Rotational Cryptanalysis of ARX

Dmitry Khovratovich and Ivica Nikolić

University of Luxembourg

dmitry.khovratovich@uni.lu, ivica.nikolic@uni.lu

Abstract. In this paper we analyze the security of systems based on modular additions, rotations, and XORs (ARX systems). We provide both theoretical support for their security and practical cryptanalysis of real ARX primitives. We use a technique called *rotational cryptanalysis*, that is universal for the ARX systems and is quite efficient. We illustrate the method with the best known attack on reduced versions of the block cipher Threefish (the core of Skein). Additionally, we prove that ARX with constants are functionally complete, i.e. any function can be realized with these operations.

Keywords: ARX, cryptanalysis, rotational cryptanalysis.

1 Introduction

A huge number of symmetric primitives using modular additions, bitwise XORs, and intraword rotations have appeared in the last 20 years. The most famous are the hash functions from MD-family (MD4, MD5) and their descendants SHA-x.

While modular addition is often approximated with XOR, for random inputs these operations are quite different. Addition provides diffusion and nonlinearity, while XOR does not. Although the diffusion is relatively slow, it is compensated by a low price of addition in both software and hardware, so primitives with relatively high number of additions (tens per byte) are still fast. The intraword rotation removes disbalance between left and right bits (introduced by the addition) and speeds up the diffusion.

Many recently design primitives use only XOR, addition, and rotation so they are grouped into a single family *ARX* (Addition-Rotation-XOR). Among them are SHA-3 competitors Skein [14], BLAKE [3], CubeHash [5], and the stream ciphers Salsa20 [4]. It is a common belief that the mixture of these operations gives a good primitive, if the number of rounds is sufficient. However, to the best of our knowledge, there is no formal theory whether all three operations are necessary and sufficient for this task.

We investigate this problem from different points of view. Certainly, the most interesting question is how secure the ARX(-C) systems are. So far, the most of the analysis of ARX systems was made in the framework of differential cryptanalysis [23, 9, 8], with a few exceptions where symmetric states were considered [1]. We investigate the security of ARX systems with a technique that

we call *rotational cryptanalysis*, where we study the propagation of *a rotational pair* $(X, X \ggg_r)$ throughout the primitive[1]. Operations XOR and rotation both preserve the rotational pair with probability 1, while the modular addition does it with probability up to $\frac{3}{8}$, depending only on the rotation amount r. Therefore, a rotational pair of inputs is converted to a rotational pair of outputs with a probability depending only on the number of additions in the scheme. Hence, we get a universal upper bound on the security of the ARX primitives.

The use of constants, which may not form a rotational pair, does not restrain our analysis, but makes it more sophisticated. We show how reduced versions of Threefish (the block cipher used in the hash function Skein) can be analyzed with rotational cryptanalysis, and our results basically are the best known cryptanalysis of this design.

We also prove that the ARX operations with a constant are functionally complete in the set of functions over Z_2^n. In other words, any function can be realized with modular addition, XOR, rotation, and a single constant (*ARX-C*). We also show that the *AR systems*, that do not use XOR, are theoretically equivalent to ARX systems. However, we prove that they are less secure with the same number of operations, because of the linear mod $2^n - 1$ approximation. It is also easy to prove that omitting addition or rotation is devastating, and such systems (XR and AX) can always be broken.

This paper is structured as follows. We survey related works in Section 2. We describe rotational cryptanalysis in Section 3 and then apply it to Threefish (Section 4). Then we prove the completeness of AR and ARX operations in Section 5. We conclude our paper with generic cryptanalysis of the AR systems (Section 6).

2 Related Work

It is hard to survey all the research done on ARX systems, so we point out only the most important. Relation between modular addition and XOR was studied in the PhD thesis of Daum [12]. Dedicated approaches were applied in many works, among them on MD5, SHA-1 [23, 9].

The impact of rotations was independently studied in the cryptanalysis of block ciphers. In the pioneering work on related keys by Biham [6] a rotational pair of keys was considered. This approach was extended by Kelsey et al. in several related-key attacks on block ciphers [15]. In these attacks the adversary tries to find pairs of plaintexts of form $(P, F(P))$, where F is the round transformation, so this is not a pure rotational cryptanalysis.

