Rotational Cryptanalysis of ARX Revisited

Dmitry Khovratovich[1], Ivica Nikolić[2], Josef Pieprzyk[3], Przemysław Sokołowski[4], Ron Steinfeld[5]

[1] University of Luxembourg, Luxembourg
[2] Nanyang Technological University, Singapore
[3] Queensland University of Technology, Australia
[4] Adam Mickiewicz University, Poland
[5] Monash University, Australia

dmitry.khovratovich@uni.lu, inikolic@ntu.edu.sg, josef.pieprzyk@qut.edu.au, przemeks@amu.edu.pl, ron.steinfeld@monash.edu

Abstract. Rotational cryptanalysis is a probabilistic attack applicable to word oriented designs that use (almost) rotation-invariant constants. It is believed that the success probability of rotational cryptanalysis against ciphers and functions based on modular additions, rotations and XORs, can be computed only by counting the number of additions. We show that this simple formula is incorrect due to the invalid Markov cipher assumption used for computing the probability. More precisely, we show that chained modular additions used in ARX ciphers do not form a Markov chain with regards to rotational analysis, thus the rotational probability cannot be computed as a simple product of rotational probabilities of individual modular additions. We provide a precise value of the probability of such chains and give a new algorithm for computing the rotational probability of ARX ciphers. We use the algorithm to correct the rotational attacks on BLAKE2 and to provide valid rotational attacks against the simplified version of Skein.

Keywords: rotational cryptanalysis, Markov cipher, Markov chain, Skein, BLAKE2,

1 Introduction

Rotational cryptanalysis, formally introduced in [13], is a probabilistic attack that follows the evolution of a so-called rotational pair through the rounds of a function (or a cipher). It targets word-oriented functions and, for a successful application, it requires all constants used in the functions to preserve their values after a rotation. Rotational cryptanalysis has been launched against the building blocks of several hash functions such as BLAKE2 [1, 10], Keccak [19], Shabal [3, 22], Skein [13–15], SM3 [16], (modified) SIMD and BMW [20], etc.

The chance of success of probabilistic attacks against a particular cryptographic primitive is computed by finding the probability that inputs and corresponding outputs (produced after application of the primitive) have a specific property that is unexpected for a random permutation/function. For instance, in differential attacks [2], the probability that a pair of plaintexts with a specific difference generates a pair of ciphertexts with another specific difference is unusually high. To compute the probability, one has to investigate the propagation of the input difference through rounds of the cipher. The propagation feature is known as a differential characteristic and its probability is computed as a product of the probabilities of all single-round characteristics. Lai and Massey [17] show that this shortcut in the probability calculation is correct as long as the keys used in the rounds are chosen at random independently and uniformly and the cipher can be modeled as a Markov cipher. This is to say that an iterated cipher can be seen as a sequence of independent rounds, where the round keys are independent and the differential probability of the round functions is independent of the inputs (but not of the input/output difference). Lai and Massey argue that in such a case, the differences after each round (which follows a differential characteristic) can be modeled by a Markov chain. Consequently, the differential probability of each round depends on the input and the output difference only (the differences in previous rounds can be ignored).

The Markov cipher assumption (which results in a Markov chain assumption) is used in practically all differential attacks as well as in other probabilistic attacks. This is despite the fact that the round keys are not independent but are produced from a single master by the key scheduling algorithm. Moreover, some ciphers do not use round keys in every round (after each non-linear operation) – see LED [11] and Zorro [9], for an example of such ciphers. Nevertheless, in practical cryptographic primitives, the role of independent and random round keys is usually replaced by some state words. They introduce sufficient entropy so one can argue that the Markov chain assumption holds. However, exceptions from this rule with respect to differential analysis can be found in the case of ARX or ARX-like primitives – we refer the reader to [4, 18, 21, 23, 24] for such examples[1].

Likewise, in the case of rotational cryptanalysis, probabilities are computed under the Markov chain assumption. This makes rotational cryptanalysis exceptionally easy to apply as the rotational probability of a cryptographic primitive is the product of rotational probabilities of all the operations/rounds that are used in the primitive. For instance, in schemes based on modular additions, rotations and XORs (further referred as ARX primitives), only modular addition has rotational probability p_+ smaller than 1. Thus the rotational probability of the whole ARX primitive can be computed easily. If it uses q additions, then the rotational probability is p_+^q. With minor modifications, this formula has been used in rotational cryptanalysis of ARX designs.

Our Contribution. We show that in rotational analysis of cryptographic algorithms based on ARX, the Markov chain assumption does not always hold. In particular, we establish that rotational probability of an ARX primitive

[1] We are grateful to the reviewers of FSE'15 for pointing out these exceptions.

depends not only on the number of modular additions but also on their positions. In general, the more modular additions are chained (output of the previous additions is the input of the next), the smaller the probability. We work out an explicit formula for the probability of such chained additions and show that the rotational probability of ARX should be computed as the product of the rotational probabilities of modular addition chains. We also reveal that the way the round keys are incorporated into the state, plays a crucial role in calculation of the rotational probability. When round keys are XORed to the state, they might break modular addition chains and thus increase the probability. On the other hand, if they are merged using modular addition, the rotational probability of ARX may be reduced.

Chained modular additions are used in ARX hash functions such as BLAKE2 [1] and `Skein` [6, 7]. Both functions have been successfully attacked using rotational cryptanalysis (in fact, rotational cryptanalysis was officially introduced as a method of analysis on an instance of `Skein` [13]). The success can be attributed to the lack of constants or to the use of (almost rotational) constants in the designs. We correct the claimed complexity of rotational attacks against BLAKE2 [10]. Our analysis suggests that, due to the aforementioned chains of modular additions, the rotational attacks are applicable only to 7 rounds of BLAKE2 instead of the claimed full 12 rounds in [10]. We also provide analysis of the compression function of `Skein`. Note, in [13] it is shown that the compression function reduced to 42 rounds is vulnerable to rotational attacks, and further, the attack was extended in [14, 15] to include a rebound part, but the rotational part of the attack is still on 42 rounds. We show that due to the structure of addition chains in `Skein`, the rotational attacks on a version of `Skein` without any subkey additions, works for 24-28 rounds only (depending on the rotation amount).

The correctness of the results presented in this paper has been experimentally verified on ARX primitives with different state sizes and different numbers of rounds. A part of the experiments is given in Appendix A.

2 Differential Cryptanalysis and Markov Ciphers

Differential cryptanalysis, introduced by Biham and Shamir [2], is a probabilistic chosen-plaintext attack. It works against ciphers (and other cryptographic primitives) whose differential properties deviate from those expected by a random cipher. Consider a cipher. We can determine a collection of input (plaintext) differences and the corresponding output (ciphertext) differences. In differential cryptanalysis, we try to identify a pair (α, β), which occurs with a much higher probability than other (possible) pairs, where α is an input (plaintext) difference and β is an output (ciphertext) difference. This property allows to distinguish the cipher from a random permutation and in many cases may lead to secret key recovery (or total cipher break). Finding two differences (α, β) (the pair (α, β) is called a differential) that maximizes the probability is, in fact, the main goal (and the most challenging task) of differential cryptanalysis.

For a chosen plaintext difference, the ciphertext difference can be found by propagating the plaintext difference through the encryption function of the cipher. Most ciphers are iterated, i.e. their encryption function consists of repetitive application of some (possibly weak) non-linear round function $Y = f(X, Z_i)$, where X is a state at the beginning of the round, Z_i is a key used in the round i and Y is an output state. To find β, an initial plaintext difference α is propagated round-by-round and after r rounds (for an r-round cipher), the ciphertext difference β can be obtained. The evolution of differences generated after each round is called a differential characteristic and can be represented by the following sequence $\alpha = \varDelta Y(0), \varDelta Y(1), \ldots, \varDelta Y(r) = \beta$, where $\varDelta Y(i)$ is the difference at the output of the ith round.

