolean function for Group One, Group Two respectively. Obviously, this is a conditional dependent bias problem as we proposed in Sect. 4. Using our first solution in Sect. 4, we compute the bias for $A \oplus B, C$ respectively and get $2^{-11.4}, 2^{-6}$. We estimate the combined bias for above six linear approximations by

$$2^{-11.4} \times 2^{-6} = 2^{-17.4}. \tag{38}$$

Now, we want to apply our black-box solution (31) in Sect. 4. 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 get in hexadecimal form: $q = 0x24a6/0x4000$, $p_+ = 0x1e0e6fc1/(0x24a6 * 2^{25})$, $p_- = 0x123210ab/(0x1b5a * 2^{25})$. 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 get $q' = 0xafed/0x10000$, $p'_+ = 0xd4698c87/(0xafed * 2^{21})$, $p'_- = 0x4ea00ceb/(0x5013 * 2^{21})$. 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[9] . By (31), we estimate the bias $2^{-17.5}$ for $A \oplus B \oplus C$. This result agrees with our first estimation (38). Note that based on the naive independence assumption, this combined bias is estimated to be smaller, i.e., 2^{-20}, according to [5]. Consequently, we have an improved attack on Rabbit with complexity 2^{136}, based on [5].

7 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. Our synthetic approach helps make the task of bias analysis easier when multiple Boolean functions are involved. We also propose a

[7] We obtain it by replacing '$\approx$' with '$\oplus$' in (35).

[8] We obtain it by replacing '$\approx$' with '$\oplus$' in all the linear approximations in the group and XORing them together.

[9] The computations were run several times and we always got these same statistics.

conditional dependent bias problem. Based on naive heuristics and certain ideal assumptions, we give the synthetic bias analysis to estimate the bias. Our synthetic approach is successfully applied to improve the best linear attacks [1, 5] 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^{513}, which is sharply close to the security parameter 2^{512} of CubeHash. We also give an improved attack on Rabbit with complexity 2^{136}. Moreover, it seems that our results might be further improved, from our ongoing computations.

Acknowledgments

This work 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. T. Ashur, O. Dunkelman, *Linear analysis of reduced-round CubeHash*, ACNS 2011, LNCS vol.6715, pp. 462-478, Springer-Verlag, 2011.
2. D.J.Bernstein, CubeHash specification (2.B.1), submission to NIST, 2009.
3. M. Boesgaard, M. Vesterager, T. Christensen and E. Zenner, *The stream cipher Rabbit*, the ECRYPT stream cipher project, `http://www.ecrypt.eu.org/stream/`.
4. J.Y.Cho, J.Pieprzyk, *Multiple modular additions and crossword puzzle attack on NLSv2*, ISC 2007, LNCS vol. 4779, pp.230-248, 2007.
5. Y. Lu, Y. Desmedt, *Improved distinguishing attack on Rabbit*, ISC 2010, LNCS vol. 6531, pp.17-23, Springer-Verlag, 2011.
6. M. Matsui, *Linear cryptanalysis method for des cipher*, EUROCRYPT 93, LNCS vol. 765, pp.386-397, 1994.
7. A. J. Menezes, P. C. van. Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*, CRC, 1996.
8. N.I.S.T., *Cryptographic hash algorithm competition*, `http://www.nist.gov/hash-competition`.
9. 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: New Linear Approximations on 12-Round CubeHash

The five new linear approximations on 12-round CubeHash, which we used in Section 5, are given below (x, x' denote the inputs, outputs respectively):

$$0x18199800 \cdot x_{00000} \oplus 0x18199800 \cdot x_{00010} \oplus 0xe7999f81 \cdot x_{01101} \oplus 0xe7999f81 \cdot x_{01111} \oplus 0x18199800 \cdot x_{10001} \oplus 0x18199800 \cdot x_{10011} \oplus 0x30333 \cdot x_{10101} \oplus 0x30333 \cdot x_{10111} \oplus 0x1819980 \cdot x_{11101} \oplus 0x1819980 \cdot x_{11111} \approx 0x99800181 \cdot x'_{00000} \oplus 0x99800181 \cdot x'_{00010} \oplus 0x18006018 \cdot x'_{01101} \oplus 0x18006018 \cdot x'_{01111} \oplus 0x99800181 \cdot x'_{10001} \oplus 0x99800181 \cdot x'_{10011} \oplus 0x30333000 \cdot x'_{10101} \oplus 0x30333000 \cdot x'_{10111} \oplus 0x19980018 \cdot x'_{11101} \oplus 0x19980018 \cdot x'_{11111}$$

$$0x6660006 \cdot x_{00000} \oplus 0x6660006 \cdot x_{00010} \oplus 0xe667e079 \cdot x_{01101} \oplus 0xe667e079 \cdot x_{01111} \oplus 0x6660006 \cdot x_{10001} \oplus 0x6660006 \cdot x_{10011} \oplus 0xc0ccc0 \cdot x_{10101} \oplus 0xc0ccc0 \cdot x_{10111} \oplus 0x60666000 \cdot x_{11101} \oplus 0x60666000 \cdot x_{11111} \approx 0x60006066 \cdot x'_{00000} \oplus 0x60006066 \cdot x'_{00010} \oplus 0x180606 \cdot x'_{01101} \oplus 0x180606 \cdot x'_{01111} \oplus 0x60006066 \cdot x'_{10001} \oplus 0x60006066 \cdot x'_{10011} \oplus 0xccc000c \cdot x'_{10101} \oplus 0xccc000c \cdot x'_{10111} \oplus 0x66000606 \cdot x'_{11101} \oplus 0x66000606 \cdot x'_{11111}$$