An AR system RC5P was attacked with mod n cryptanalysis in [16], and the property (1) was introduced in the same paper. The internal states were, however, computed modulo 3, and the computation modulo $2^n - 1$ was only briefly pointed out.

[1] The technique of rotational cryptanalysis has been known and applied before, see Section 2.

A common approach to cryptanalysis of ARX systems is linearization ([10, 9], see also the systematic treatment in [8]), when modular additions are approximated by XOR, and the resulting function is linear. In the differential cryptanalysis there is no need to approximate the whole addition by XOR, it is sufficient to assume that difference propagates in a linear way. The linearization approach works worse if the diffusion is good, so that even in a linear model differentials involve too many active bits. As a result, the probability of the approximation becomes very low even for a single modular addition.

A systematic treatment of AX-systems can be found in the work by Paul and Preneel [20]. It was demonstrated that all such systems can be broken with a low complexity, since one can work with bits from rightmost to leftmost.

Related-key attacks were introduced by Biham [6] and were used in the practical break of WEP [22]. The key relations used in the attacks vary from fixed differences [17] to non-trivial subkey relations [7].

A rotational pair of inputs (though it was not named so) was used in the attack on the compression function of Shabal [18]. However, it was traced only through bitwise operations, and not through additions. Bernstein [4] explicitly prevented from use of rotational pairs in Salsa20 by fixing non-symmetric constants in the input of the permutation. However, he did not provide any complexity or probability estimates for this kind of attack.

The designers of the block cipher SEA [21] described the technique of rotational cryptanalysis in 2006 and defended against it with non-linear key-schedule and pseudo-random constants. A modified version of block cipher Serpent, with key schedule constants removed, is vulnerable to rotation cryptanalysis due to the bit-slice nature of the S-boxes [13]. Also, the rotational pairs were tested for Threefish [19], but this did not result into a full attack.

3 Review of Rotational Cryptanalysis

In this section we describe a generic method for the analysis of ARX systems. The main idea is to consider pair of words where one is the rotation of the other one.

We denote the intraword rotation operations by $\lll_r$ and $\ggg_r$. A rotated variable is then denoted by $\overleftarrow{X}$ and $\overrightarrow{X}$, respectively. Now, let $\overrightarrow{X}$ be the rotation of X by r bits to the right. We call $(X, \overrightarrow{X})$ a *rotational pair* [with a rotation amount r]. It is easy to prove that a rotational pair is preserved by any bitwise transformation, particularly by the bitwise XOR and by any rotation:

$$\overrightarrow{X \oplus Y} = \overrightarrow{x} \oplus \overrightarrow{y}, \qquad \overrightarrow{x} \ggg_{r'} = \overrightarrow{x \ggg_{r'}}.$$

Now consider addition modulo 2^n. The probability that the rotational pair comes out of the addition is given by the following lemma.

Lemma 1 (Daum, [12]). $\mathbf{P}(\overrightarrow{x+y} = \overrightarrow{x} + \overrightarrow{y}) = \frac{1}{4}(1 + 2^{r-n} + 2^{-r} + 2^{-n})$.

For large n and small r we get the following table:

r	p_r	$\log_2(p_r)$
1	0.375	-1.415
2	0.313	-1.676
3	0.281	-1.831

For $r = n/2$ the probability is close to $1/4$. The same holds for rotations to the left.

Now consider an arbitrary scheme $\mathcal{S}$ with additions, rotations, and XORs over n-bit words. Then the following theorem holds under independency assumptions.

Theorem 1. *Let q be the number of addition operations in an ARX scheme $\mathcal{S}$. Let $\overrightarrow{I}$ be the input I of scheme $\mathcal{S}$ rotated to the right by r bits. Then $\mathcal{S}(\overrightarrow{I}) = \overrightarrow{\mathcal{S}(I)}$ with probability $(p_r)^q$.*

Proof. It can be proved by induction on the scheme size.□

We would like to stress that in order the rotational analysis to work, **all** inputs to the ARX scheme should compose rotational pairs.