The efficiency of differential cryptanalysis is tightly related to the probability of differentials (differential characteristics) – the higher the probability the lower the complexity. Lai and Massey [17] put a focus on the probability of differential characteristics and study conditions for differential characteristics to form a Markov chain. Note that a sequence of discrete random variables $v_0, \ldots, v_r$ is a *Markov chain* if, for $0 \leq i < r$,

$$\Pr(v_{i+1} = \beta_{i+1} | v_i = \beta_i, v_{i-1} = \beta_{i-1}, \ldots, v_0 = \beta_0) = \Pr(v_{i+1} = \beta_{i+1} | v_i = \beta_i).$$

They introduce the notion of *Markov cipher* as follows.

Definition 1. *An iterated cipher with round function $Y = f(X, Z)$ is a Markov cipher if for all choices of $\alpha \neq 0, \beta \neq 0$,*

$$\Pr(\varDelta Y = \beta | \varDelta X = \alpha, X = \gamma)$$

is independent of γ when the round key Z is uniformly random, or equivalently, if

$$\Pr_Z(\varDelta Y = \beta | \varDelta X = \alpha, X = \gamma) = \Pr_Z(\varDelta Y(1) = \beta_1 | \varDelta X = \alpha)$$

for all choices of γ.

Their main result is described by the following theorem.

Theorem 1 ([17]). *If an r-round iterated cipher is a Markov cipher and the r round subkeys are independent and uniformly random, then the sequence of differences $\varDelta Y(0), \varDelta Y(1), \ldots, \varDelta Y(r)$ is a Markov chain.*

In other words, the probability of differential characteristics is a product of the probabilities of the single-round characteristics (as they form a Markov chain), as long as *the probabilities of the single-round characteristics do not depend on the value of the input state*, where round keys are independent and uniformly at random (if the cipher is Markov).

It is important to notice that Lai-Massey result does not apply to the cases when round keys are not injected in every round (see LED [11], Zorro [9]) or when they are dependent (this is the case for all modern ciphers as all round keys are produced from a single master key).

However, even in such cases, the probability of a characteristic for the whole cipher is still computed as a product of probabilities of single-round characteristics. The justification is based on the fact (or belief) that a round function introduces enough entropy. The entropy makes the inputs to the next round look completely random thus multiplying the probabilities of single-round characteristics seems to be a valid estimate of the probability of the whole characteristic.

3 Rotational Cryptanalysis and Chained Modular Additions

Rotational cryptanalysis is a probabilistic chosen-plaintext attack. It turns to be an effective cryptanalytical tool against ciphers and hash functions that are based on the three operations: modular additions (denoted as $+$), rotations (denoted as $\lll r$) and XORs (denoted as $\oplus$). Cryptographic algorithms that use the three operations are called ARX primitives. A crucial requirement for rotational cryptanalysis to work effectively is that all constants used in the ARX primitive must preserve their values when rotated[2]. To launch a rotational attack, one starts from a rotational pair of inputs, i.e. two states such that in the first state all words are chosen at random, while the second state is produced by rotating the words of the first state by a fixed amount. If for such input pair (called a rotational pair), the corresponding output pair of the ARX primitive is also rotational, with a probability higher than for a random permutation (or a function), then this property can be used to distinguish the ARX primitive from a random permutation (or a function).

It is claimed in [13] that rotational probabilities of an ARX primitive could be found by multiplying individual rotational probabilities of all the transformations used in the primitive. As ARX is composed of three distinct operations only, the rotational probabilities of addition, rotation and XOR can be computed. Probabilities of the latter two are:

$$\Pr((x \lll r_1) \lll r_2 = (x \lll r_2) \lll r_1) = 1,$$
$$\Pr(x \lll r \oplus y \lll r = (x \oplus y) \lll r) = 1,$$

while rotational probability of modular addition is given by the following lemma.

Lemma 1 (Daum[5]).

$$\Pr((x+y) \lll r = x \lll r + y \lll r) = \frac{1}{4}(1 + 2^{r-n} + 2^{-r} + 2^{-n}).$$

[2] This requirement can be relaxed to some extent and instead of assuming completely rotational constant, one can work with constants that are almost rotational, i.e. the XOR difference between the initial constant and the rotated one gives a word of small Hamming weight.

As modular addition only has rotational probability less than one, it is concluded that the theorem given below holds.

Theorem 2 ([13]). *Let q be the number of modular additions in an ARX primitive (that has an arbitrary number of rotations and XORs). Then rotational probability of the ARX primitive is p_+^q, where p_+ is rotational probability of modular addition (which depends on the rotation parameter r and the word size n).*

In other words, to find rotational probability of ARX, one has to count the number of additions q only. If $p_+^q > 2^{-m}$, where m is the state size, then the primitive is susceptible to rotational cryptanalysis. Theorem 2 is true under the (tacit) assumption that an ARX cipher is Markov and round keys are chosen independently and uniformly at random. Note that as in differential cryptanalysis, if round keys are not used in every round, then randomness (required by the Markov chain) must come from the state words, which are updated by the three operations of ARX. Rotations and XORs have rotational probability of 1 and thus are independent of the inputs. The case of modular addition is different. Rotational probability of modular addition is as determined by Lemma 1 as long as inputs are random. The output of modular addition is biased when an input pair is rotational. That is if $(x + y) \lll r = x \lll r + y \lll r$ and $r > 0$, then the value $z = x + y$ is biased. If the output of modular addition is taken as input to another addition, then rotational probability of the second addition may not follow Lemma 1 although Theorem 2 states that this should be irrelevant.

To illustrate the issue, let us focus on two toy ARX primitives given in Fig. 1. Each of them has three inputs a, b, k, two outputs u and w, and uses two modular additions. If the rotation amount r equals 1 and the word size is 64 bits, then by Lemma 1, rotational probability of modular addition is $2^{-1.415}$ and thus by Theorem 2, rotational probability of both of the primitives should be $2^{-2.83}$. Note that for rotation amount of 1, rotational probability of modular addition strongly depends on the value of the most significant bits of the inputs. More precisely, the sum of the most significant bits of the inputs should not be larger than 1.

In the ARX construction on the left of Fig. 1, the two modular additions are chained, i.e. the output of the first is the input to the second. The most significant bit of the word $d = a + b$, when $(a + b) \lll 1 = a \lll 1 + b \lll 1$, is biased towards 1. Therefore, the second modular addition $u = k + d$ has rotational probability smaller than the one given by Lemma 1. As the result, Theorem 2 fails to give the correct probability.

In the ARX on the right of Fig. 1, the two modular additions are separated by rotation. In this case, although the most significant bit of d is still biased towards 1, the rotation moves this bit to a different position, where the mentioned bias is negligible for the computation of the rotational probability. Furthermore, the least significant bit of $d \lll \overline{r}$ becomes a completely random

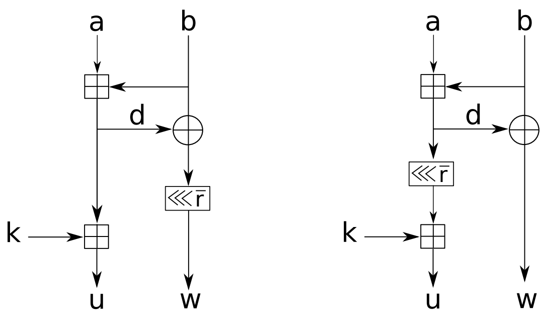

Fig. 1. Two ARX primitives with equal number of additions but different rotational probabilities.

bit and thus the second modular addition $d \lll \bar{r} + k$ has probability given by Lemma 1. Therefore, in this case, Theorem 2 works as expected.