$$0x66600060 \cdot x_{00000} \oplus 0x66600060 \cdot x_{00010} \oplus 0x667e079e \cdot x_{01101} \oplus 0x667e079e \cdot x_{01111} \oplus 0x66600060 \cdot x_{10001} \oplus 0x66600060 \cdot x_{10011} \oplus 0xc0ccc00 \cdot x_{10101} \oplus 0xc0ccc00 \cdot x_{10111} \oplus 0x6660006 \cdot x_{11101} \oplus 0x6660006 \cdot x_{11111} \approx 0x60666 \cdot x'_{00000} \oplus 0x60666 \cdot x'_{00010} \oplus 0x1806060 \cdot x'_{01101} \oplus 0x1806060 \cdot x'_{01111} \oplus 0x60666 \cdot x'_{10001} \oplus 0x60666 \cdot x'_{10011} \oplus 0xccc000c0 \cdot x'_{10101} \oplus 0xccc000c0 \cdot x'_{10111} \oplus 0x60006066 \cdot x'_{11101} \oplus 0x60006066 \cdot x'_{11111}$$

$$0x30003033 \cdot x_{00000} \oplus 0x30003033 \cdot x_{00010} \oplus 0x3f03cf33 \cdot x_{01101} \oplus 0x3f03cf33 \cdot x_{01111} \oplus 0x30003033 \cdot x_{10001} \oplus 0x30003033 \cdot x_{10011} \oplus 0x6660006 \cdot x_{10101} \oplus 0x6660006 \cdot x_{10111} \oplus 0x33000303 \cdot x_{11101} \oplus 0x33000303 \cdot x_{11111} \approx 0x3033300 \cdot x'_{00000} \oplus 0x3033300 \cdot x'_{00010} \oplus 0xc0303000 \cdot x'_{01101} \oplus 0xc0303000 \cdot x'_{01111} \oplus 0x3033300 \cdot x'_{10001} \oplus 0x3033300 \cdot x'_{10011} \oplus 0x60006066 \cdot x'_{10101} \oplus 0x60006066 \cdot x'_{10111} \oplus 0x303330 \cdot x'_{11101} \oplus 0x303330 \cdot x'_{11111}$$

$$
\begin{aligned}
&0x30333 \cdot x_{00000} \oplus 0x30333 \cdot x_{00010} \oplus 0xf03cf333 \cdot x_{01101} \\
&\oplus 0xf03cf333 \cdot x_{01111} \oplus 0x30333 \cdot x_{10001} \oplus 0x30333 \cdot x_{10011} \\
&\oplus 0x66600060 \cdot x_{10101} \oplus 0x66600060 \cdot x_{10111} \oplus 0x30003033 \cdot x_{11101} \\
&\oplus 0x30003033 \cdot x_{11111} \approx 0x30333000 \cdot x'_{00000} \oplus 0x30333000 \cdot x'_{00010} \\
&\oplus 0x303000c \cdot x'_{01101} \oplus 0x303000c \cdot x'_{01111} \oplus 0x30333000 \cdot x'_{10001} \\
&\oplus 0x30333000 \cdot x'_{10011} \oplus 0x60666 \cdot x'_{10101} \oplus 0x60666 \cdot x'_{10111} \\
&\oplus 0x3033300 \cdot x'_{11101} \oplus 0x3033300 \cdot x'_{11111}
\end{aligned}
$$

Appendix B: Short Description on Stream Cipher Rabbit

We give a short description on Rabbit here. We refer to [3, 5] 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,

$$x_{0,i+1} = g_{0,i} + (g_{7,i} \lll 16) + (g_{6,i} \lll 16) \quad (39)$$
$$x_{1,i+1} = g_{1,i} + (g_{0,i} \lll 8) + g_{7,i} \quad (40)$$
$$x_{2,i+1} = g_{2,i} + (g_{1,i} \lll 16) + (g_{0,i} \lll 16) \quad (41)$$
$$x_{3,i+1} = g_{3,i} + (g_{2,i} \lll 8) + g_{1,i} \quad (42)$$
$$x_{4,i+1} = g_{4,i} + (g_{3,i} \lll 16) + (g_{2,i} \lll 16) \quad (43)$$
$$x_{5,i+1} = g_{5,i} + (g_{4,i} \lll 8) + g_{3,i} \quad (44)$$
$$x_{6,i+1} = g_{6,i} + (g_{5,i} \lll 16) + (g_{4,i} \lll 16) \quad (45)$$
$$x_{7,i+1} = g_{7,i} + (g_{6,i} \lll 8) + g_{5,i} \quad (46)$$

where $\lll$ denotes left bit-wise rotation and all additions are computed modulo 2^{32}. The description of computing g's (see [3, 5]) is not relevant for us and we omit it here.