For a random function $\mathcal{P}$ that maps to Z_2^t the probability that $\mathcal{P}(\overrightarrow{I}) = \overrightarrow{\mathcal{P}(I)}$ for random I is 2^{-t}. Therefore, we can detect nonrandomness if a function can be implemented with q additions, and $(p_r)^q > 2^{-t}$. For example, when $r = 1$ we get that any ARX scheme that can be implemented with less than $t/1.415$ additions, is vulnerable to rotational cryptanalysis.

Dealing with constants

In contrast to random schemes, iterative schemes with identical rounds suffer from slide attacks and their modifications. The use of different round constants is typical countermeasure. As a result, many designs explicitly use constants, and we have to adapt our method to work with them.

Let us introduce the notion a *rotation error* $\overrightarrow{E}$. In the further text r is a fixed rotation amount. We define

$$E(X, Y) = \overrightarrow{X} \oplus Y.$$

Clearly, $E(X, \overrightarrow{X}) = 0$.

An addition of a constant may generate a rotation error (the exact probability depends on r, the constant value, and which type of addition is used — modular or XOR). On the other hand, a modular addition of variables also may generate an error, and with some probability these errors compensate each other:

$$E(X + Y + Z + C, \overrightarrow{X} + \overrightarrow{Y} + \overrightarrow{Z} + C) = 0.$$

The probability is higher if the constant has low Hamming weight and its ones are concentrated close to the positions where addition errors appear. It may

also happen that a constant is added by XOR and is invariant of the rotation: $C = \overrightarrow{C}$. Then the rotation property passes the addition of a constant for free.

The subkey indices in Threefish are examples of low-weight constants. However, they are not compensated by a single addition, only by two previous additions, which leads to an error in the adjacent key addition. We have to introduce additional corrections in the key to cancel the impact of constants (see Section 4 for more details).

4 Rotational Cryptanalysis of Threefish

In this section we attack the block cipher Threefish with rotational cryptanalysis. We demonstrate that a rotational pair of Threefish ciphertexts can be obtained faster than for a random permutation, which provides both a distinguisher and a key recovery attack.

4.1 Specification of Threefish

Threefish is a family of block ciphers underlying the compression function of Skein[14]. Threefish supports three different versions: 1) Threefish-256 — 256-bit block cipher with 256-bit key, 2) Threefish-512 — 512-bit block and key, and 3) Threefish-1024 — 1024-bit block and key. Both the internal state I and the key K consist of N_w ($N_w = 4, 8, 16$ for Threefish-256,-512,-1024, respectively) 64-bit words. The N_w words of the s-th subkey K^s are defined as follows:

$$\begin{aligned} K^s_j &= K_{(s+j) \bmod (N_w+1)}, \quad 0 \le j \le N_w - 4; \\ K^s_{N_w-3} &= K_{(s+N_w-3) \bmod (N_w+1)} + t_{s \bmod 3}; \\ K^s_{N_w-2} &= K_{(s+N_w-2) \bmod (N_w+1)} + t_{(s+1) \bmod 3}; \\ K^s_{N_w-1} &= K_{(s+N_w-1) \bmod (N_w+1)} + s, \end{aligned}$$

where s is a round counter, t_0 and t_1 are tweak words, and

$$t_2 = t_0 + t_1, \quad K_{N_w} = \lfloor 2^{64}/3 \rfloor \oplus \bigoplus_{j=0}^{N_w-1} K_j.$$

Further in our analysis we fix the tweaks $t_0 = t_1 = 0$.

The formal description of internal rounds is as follows. Let N_r be the number of rounds. Then for every $1 \le d \le N_r$

- If $d \bmod 4 = 1$ add a subkey by setting $I_j \leftarrow I_j + K_j^{d/4}$;
- For $0 \le j < N_w/2$ set $(I_{2j}, I_{2j+1}) \leftarrow \mathrm{MIX}((I_{2j}, I_{2j+1}))$;
- Apply the permutation π on the state words.

Rounds	Type	Reference
Threefish-256 (72 rounds)		
24	Related-key differential	[14]
39	Related-key rotational	Sec. 4
Threefish-512 (72 rounds)		
25	Related-key differential	[14]
32	Related-key boomerang	[2]
33	Related-key boomerang	[11]
42	Related-key rotational	Sec. 4
35	Known-related-key distinguisher	[2]
Threefish-1024 (80 rounds)		
26	Related-key differential	[14]
43.5	Related-key rotational	Sec. 4

Table 1. Summary of the attacks on Threefish

In the end a subkey $K^{N_r/4}$ is added. The operation MIX has two inputs x_0, x_1 and produces two outputs y_0, y_1 with the following ARX transformation:

$$y_0 = x_0 + x_1$$
$$y_1 = (x_1 \lll_{R_{(d \bmod 8)+1,j}}) \oplus y_0$$

The exact values of the rotation constants $R_{i,j}$ as well the permutations π (which are different for each version of Threefish) can be found in [14].