These two examples suggest that rotational probability of ARX cannot be computed simply by counting the number of modular additions. Instead, one has to investigate the relative positions of modular additions, i.e. if they are chained or separated by rotations. In fact, the longer the chain of modular additions, the lower the rotational probability for each consecutive addition. The rotational probability of chained modular additions is given by the following lemma.

Lemma 2 (Chained modular additions). *Let $a_1, \ldots, a_k$ be n-bit words chosen at random and let r be a positive integer such that $0 < r < n$. Then*

$$\begin{aligned}&\Pr([(a_1 + a_2) \lll r = a_1 \lll r + a_2 \lll r] \wedge \\ &\wedge[(a_1 + a_2 + a_3) \lll r = a_1 \lll r + a_2 \lll r + a_3 \lll r] \wedge \\ &\wedge \ldots \\ &\wedge[(a_1 + \ldots + a_k) \lll r = a_1 \lll r + \ldots + a_k \lll r]) = \\ &= \frac{1}{2^{nk}} \binom{k + 2^r - 1}{2^r - 1} \binom{k + 2^{n-r} - 1}{2^{n-r} - 1}\end{aligned}$$

Proof. First we consider the rotational probability of addition of l terms:

$$(a_1 + a_2 + \ldots + a_l) \lll r = a_1 \lll r + a_2 \lll r + \ldots + a_l \lll r. \tag{1}$$

Each of the n-bit words a_i can be seen as a concatenation of two words: r-bit word x_i and $(n - r)$-bit word y_i, that is, $a_i = x_i \| y_i, |x_i| = r, |y_i| = n - r$. Then (1) becomes:

$$(x_1 \| y_1 + \ldots + x_l \| y_l) \lll r = (x_1 \| y_1) \lll r + \ldots (x_l \| y_l) \lll r. \tag{2}$$

The terms $(x_i \| y_i) \lll r$ in the right side of (2), after the rotation on r bits, become $(x_i \| y_i) \lll r = y_i \| x_i$, thus (2) can be rewritten as:

$$(x_1 \| y_1 + \ldots + x_l \| y_l) \lll r = y_1 \| x_1 + \ldots y_l \| x_l. \tag{3}$$

The sum $x_1||y_1 + \ldots + x_l||y_l$ in the left side of (3) can be expressed as $(x_1 + \ldots + x_l + C_{y_1,\ldots,y_l})||(y_1 + \ldots + y_l)$, where $C_{y_1,\ldots,y_k}$ is the carry from the sum $y_1 + \ldots + y_l$. Similarly, the sum in the right side of (3) can be expressed as $(y_1 + \ldots + y_l + C_{x_1,\ldots,x_l})||(x_1 + \ldots + x_l)$. Therefore, after the rotation of the left sum, we obtain:

$$(y_1+\ldots+y_l)\|(x_1+\ldots+x_l+C_{y_1,\ldots,y_l}) = (y_1+\ldots+y_l+C_{x_1,\ldots,x_l})\|(x_1+\ldots+x_l). \tag{4}$$

If we take into account the size of the words x_i and y_i, from (4) we get:

$$\begin{aligned} y_1 + \ldots + y_l &\equiv y_1 + \ldots + y_l + C_{x_1,\ldots,x_l} \pmod{2^{n-r}}, \\ x_1 + \ldots + x_l + C_{y_1,\ldots,y_l} &\equiv x_1 + \ldots x_l \pmod{2^r}, \end{aligned}$$

that is:

$$C_{x_1,\ldots x_l} \equiv 0 \pmod{2^{n-r}},\ C_{y_1,\ldots,y_l} \equiv 0 \pmod{2^r}. \tag{5}$$

As a result, the rotational probability of l-sum addition is equivalent to the probability that (5) will hold for a random values of $x_i, y_i, i = 1, \ldots, l$.

The probability of chained modular additions given in the Lemma is therefore equivalent to the probability of the following system:

$$\begin{aligned} C_{x_1,x_2} &\equiv 0 \pmod{2^{n-r}},\ C_{y_1,y_2} \equiv 0 \pmod{2^r} \\ C_{x_1,x_2,x_3} &\equiv 0 \pmod{2^{n-r}},\ C_{y_1,y_2,y_3} \equiv 0 \pmod{2^r} \\ &\ldots \\ C_{x_1,\ldots x_k} &\equiv 0 \pmod{2^{n-r}},\ C_{y_1,\ldots,y_k} \equiv 0 \pmod{2^r}. \end{aligned}$$

Further we will show that the whole system is equivalent to

$$C_{x_1,\ldots x_k} = 0, \quad C_{y_1,\ldots,y_k} = 0. \tag{6}$$

To do so, we show, by induction on k, that the system of congruences on the left-hand side above, i.e. $C_{x_1,\ldots,x_i} \equiv 0 \pmod{2^{n-r}}$ for all $2 \le i \le k$ is equivalent to the equation $C_{x_1,\ldots,x_k} = 0$. The same sequence of reasoning applies to the right hand side of the above system to show that $C_{y_1,\ldots,y_i} \equiv 0 \pmod{2^r}$ for all $2 \le i \le k$ is equivalent to the equation $C_{y_1,\ldots y_k} = 0$.

First we deal with the easier reverse direction of the equivalence. Indeed, if $C_{x_1,\ldots x_k} = \lfloor \frac{x_1+\ldots+x_k}{2^r} \rfloor = 0$ and hence $x_1 + \ldots + x_k < 2^r$, it follows (by positivity of the x_i's) that $x_1 + \ldots x_i < 2^r$ and hence $C_{x_1,\ldots,x_i} = 0$ and also $C_{x_1,\ldots,x_i} \equiv 0 \pmod{2^{n-r}}$ for all $2 \le i \le k$, as required for the reverse direction.

We now prove the forward direction of the equivalence by induction on k. For the induction base case, we take $k = 2$.The congruence $C_{x_1,x_2} \equiv 0 \pmod{2^{n-r}}$ is equivalent to $C_{x_1,x_2} = t \cdot 2^{n-r}$ for some non-negative integer t. As the carry of addition of two words cannot be larger than 1, it means that $C_{x_1,x_2} \in \{0, 1\}$. If the carry is 1, then from $1 = t \cdot 2^{n-r}$ it follows that $t = 1$ and $2^{n-r} = 1$. However, $r < n$ and thus $2^{n-r} > 1$. Therefore, the carry

C_{x_1,x_2} can only equal zero, and thus $C_{x_1,x_2} \equiv 0 \pmod{2^{n-r}}$ is equivalent to $C_{x_1,x_2} = 0$, proving the induction base case $k = 2$.