The best known analysis [2] of Threefish-512 is 33-round attack in the related-key model, and 35-round attack in the known-related-key model (Tbl. 1). The designers of Threefish have changed the MIX rotation constants in the latest tweak to get better diffusion. We note that our attack is independent of these constants.

4.2 Attacks on Simplified Versions of Threefish

There are two places in the key schedule of Threefish where we encounter constants: 1) K_{N_w} is obtained with a XOR of all key words and the constant $C_5 = \lfloor 2^{64}/3 \rfloor$, and 2) the last subkey word $K^s_{N_w-1}$ has a modular addition of the round counter s. Hence, in addition to the original Threefish, we can obtain three simplified versions by discarding these constant XOR and counter additions. Our attacks are in the related-key scenario, where all the key and plaintext words compose rotational pairs, i.e. if the first key and the plaintext have the values $(k_0, \ldots, k_{N_w}), (p_0, \ldots, p_{N_w-1})$ then the second (related) key and the plaintext have the values $(\overrightarrow{k_0}, \ldots, \overrightarrow{k_{N_w}}), (\overrightarrow{p_0}, \ldots, \overrightarrow{p_{N_w-1}})$.

The simplest version of Threefish is without the XOR of C_5 and the additions of the round counters. We can fix the rotation amount in the rotational pair to 1 in order to get the best probability — $2^{-1.415}$ per addition. A simple MIX has only one addition, hence a round of Threefish-256 has only two additions. The 59-round version of Threefish-256 has $2 \cdot 59 = 118$ additions in the MIX of the rounds and $4 \cdot 15 = 60$ additions of the subkey words, so the probability that a rotational pair of key/plaintext (with a rotation equal to 1) will produce a rotational pair of ciphertexts is $2^{-1.415 \cdot (118+60)} = 2^{-252}$, which is higher than for a random permutation. Every right pair also provides information on leftmost key bits of each key word, so we get a valid key recovery attack with a complexity of about 2^{252} encryptions. The same reasoning is applicable for 59-round distinguishers for Threefish-512 which has a complexity of 2^{504} and to Threefish-1024 and 2^{1008}, because these ciphers differ from Threefish-256 only in the size of the state and the key.

When the XOR of C_5 is present, then the only difference is that we cannot use the rotation amount 1 because $C_5 \lll 1 \neq C_5$, i.e. the constant C_5 is not invariant of rotation 1. Instead we can use rotation 2, and get attacks on 50 rounds. The complexity of the attack on Threefish-256 is $2^{1.67 \cdot (2 \cdot 50 + 4 \cdot 13)} = 2^{253.8}$. For Threefish-512 and Threefish-1024 they are $2^{507.6}$ and $2^{1015.2}$.

For the version of Threefish without the constant C_5 and the round counters, we get much better results if we consider a weak key class, for which it is unlikely to get errors during the modular addition. Let the three leftmost bits of each key word be zero, and consider rotation to the left by one bit. Then the probability that $\overleftarrow{X + K} = \overleftarrow{X} + \overleftarrow{K}$ is equal to $2^{-0.28}$ for a random X, and so the total probability for the full 72-round Threefish-256, the version without C_5 and round counters, is $2^{-1.415 \cdot 2 \cdot 72 - 0.28 \cdot 4 \cdot 18} = 2^{-224}$. The size of the weak key class that we attack is $2^{61 \cdot 4} = 2^{244}$, so we get a valid attack on a very large key class. Analogously, we can attack a weak key class with 2^{488} keys of Threefish-512 with complexity 2^{448}, and Threefish-1024 with a complexity 2^{950} (the complexity is slightly higher because Threefish-1024 has 80 rounds).