For the induction step, suppose that for some $k \geq 2$ the congruence system $C_{x_1,\dots,x_i} \equiv 0 \pmod{2^{n-r}}$ for all $2 \leq i \leq k$ implies the equation $C_{x_1,\dots x_k} = 0$. We show that the congruence system $C_{x_1,\dots,x_i} \equiv 0 \pmod{2^{n-r}}$ for all $2 \leq i \leq k+1$ implies the equation $C_{x_1,\dots x_{k+1}} = 0$. Indeed, by the induction hypothesis, we have that the first k congruences of the system imply that $C_{x_1,\dots,x_k} = 0$ and hence $x_1+\ldots+x_k < 2^r$, whereas $x_{k+1} < 2^r$, so $x_1+\ldots x_{k+1} < 2^r+2^r = 2\cdot 2^r$ and thus $C_{x_1,\dots,x_{k+1}} = \lfloor \frac{x_1+\ldots+x_{k+1}}{2^r} \rfloor \in \{0,1\}$. Then, similarly as in the base case above, the congruence $C_{x_1,\dots,x_{k+1}} \equiv 0 \pmod{2^{n-r}}$ and the fact that $r < n$ imply that the value of the carry $C_{x_1,\dots,x_{k+1}}$ must be zero, which completes the proof of the induction step. As a result, we have reduced the whole system to the two equations given in (6).

Finally, let us find the probability that (6) holds, when $x_i, y_i, i = 1\ldots,k$ are random r-bit and $(n-r)$-bit words, respectively. Namely, we are looking at

$$\Pr(x_1+\ldots x_k < 2^r \wedge 0 \leq x_i < 2^r)\cdot \Pr(y_1+\ldots y_k < 2^{n-r} \wedge 0 \leq y_i < 2^{n-r}) \quad (7)$$

Note that

$$\Pr(x_1+\ldots x_k < 2^r \wedge 0 \leq x_i < 2^r) = \sum_{j=0}^{2^r-1} \Pr(x_1+\ldots x_k = j \wedge 0 \leq x_i < 2^r). \quad (8)$$

Furthermore, the terms in the right side of (8) can be evaluated according to the well known combinatorial formula

$$\#\{z_1+\ldots z_k = j \wedge 0 \leq z_i\} = \binom{j+k-1}{j}, \quad (9)$$

Note that in (9), the condition $0 \leq z_i$ can be replaced with $0 \leq z_i < t$ when $t > j$, as the number of tuples does not increase when $z_i \geq t$ (the sum is always larger than j). Therefore

$$\Pr(z_1+\ldots z_k = j \wedge 0 \leq z_i < t \wedge t > j) = \binom{j+k-1}{j} t^{-k} \quad (10)$$

and (7) can be expressed as:

$$\Pr(x_1+\ldots x_k < 2^r \wedge 0 \leq x_i < 2^r)\cdot \Pr(y_1+\ldots y_k < 2^{n-r} \wedge 0 \leq y_i < 2^{n-r}) =$$
$$= \sum_{j=0}^{2^r-1} \binom{j+k-1}{j} 2^{-rk} \cdot \sum_{j=0}^{2^{n-r}-1} \binom{j+k-1}{j} 2^{-(n-r)k} =$$
$$= \frac{1}{2^{nk}} \sum_{j=0}^{2^r-1} \binom{j+k-1}{j} \cdot \sum_{j=0}^{2^{n-r}-1} \binom{j+k-1}{j}$$

Finally, we use the binomial coefficient formula (for $m, n \in \mathbb{N}$)

$$\sum_{j=0}^{m} \binom{n+j}{j} = \binom{n+m+1}{m}.$$

and we conclude the proof

$$\begin{aligned}
&\Pr([(a_1 + a_2) \lll r = a_1 \lll r + a_2 \lll r] \wedge \\
&\quad \wedge [(a_1 + a_2 + a_3) \lll r = a_1 \lll r + a_2 \lll r + a_3 \lll r] \wedge \\
&\quad \wedge \ldots \\
&\quad \wedge [(a_1 + \ldots + a_k) \lll r = a_1 \lll r + \ldots + a_k \lll r]) = \\
&\quad \frac{1}{2^{nk}} \sum_{j=0}^{2^r-1} \binom{j+k-1}{j} \cdot \sum_{j=0}^{2^{n-r}-1} \binom{j+k-1}{j} = \\
&\quad = \frac{1}{2^{nk}} \binom{k+2^r-1}{2^r-1} \binom{k+2^{n-r}-1}{2^{n-r}-1}.
\end{aligned}$$

□

From the above lemma, we obtain the following important result, which forms the basis for our analysis.

Fact 1 *Chained modular additions do NOT form a Markov chain with respect to rotational differences. Rotational probabilities of chained modular additions cannot be computed as product of probabilities of the individual modular additions.*

We used `GNU Multiple Precision Floating-Point Reliably` library (GNU MPFR) to compute rotational probabilities of chained modular additions according to the results of Lemma 2. Probabilities for 64-bit words, rotation parameters of 1 and 2 and precision of 10000 digits are given in Table 1. For instance, when the rotation parameter is 1 and there are 25 chained modular additions, probability that outputs of these 25 additions are rotational is $2^{-109.6}$. This is to be compared to the claim of Theorem 2, which predicts that probability is $2^{-1.415 \cdot 25} \approx 2^{-35.4}$. We note that we have also computed the rotational probabilities when the rotation amount is greater than 2. The discrepancy is present as well, and it has the tendency to grow – the closer the rotation amount to $n/2$, the larger the discrepancy between the claims of Theorem 2 and of our Lemma 2.

Note that Lemma 2 is used when outputs of all chained additions need to be rotational. This is an important requirement as in ARX, outputs of intermediate modular additions are used as inputs to other operations and are assumed to be rotational. For instance, in Fig. 1, we need d to be rotational, as further it is used in computing the value of w. In contrast, if only the final output of multiple modular additions needs to be rotational, then the rotational probability is computed under different formula (due to space constrains, we omit the formula).

Table 1. The comparison of the rotational probabilities of chained modular additions of 64-bit words given by Theorem 2 of [13], and by our Lemma 2. The probabilities are given $\log_2$.

rotation amount : 1								
# of additions	1	2	3	4	5	6	7	8
Theorem 2 [13]	−1.4	−2.8	−4.2	−5.7	−7.1	−8.5	−9.9	−11.3
Lemma 2	−1.4	−3.6	−6.3	−9.3	−12.7	−16.3	−20.1	−24.1
# of additions	9	10	11	12	13	14	15	16
Theorem 2 [13]	−12.7	−14.1	−15.6	−17.0	−18.4	−19.8	−21.2	−22.6
Lemma 2	−28.3	−32.7	−37.1	−41.7	−46.4	−51.3	−56.2	−61.2
# of additions	17	18	19	20	21	22	23	24
Theorem 2 [13]	−24.1	−25.5	−26.9	−28.3	−29.7	−31.1	−32.5	−34.0
Lemma 2	−66.3	−71.4	−76.7	−82.0	−87.4	−92.9	−98.4	−104.0
# of additions	25	26	27	28	29	30	31	32
Theorem 2 [13]	−35.4	−36.8	−38.2	−39.6	−41.0	−42.4	−43.9	−45.3
Lemma 2	−109.6	−115.3	−121.1	−126.9	−132.8	−138.7	−144.6	−150.6
rotation amount : 2								
# of additions	1	2	3	4	5	6	7	8
Theorem 2 [13]	−1.7	−3.4	−5.0	−6.7	−8.4	−10.1	−11.7	−13.4
Lemma 2	−1.7	−4.3	−7.5	−11.1	−15.1	−19.4	−23.9	−28.7
# of additions	9	10	11	12	13	14	15	16
Theorem 2 [13]	−15.1	−16.8	−18.4	−20.1	−21.8	−23.5	−25.1	−26.8
Lemma 2	−33.6	−38.7	−44.0	−49.4	−54.9	−60.6	−66.3	−72.2
# of additions	17	18	19	20	21	22	23	24
Theorem 2 [13]	−28.5	−30.2	−31.8	−33.5	−35.2	−36.9	−38.5	−40.2
Lemma 2	−78.1	−84.2	−90.3	−96.5	−102.8	−109.1	−115.5	−122.0
# of additions	25	26	27	28	29	30	31	32
Theorem 2 [13]	−41.9	−43.6	−45.3	−46.9	−48.6	−50.3	−52.0	−53.6
Lemma 2	−128.5	−135.1	−141.8	−148.5	−155.3	−162.1	−169.0	−175.9

Above it is assumed that only rotations can break the chain of modular additions. We point out that XORs also break such chains as long as the second term of the XOR is a random value. In practice, for ARX algorithms, the chains are broken by both XORs and rotations. Moreover, due to the possibility of XOR to break chains of modular additions, *rotational probability of ARX primitives highly depends on the way round keys are incorporated into the state, i.e. it is important if round keys are modularly added or XORed to the state*. To illustrate this, let us focus on Fig. 2. We have two ARX primitives, each with two modular additions. The difference is that in the ARX on the left of the figure, the round key is modularly added to the state while in the ARX on the right, the round key is XORed to the state. We can see from the figure that in the left ARX the round key does not break the chain of modular additions, while in the right ARX it does. Hence for the left

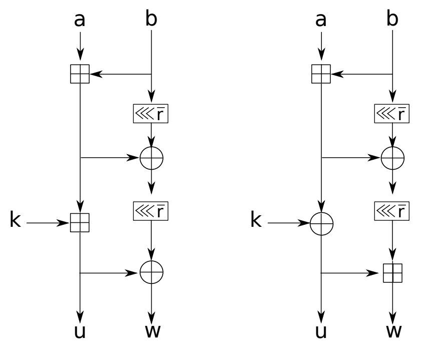

Fig. 2. Two ARX primitives with different incorporations of the subkeys: in the ARX on the left the subkey is modularly added hence does not break the chain of modular additions, while in the ARX on the right the subkey is XORed to the state, thus it breaks the chain.

ARX, we have to use Lemma 2 to compute the rotational probability and thus it is much lower.

The correctness of our analysis has been tested using computer simulations. Our tests have confirmed the predicted rotational probabilities – we refer to Appendix A for the details of the simulation tests.

4 Applications

Lemma 2 suggests that the rotational probability of ARX can be computed more accurately if we take into account not only the number of additions used in ARX but also their positions. We cluster the additions into chains and calculate the rotational probability as follows:

1. Find all chains of modular additions (including the ones that are composed of a single additions) in the ARX primitive.
2. For each chain, compute the rotational probability according to Lemma 2.
3. For the entire primitive, calculate the rotational probability as the product of rotational probabilities of chains.

Consider an example of application of the above algorithm to the case of the ARX primitives given in Fig. 1. Our task is to find rotational probabilities of these two primitives when the rotation parameter is 1 (and the word size is 64 bits). The ARX scheme on the left of Fig. 1 has only one chain of two modular additions. Therefore, according to Table 1, the rotational probability of this chain is $2^{-3.6}$. On the other hand, the ARX on the right of Fig. 1, has two chains composed of a single modular addition and thus the rotational probability of this scheme is $2^{-1.4} \cdot 2^{-1.4} = 2^{-2.8}$.

Now we are ready to revisit the existing rotational attacks on the ARX primitives functions.

4.1 Application to Rotational Cryptanalysis of `BLAKE2`

`BLAKE2` [1] is a hash function, which supports 256 and 512-bit outputs. Further on we analyze the version with 512-bit output only but we note that similar analysis applies to the other version too. The compression function of `BLAKE2` is based on a permutation $P(V, M)$, where V is a state of sixteen 64-bit words v_i, and M is a message input also composed of sixteen words m_i. The function P consists of 12 identical rounds and each round uses 8 applications of the sub-primitive $G_i(a, b, c, d) = G(a, b, c, d, m_{f(i)}, m_{g(i)})$, where $f(i)$ $g(i)$ implement a permutation on a set of 16 message words. The primitive G is first applied columnwise to 4 columns, and then diagonalwise. The column and diagonal steps are defined, respectively, as:

$$G_0(v_0, v_4, v_8, v_{12}), G_1(v_1, v_5, v_9, v_{13}), G_2(v_2, v_6, v_{10}, v_{14}), G_3(v_3, v_7, v_{11}, v_{15}),$$
$$G_4(v_0, v_5, v_{10}, v_{15}), G_5(v_1, v_6, v_{11}, v_{12}), G(v_2, v_7, v_8, v_{13}), G_7(v_3, v_4, v_9, v_{14}).$$

The function $G(a, b, c, d, m_1, m_2)$ itself works as follows:

$$\begin{array}{ll}
1: a \leftarrow a + b + m_i & 5: a \leftarrow a + b + m_j \\
2: d \leftarrow (d \oplus a) \ggg 32 & 6: d \leftarrow (d \oplus a) \ggg 16 \\
3: c \leftarrow c + d & 7: c \leftarrow c + d \\
4: b \leftarrow (b \oplus c) \ggg 24 & 8: b \leftarrow (b \oplus c) \ggg 63
\end{array}$$

Rotational cryptanalysis of the `BLAKE2` permutation [10] uses the rotational parameter equal to 1 (in order to increase the probability) and thus rotational probability of addition is around $2^{-1.4}$. Further, it is noted that the function G has 6 modular additions thus the expected rotational probability of G is $2^{-1.4 \cdot 6} = 2^{-8.4}$. The authors also note that the experimental results show that rotational probability is slightly lower or around $2^{-9.1}$. They took this as rotational probability of one application of G and because the whole permutation has 12 rounds, each with 8 calls to G, they conclude that rotational probability of the permutation used in the compression function of `BLAKE2` is $2^{-9.1 \cdot 12 \cdot 8} = 2^{-873.6}$. Since this permutation works for 1024 bits, a rotational distinguisher is claimed for the full 12-round permutation.

Our rigorous analysis demonstrates that the actual probability would be far lower due to chaining of modular additions, so the conclusion mentioned above is incorrect. Without the loss of generality, we set all the message words to 0, as this yields the rotational pair of messages delivering the highest rotational probability. Then identify all chained modular additions. Fig. 3 shows a round of the permutation, where one can see exactly 8 chains of 4 modular additions each. We can assume the non-chaining inputs of modular additions are independent since they always go through rotations. Therefore, Lemma 2 can be applied. Note that the 8 chains of modular additions continue through the next rounds, totaling $4R$ additions in each chain over R rounds. Consequently, a 7-round permutation (with 8 chains of 28 modular additions each) has rotational probability equal to $(2^{-126.9})^8 = 2^{-1015.2}$ when the rotation amount equal to 1 (see Table 1). Taking more rounds would result in the