4.3 Attacks on the Original Threefish

Let us try to apply rotational analysis to the original version of Threefish. This means we have to deal with the round counters – low weight constants. In order to bypass them we introduce corrections in the key pair. Let K be the first secret key. Then the second key K' is defined as follows:

$$K'_i = \overleftarrow{K_i} \oplus e_i$$

The use of rotational pairs with errors is illustrated in Fig. 1.

We have found experimentally, that the values of the corrections e_i should not be larger than 16 (otherwise they do not cancel the round counters). For Threefish-256 and Threefish-512 it is feasible to find by brute force the exact values for the corrections that cancel the counters with maximal probability. For Threefish-1024 we took the values that were good in Threefish-512.

The corrections forbid to obtain clear formula for the probability of addition of a rotational pair. Hence, we have found these probabilities empirically. We have grouped two rounds with a subkey addition (round – subkey addition – round), and by Monte Carlo method found the probability that a rotational pair of states at the input of these two rounds and a rotational pair of subkeys with corrections will produce a rotational pair of states at the output. Based on these values, we have produced the probabilities of the best round-reduced rotational pairs. The explicit round-by-round values of the probabilities are given in Tbl. 4 in the Appendix. The results are given for the original versions as well as for the versions without the C_5 (except in the case for Threefish-1024 where the probability of the version without C_5 is lower than for the original Threefish-1024).

We can break 39, 42, and 43.5[2] rounds of the original versions of Threefish-256,-512, and -1024 with complexity of $2^{252.4}, 2^{507}, 2^{1014.5}$ encryptions respectively. The attacks procedures follow the same algorithm:

1. Generate a random plaintext P and encrypt it on K;
2. Compute P' and encrypt on K';
3. Check whether $(E_K(P), E_{K'}(P'))$ is a rotational pair.

A rotational pair discloses information about leftmost key bits of every key word. The plaintext P' is computed by the following rule:

$$P'_i = \overleftarrow{P_i} \oplus d_i.$$

The plaintext and the key corrections are defined separately for all the three versions of Threefish in Tbl. 2. The correction values for the version without C_5 are given in Appendix.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Threefish-256																
d_i	3	10	3	15												
e_i	6	10	6	15												
Threefish-512																
d_i	0	6	3	6	3	6	3	6								
e_i	5	6	6	6	6	6	6	6								
Threefish-1024																
d_i	0	6	3	6	3	6	3	6	3	6	3	6	3	6	3	6
e_i	5	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6

Table 2. Corrections in the plaintext pairs (d_i) and the key pairs (e_i) in Threefish.

[2] .5 means without the last subkey addition.

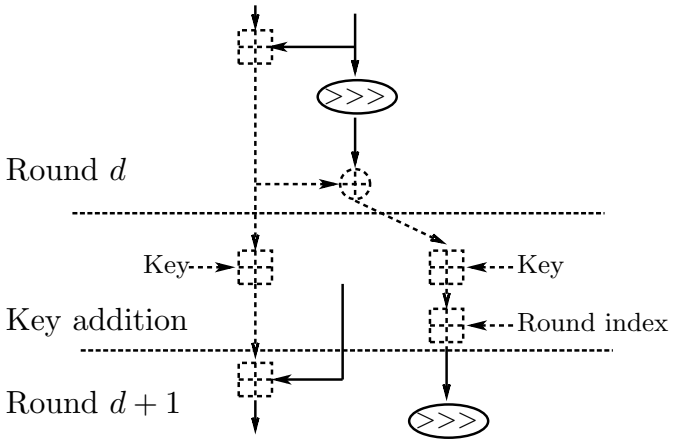

Fig. 1. Rotational errors in the key addition layer of Threefish. Dashed lines contain rotational pairs with errors.

For the versions with counters but without C_5, again we only change the rotation amount to 1, and obtain attacks on 44 and 51.5 rounds of Threefish-256, -512 with a complexity of 2^{252} and $2^{506.7}$ encryptions.

5 Completeness of ARX

In this section we investigate which primitives can be obtained in the ARX framework. We show that any function can be implemented with the ARX operations and a constant 1. Therefore, there is no generic property that holds for the ARX-C systems with probability 1, and any cryptanalysis method fails for a reasonably large system. However, rotational cryptanalysis (Section 3) shows that such a system should be much larger than expected.