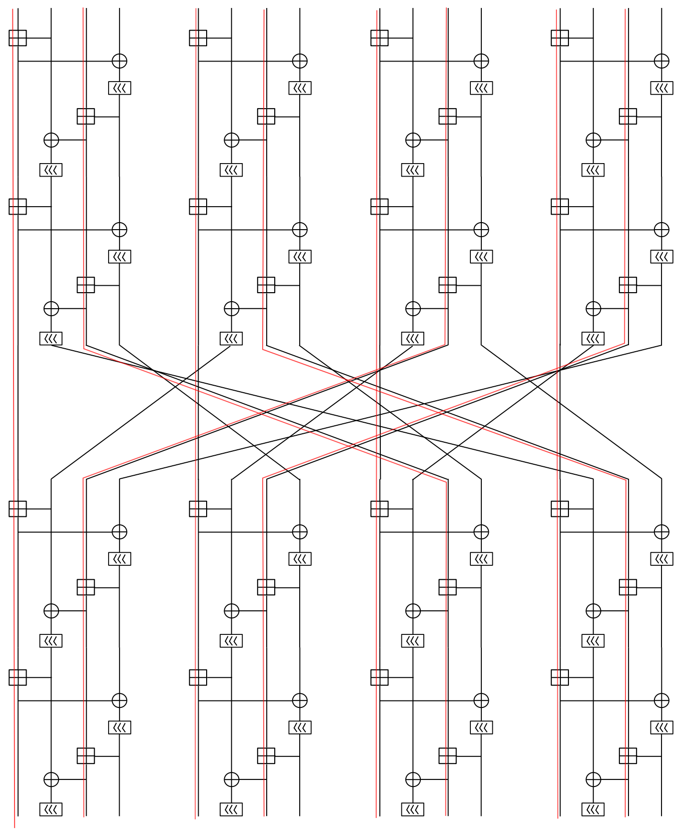

Fig. 3. The 8 chains (denoted in red) of 4 modular additions in one round of the permutation of BLAKE2.

rotational probability smaller than 2^{-1024}. Indeed, Table 1 gives the probability of $2^{-132.8}$ for 29 chained modular additions smaller or equal to $2^{-1062.4}$. Hence, a rotational distinguisher for the permutation of BLAKE2 works for up to 7 rounds only, which is smaller than 12 rounds claimed in [10].

4.2 Application to Rotational Cryptanalysis of Skein

Skein [8] is a hash function proposed for the NIST SHA-3 competition, which reached the final round of the competition. At each round the authors proposed some tweaks to the previous version. Here we consider two such versions Skein v1 [6] and Skein v2 [7] and refer to them as Skein, as the attacks [13, 15] target them both. In order to stop rotational attacks, the designers changed the key schedule in v3.

We consider the version of Skein with a 512-bit internal state, which we denote by Skein-512. The same analysis applies to other versions. The compression function of Skein-512 is based on the block cipher *Threefish*, which is a 72-round ARX scheme with 512-bit state seen as eight 64-bit words. Each round applies 4 parallel MIX functions (Fig. 4), and subkeys (message words) are added every 4 rounds. Subkeys are a bitwise linear function of the master key, the tweak value (not to be confused with the submission tweak), and a round counter.

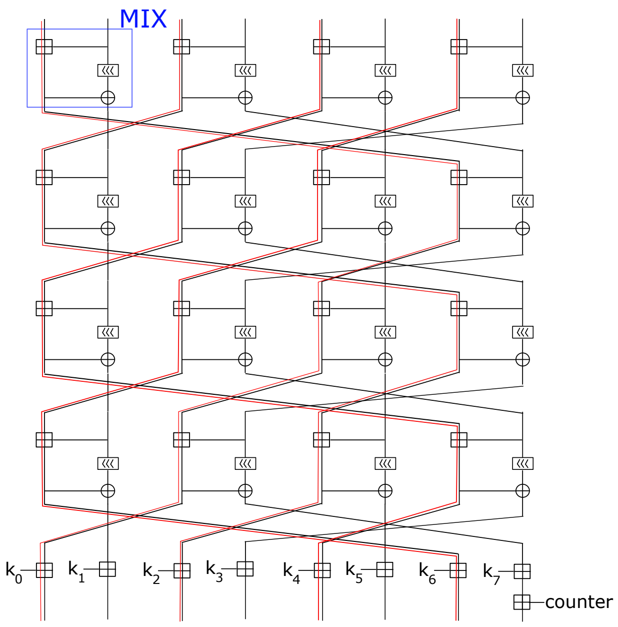

Fig. 4. Four rounds of *Threefish* followed by a subkey addition. In total, *Threefish* uses 18 such four rounds. The rounds use different rotation amounts.

`Skein` (and the underlying *Threefish*) was used as a testbed for rotational cryptanalysis [13] due to the rotation-invariant constant `0x555...555` used in the key schedule and a low-weight counter. By setting the rotational amount to 2, 42 rounds were attacked in [13]. Then the attack was combined with the rebound method and extended to 55 rounds in [15]. To deal with the round counter, the authors had to drop the requirement of having rotational property preserved after each round. Instead, using so called rotational errors and corrections, they introduced a disturbance in the internal state and later corrected it in the manner similar to the local collision concept.

The authors also noticed the difference between the theoretical rotational probabilities (Lemma 1) and experimental values. To cope with this problem, they used the experimental value for the 2-round span where the rotational property is required, and a separate value for the local collision part around the subkey injection. Their experiments are well matched with Table 1 and thus are not questioned.

We do not attempt to do rigorous analysis of the corrections method, as it would involve a much more tedious process of taking all constraints and counter properties into account. Instead, we show that a simplified version of the permutation, where all subkeys and counters are set to 0, has far lower rotational probability than expected by Theorem 2.

It is easy to see 4 parallel addition chains on Fig. 4, which cover state words $S[0], S[2], S[4], S[6]$, with one addition per chain per round. We note that the inputs to these additions coming from the other state words undergo rotations and thus can be considered independent. Therefore, for R rounds of *Threefish* we get 4 chains with R modular additions each. Since there are no constants, we can set the rotation amount to 1 as the most beneficial for the attacker. Table 1 clearly implies that a chain of length 28 has rotational probability smaller or equal to $(2^{-126.9})$. Therefore, 4 chains over 28 rounds yield the rotational probability around 2^{-508}. Setting the rotation parameter to 2 (as in the previous cryptanalyses of `Skein`) would reduce the number of attacked rounds to 24 (as $(2^{-122.0})^4 = 2^{-488}$). For comparison, [15] claims 42-round rotational distinguisher on `Skein` (with all subkeys and constants included), but with the use of rotational corrections. We cannot disprove these results as our formulas do not apply to the case when rotational corrections are used.

5 Conclusion

We have shown that the rotational probability of ARX depends not only on the number of modular additions, but also on how they are connected. The rotational probability of a chain of modular additions cannot be computed as a product of probabilities of the individual additions. This is because the Markov cipher assumption, used implicitly for computing the probability in such a way, does not hold. Therefore, the chain of additions cannot be a Markov chain with respect to rotational analysis, and thus its probability is lower and is defined by Lemma 2. Our analysis also suggests that the way the subkeys are incorporated into the cipher can influence the rotational probability not only because modular added subkeys simply increase the total number of additions, but because XORed subkeys can break the addition chains and thus can increase the probability.

We have investigated the application of rotational cryptanalysis only to ARX, but we note that our methodology can be used for analysis of rotational attacks on other primitives as well. For instance, the Markov cipher assumption used in the recent rotational attacks on Keccak [19] is valid as in each round of Keccak, there is a very strong diffusion, and, moreover, each of the 25 words goes through a rotation. Therefore the rotational probability in the attacks on Keccak [19] is correct. On the other hand, the assumption used the rotational analysis of NORX [12] is clearly not valid (see Appendix B) as the diffusion in the rounds of NORX does not introduce sufficient entropy at the inputs of the non-linear operations. Hence, the probability of the rotational analysis of NORX [12] is miscalculated[3].

To summarize, the rotational probability of a cryptographic primitive not necessarily equals to the product of the rotational probabilities of the individual operations used in the primitive. Such shortcut in the estimation of probability gives neither upper nor lower bound on the actual probability. The estimation can be used only after confirming that the Markov assumption applies to the primitive. Otherwise, the rotational probability must be computed ad-hoc.

Acknowledgements. The authors would like to thank the anonymous reviewers of FSE'15 for very helpful comments and suggestions. Ivica Nikolić is supported by the Singapore National Research Foundation Fellowship 2012 (NRF-NRFF2012-06). Josef Pieprzyk was supported by Australian Research Council grant DP0987734.

References

1. J. Aumasson, S. Neves, Z. Wilcox-O'Hearn, and C. Winnerlein. BLAKE2: simpler, smaller, fast as MD5. In M. J. J. Jr., M. E. Locasto, P. Mohassel, and R. Safavi-Naini, editors, *Applied Cryptography and Network Security - 11th International Conference, ACNS 2013, Banff, AB, Canada, June 25-28, 2013. Proceedings*, volume 7954 of *Lecture Notes in Computer Science*, pages 119–135. Springer, 2013.
2. E. Biham and A. Shamir. Differential cryptanalysis of DES-like cryptosystems. *J. Cryptology*, 4(1):3–72, 1991.
3. E. Bresson, A. Canteaut, B. Chevallier-Mames, C. Clavier, T. Fuhr, A. Gouget, T. Icart, J.-F. Misarsky, M. Naya-Plasencia, P. Paillier, et al. Shabal, a submission to NISTs cryptographic hash algorithm competition. *Submission to NIST*, 2008.
4. C. D. Cannière and C. Rechberger. Finding SHA-1 characteristics: General results and applications. In X. Lai and K. Chen, editors, *Advances in Cryptology - ASIACRYPT 2006, 12th International Conference on the Theory and Application of Cryptology and Information Security, Shanghai, China, December 3-7, 2006, Proceedings*, volume 4284 of *Lecture Notes in Computer Science*, pages 1–20. Springer, 2006.
5. M. Daum. *Cryptanalysis of Hash Functions of the MD4-Family.* PhD thesis, Ruhr-Universität Bochum, May 2005.
6. N. Ferguson, S. Lucks, B. Schneier, D. Whiting, M. Bellare, T. Kohno, J. Callas, and J. Walker. The Skein hash function family. Submission to NIST (Round 1), 2008.
7. N. Ferguson, S. Lucks, B. Schneier, D. Whiting, M. Bellare, T. Kohno, J. Callas, and J. Walker. The Skein hash function family. Submission to NIST (Round 2), 2009.
8. N. Ferguson, S. Lucks, B. Schneier, D. Whiting, M. Bellare, T. Kohno, J. Callas, and J. Walker. The Skein hash function family. 2010.

[3] We have confirmed this experimentally as well. The correct probability is lower than the one predicted in [12]. Therefore, as the paper only gives upper bound on the rotational probability, all the claims of resistance against rotational attacks of NORX still apply.

9. B. Gérard, V. Grosso, M. Naya-Plasencia, and F. Standaert. Block ciphers that are easier to mask: How far can we go? In G. Bertoni and J. Coron, editors, *Cryptographic Hardware and Embedded Systems - CHES 2013 - 15th International Workshop, Santa Barbara, CA, USA, August 20-23, 2013. Proceedings*, volume 8086 of *Lecture Notes in Computer Science*, pages 383–399. Springer, 2013.
10. J. Guo, P. Karpman, I. Nikolić, L. Wang, and S. Wu. Analysis of BLAKE2. In J. Benaloh, editor, *Topics in Cryptology - CT-RSA 2014 - The Cryptographer's Track at the RSA Conference 2014, San Francisco, CA, USA, February 25-28, 2014. Proceedings*, volume 8366 of *Lecture Notes in Computer Science*, pages 402–423. Springer, 2014.
11. J. Guo, T. Peyrin, A. Poschmann, and M. J. B. Robshaw. The LED block cipher. In B. Preneel and T. Takagi, editors, *Cryptographic Hardware and Embedded Systems - CHES 2011 - 13th International Workshop, Nara, Japan, September 28 - October 1, 2011. Proceedings*, volume 6917 of *Lecture Notes in Computer Science*, pages 326–341. Springer, 2011.
12. P. Jovanovic, S. Neves, and J. Aumasson. Analysis of NORX: Investigating differential and rotational properties. *Latincrypt*, 2014. To appear.
13. D. Khovratovich and I. Nikolić. Rotational cryptanalysis of ARX. In S. Hong and T. Iwata, editors, *Fast Software Encryption, 17th International Workshop, FSE 2010, Seoul, Korea, February 7-10, 2010, Revised Selected Papers*, volume 6147 of *Lecture Notes in Computer Science*, pages 333–346. Springer, 2010.
14. D. Khovratovich, I. Nikolić, and C. Rechberger. Rotational rebound attacks on reduced Skein. In M. Abe, editor, *Advances in Cryptology - ASIACRYPT 2010 - 16th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 5-9, 2010. Proceedings*, volume 6477 of *Lecture Notes in Computer Science*, pages 1–19. Springer, 2010.
15. D. Khovratovich, I. Nikolić, and C. Rechberger. Rotational rebound attacks on reduced Skein. *J. Cryptology*, 27(3):452–479, 2014.
16. A. Kircanski, Y. Shen, G. Wang, and A. M. Youssef. Boomerang and slide-rotational analysis of the SM3 hash function. In L. R. Knudsen and H. Wu, editors, *Selected Areas in Cryptography, 19th International Conference, SAC 2012, Windsor, ON, Canada, August 15-16, 2012, Revised Selected Papers*, volume 7707 of *Lecture Notes in Computer Science*, pages 304–320. Springer, 2012.
17. X. Lai and J. L. Massey. Markov ciphers and differentail cryptanalysis. In D. W. Davies, editor, *Advances in Cryptology - EUROCRYPT '91, Workshop on the Theory and Application of of Cryptographic Techniques, Brighton, UK, April 8-11, 1991, Proceedings*, volume 547 of *Lecture Notes in Computer Science*, pages 17–38. Springer, 1991.
18. G. Leurent. Analysis of differential attacks in ARX constructions. In X. Wang and K. Sako, editors, *Advances in Cryptology - ASIACRYPT 2012 - 18th International Conference on the Theory and Application of Cryptology and Information Security, Beijing, China, December 2-6, 2012. Proceedings*, volume 7658 of *Lecture Notes in Computer Science*, pages 226–243. Springer, 2012.
19. P. Morawiecki, J. Pieprzyk, and M. Srebrny. Rotational cryptanalysis of round-reduced Keccak. In S. Moriai, editor, *Fast Software Encryption - 20th International Workshop, FSE 2013, Singapore, March 11-13, 2013. Revised Selected Papers*, volume 8424 of *Lecture Notes in Computer Science*, pages 241–262. Springer, 2013.
20. I. Nikolić, J. Pieprzyk, P. Sokołowski, and R. Steinfeld. Rotational cryptanalysis of (modified) versions of BMW and SIMD, 2010.
21. M. Stevens. New collision attacks on SHA-1 based on optimal joint local-collision analysis. In T. Johansson and P. Q. Nguyen, editors, *Advances in Cryptology - EUROCRYPT 2013, 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece, May 26-30, 2013. Proceedings*, volume 7881 of *Lecture Notes in Computer Science*, pages 245–261. Springer, 2013.
22. G. Van Assche. A rotational distinguisher on Shabals keyed permutation and its impact on the security proofs. *NIST mailing list*, 2010.