Let us now analyze the ARX operations and start with a system of notations. In the further text + always stands for modular addition. Let an integer n be the word length, and denote an $n-bit$ word by W. Denote also by $\mathcal{F}$ the set of all functions to W:

$$\mathcal{F} = \{f : W^m \to W \,|\, m \in \mathbb{N}\}.$$

We say that a set $\mathcal{Q}$ is a basis in $\mathcal{F}$ if any function from $\mathcal{F}$ can be realized by a scheme with elements from $\mathcal{Q}$.

Theorem 2. *The set of functions* $\{+, \oplus, \ggg_1, 1\}$ *is a basis in* $\mathcal{F}$.

Proof. We show how to realize an arbitrary function $f \in \mathcal{F}$.

1. Realize $s_i(X) = 00\cdots0x_i$, where $X = x_1x_2\cdots x_n$. First we rotate X by $n-i-1$ bits to the right using $n-i-1$ simple rotations. Then we add it

Cipher	Round index	Constant $\lfloor 2^{64}/3 \rfloor$	Rounds	Complexity
Threefish-256	no	no	59	2^{252}
	no	yes	50	$2^{253.8}$
	yes	no	44	$2^{251.4}$
	yes	**yes**	39	$2^{254.1}$
(weak key of 2^{244})	no	no	72	2^{224}
Threefish-512	no	no	59	2^{504}
	no	yes	50	$2^{507.6}$
	yes	no	51.5	$2^{505.5}$
	yes	**yes**	42	2^{507}
(weak key of 2^{488})	no	no	72	2^{448}
Threefish-1024	no	no	59	2^{1008}
	no	yes	50	$2^{1015.2}$
	yes	**yes**	43.5	$2^{1014.5}$
(weak key of 2^{976})	no	no	80	2^{950}

Table 3. Attack complexities for different versions of Threefish; the weak key class on full-round Threefish.

to itself $n-1$ times thus getting $x_i 00\cdots 0$. Finally, we rotate the result to $00\cdots 0x_i$.

2. Realize all functions $M_k(X,Y) = 00\cdots 0(x_k y_k)$.
3. Realize all functions $j_C(X) = \begin{cases} 00\cdots 01, \text{ if } X = C; \\ 0, \text{ else.} \end{cases}$ as follows. Let $C = (c_1, c_2, \ldots, c_{nm})$. Then $j_C(X) = \prod(x_i \oplus c_i \oplus 1)$, which can be computed with functions $\{M_k(X,Y)\}$ and the constant 1.
4. Realize all functions $J_{C_1,C_2}(X) = \begin{cases} C_2, \text{ if } X = C_1; \\ 0, \text{ else.} \end{cases}$ as $C_2 \cdot j_{C_1}(X)$.
5. Realize f as $\bigoplus_{X \in W^m} J_{X,f(X)}$.

□

Theorem 3. *The set of functions $\{+, \ggg_1, 1\}$ is a basis in $\mathcal{F}$.*

Proof. We only have to prove that $\oplus$ can be realized with AR operations. Indeed, we realize $s'_i(X) = x_i 00\cdots 0$ in the same way as in Theorem 2. Then we add $s'_i(X)$ to $s'_i(Y)$ and get $x_i \oplus y_i$ in the leftmost bit. Then we rotate this to the position of x_i and get the function $S_i(X,Y) = \underbrace{00\cdots 0}_{i-1}(x_i \oplus y_i)\underbrace{00\cdots 0}_{n-i}$. Finally we note that $\oplus(X,Y) \equiv S_1(X,Y) + S_2(X,Y) + \cdots + S_n(X,Y)$. This concludes the proof. □

6 Cryptanalysis of generic AR systems

In this section we consider the AR schemes, which involve only modular additions and rotations (constants are admitted). Although they are theoretically

equivalent to the ARX systems, they are more vulnerable with the same number of operations. We show that the resulting function can be approximated with a simple formula, and this approximation becomes invalid only after a large number of additions.