23. X. Wang, Y. L. Yin, and H. Yu. Finding collisions in the full SHA-1. In V. Shoup, editor, *Advances in Cryptology - CRYPTO 2005: 25th Annual International Cryptology Conference, Santa Barbara, California, USA, August 14-18, 2005, Proceedings*, volume 3621 of *Lecture Notes in Computer Science*, pages 17–36. Springer, 2005.
24. X. Wang and H. Yu. How to break MD5 and other hash functions. In R. Cramer, editor, *Advances in Cryptology - EUROCRYPT 2005, 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22-26, 2005, Proceedings*, volume 3494 of *Lecture Notes in Computer Science*, pages 19–35. Springer, 2005.

A Computer Simulations of Rotational Attacks

We tested the rotational probabilities predicted by our Lemma 2 and the new algorithm on the case of three ARX primitives with round functions given in Fig. 5. The first ARX has no chains of modular additions, i.e. all chains are

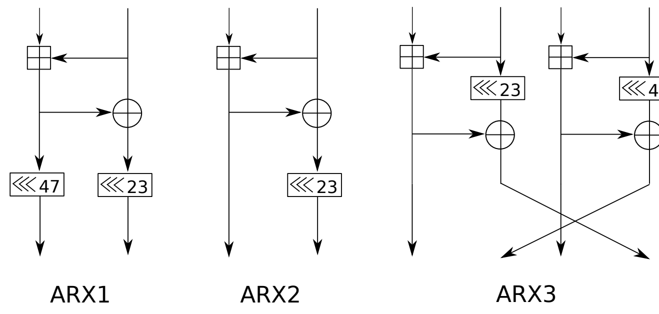

Fig. 5. The round functions of three ARX primitives used for testing the experimental rotational probabilities.

broken by rotations and/or XORs. The second ARX has one chain of modular additions through the rounds. The third ARX has has two chains and is in fact based on the cipher *Threefish*-256 (but without subkey additions and counters and with modified rotations). For each of the three ARX primitives, we obtained experimentally the rotational probabilities by taking 2^{32} random rotational pairs of inputs (with rotation amounts 1 and 2), and counting how many output pairs were also rotational. The outcomes of our computer simulations are summarized in Tbl. 2. From the table we can see that the experimental probabilities matched our predicted probabilities.

B On the Rotational Analysis of NORX

One round of NORX (similar to that of BLAKE) is composed of 8 applications of the following update function G:

Table 2. The comparison of the theoretical and experimental rotational probabilities of the tree ARX primitives from Fig. 5. The probabilities are given $\log_2$. **np** stands for not performed – when the theoretical probabilities were below 2^{-32} we did not run the experiments.

rotation amount : 1						
# of rounds	3	4	5	6	7	8
ARX1 theoretical	−4.25	−5.66	−7.08	−8.49	−9.91	−11.32
ARX1 experimental	−4.25	−5.66	−7.08	−8.49	−9.91	−11.32
ARX2 theoretical	−6.26	−9.32	−12.68	−16.30	−20.13	−24.15
ARX2 experimental	−6.26	−9.32	−12.69	−16.30	−20.15	−24.15
ARX3 theoretical	−12.53	−18.64	−25.37	−32.60	−40.26	−48.29
ARX3 experimental	−12.52	−18.61	−25.16	np	np	np
rotation amount : 2						
# of rounds	1	2	3	4	5	6
ARX1 theoretical	−1.68	−3.35	−5.03	−6.70	−8.38	−10.06
ARX1 experimental	−1.68	−3.35	−5.03	−6.71	−8.39	−10.07
ARX2 theoretical	−1.68	−4.26	−7.46	−11.10	−15.10	−19.39
ARX2 experimental	−1.68	−4.26	−7.46	−11.10	−15.11	−19.38
ARX3 theoretical	−3.36	−8.53	−14.91	−22.20	−30.20	−38.78
ARX3 experimental	−3.36	−8.53	−14.91	−22.25	−29.68	np

$$\begin{array}{ll}
1 : a \leftarrow (a \oplus b) \oplus ((a \wedge b) \ll 1) & 5 : a \leftarrow (a \oplus b) \oplus ((a \wedge b) \ll 1) \\
2 : d \leftarrow (a \oplus d) \ggg r_0 & 6 : d \leftarrow (a \oplus d) \ggg r_0 \\
3 : c \leftarrow (c \oplus d) \oplus ((c \wedge d) \ll 1) & 7 : c \leftarrow (c \oplus d) \oplus ((c \wedge d) \ll 1) \\
4 : b \leftarrow (b \oplus c) \ggg r_1 & 8 : b \leftarrow (b \oplus c) \ggg r_1
\end{array}$$

Among all the transformation in the round, obviously only

$$H(x, y) = (x \wedge y) \ll 1$$

does not preserve the rotational property with probability 1. In fact, in [12] it has been proven that the rotational probability of this non-linear function is 9/16. Then, similar to the previous analysis of ARX, the authors counted only the number of applications of $H(x, y)$ per round of NORX (which is 32, as there are 8 applications of G and each of them applies $H(x, y)$ 4 times), and computed the rotational probabilities of NORX reduced(or extended) to R rounds as $(9/16)^{32 \cdot R}$.

This probability estimation is correct only under the Markov cipher assumption. However, this assumption does not hold in the case of NORX. The reason for this is as follows. As in [12], let us assume that the rotation is to the right (and the rotation amount is arbitrary, i.e. $0 < r < 64$). A given pair of words (a, b) composes a rotational pair for H, i.e.

$$H(a, b) \ggg r = H(a \ggg r, b \ggg r)$$

iff the two most significant bits of a and b, a_{63} and b_{63}, and the two r-least significant bits a_{r-1}, b_{r-1} satisfy $a_{63} \wedge b_{63} = 0, a_{r-1} \wedge b_{r-1} = 0$. This has been proven in [12], and actually gives a hint where the probability 9/16 comes from – obviously the bit equation $z \wedge t = 0$ holds with probability 3/4 when z and t are random. Therefore, when $a_{63}, b_{63}, a_{r-1}, b_{r-1}$ are *random*, then $H(a,b) \ggg r = H(a \ggg r, b \ggg r)$ with probability 9/16.

But what happens after a, b compose a rotational pair? Let us only focus on the value of the most significant bits a_{63} and b_{63}. From the rotational requirement of H it follows that only the values $(0,0), (0,1), (1,0)$ for (a_{63}, b_{63}) will satisfy $a_{63} \wedge b_{63} = 0$. Furthermore, the value of a is updated in G according to:

$$\overline{a} = (a \oplus b) \oplus ((a \wedge b) \ll 1).$$

Let us focus only on the value of $\overline{a_{63}}$, i.e.:

$$\overline{a_{63}} = a_{63} \oplus b_{63} \oplus (a_{62} \wedge b_{62})$$

The term $a_{63} \oplus b_{63}$ as we have seen from above (from the rotational requirement, we take only the values of three possible $(0,0), (0,1), (1,0)$) is 0 with probability 1/3 and 1 with probability 2/3. On the other hand, the term $a_{62} \wedge b_{62}$ takes the value of 0 with probability 3/4 and 1 with 1/4 (we assume a_{62}, b_{62} are random bits when $r \neq 63$). Therefore, the value of $\overline{a_{63}}$ is 0 with probability 5/12 and 1 with 7/12, i.e. *it is biased towards 1*. Hence in the next update of a (where $\overline{a}$ will be used), the value of this bit is not completely random, and it will contribute towards reducing the rotational probability of H (because the rotational probability is lower when the value of a_{63} one). Instead of the predicted 9/16 the probability will reduce to $3/4 \cdot (5/12 + 5/12 + 7/12) \cdot 1/2 = 51/96$.

The same observation applies not only to the word a, but also to c. Hence, one G actually has two chained non-linear operations with respect to rotational analysis (they correspond to the chained modular additions in the case of ARX).