The first important instrument is the following observation made in [16]:

$$X \lll_r \equiv 2^r \cdot X \pmod{2^n - 1}. \tag{1}$$

Let us note that addition modulo 2^n is equivalent to the addition modulo $2^n - 1$ if the sum is smaller than $2^n - 1$ (which happens with probability $1/2$). The main idea is to replace in a scheme $\mathcal{S}$ all the modular additions with additions modulo $2^n - 1$. With probability 2^{-q}, where q is the number of additions in the scheme, a new scheme $\mathcal{S}'$ produces the same output as $\mathcal{S}$. Finally, we replace rotations with multiplications using (1). As a result, we get a scheme with only additions and multiplications modulo $2^n - 1$. Since we multiply variables only by constants, the resulting function is linear in its inputs.

Corollary 1. *An AR scheme can be approximated by a linear function with probability 2^{-q} where q is the number of additions in the scheme.*

Applications

Any AR scheme claiming n bits of security and having less than n additions is potentially vulnerable to our attack. Consider, for example, an n-bit AR hash function H with $q < n$ additions. Given $H(M)$ we substitute it with a linear formula and solve a linear equation. If there is enough input freedom, we find a second preimage with complexity 2^q.

An n-bit AR block cipher is also vulnerable to this kind of attack. Let us replace all XORs in Threefish with additions (the constants remain). Then there are N_w additions per round and per subkey addition, so we can break a 50-round version, which would have 504 additions.

7 Conclusion

We have investigated security of ARX systems from both theoretical and practical points of view. We described a technique — rotational cryptanalysis — that is very efficient for ARX systems. The complexity of the rotational attack depends only on the number of modular additions, and does not depend on the number of XORs and rotations, nor on the rotation amounts. The rotational attacks on Threefish-256, -512, -1024 (39/42/43.5 rounds out of 72/72/80 rounds) are the best attacks on this primitive.

On the other hand, we proved that actually any function can be implemented with ARX operations and constants. As a result, there is no attack that works for any ARX system, though our rotational cryptanalysis demonstrates that secure systems must be large enough. Roughly, a primitive claiming n-bit security must

have at least $0.7n$ addition operations in any implementation. Use of constants, however, makes rotational cryptanalysis more complicated.

We also showed that though AR systems (not using XOR) are theoretically equivalent to ARX, they are vulnerable to a linear approximation attack regardless of constants used in the primitive.

Acknowledgements

The authors thank Christian Rechberger and Ralf-Philipp Weinmann for fruitful discussions and anonymous reviewers for their helpful comments. Dmitry Khovratovich is supported by PRP "Security & Trust" grant of the University of Luxembourg. Ivica Nikolić is supported by the BFR grant 07/031 of the FNR Luxembourg.

References

1. J.-P. Aumasson, E. Brier, W. Meier, M. Naya-Plasencia, and T. Peyrin. Inside the hypercube. In *ACISP'09*, volume 5594 of *Lecture Notes in Computer Science*, pages 202–213. Springer, 2009.
2. J.-P. Aumasson, Çagdas Çalik, W. Meier, O. Özen, R. C.-W. Phan, and K. Varici. Improved cryptanalysis of Skein. In *ASIACRYPT'09*, volume 5912 of *Lecture Notes in Computer Science*, pages 542–559. Springer.
3. J.-P. Aumasson, L. Henzen, W. Meier, and R. C.-W. Phan. SHA-3 proposal BLAKE. Submission to NIST, 2008.
4. D. J. Bernstein. Salsa20. Technical Report 2005/025. In *eSTREAM, ECRYPT Stream Cipher Project (2005), http://cr.yp.to/snuffle.html.*
5. D. J. Bernstein. CubeHash specification (2.b.1). Submission to NIST, 2008.
6. E. Biham. New types of cryptanalytic attacks using related keys. *J. Cryptology*, 7(4):229–246, 1994.
7. A. Biryukov and D. Khovratovich. Related-key cryptanalysis of the full AES-192 and AES-256. In *ASIACRYPT'09*, volume 5912 of *Lecture Notes in Computer Science*, pages 1–18. Springer, 2009.
8. E. Brier, S. Khazaei, W. Meier, and T. Peyrin. Linearization framework for collision attacks: Application to CubeHash and MD6. In *ASIACRYPT'09*, volume 5912 of *Lecture Notes in Computer Science*, pages 560–577. Springer, 2009.
9. C. D. Cannière and C. Rechberger. Finding SHA-1 characteristics: General results and applications. In *ASIACRYPT'06*, volume 4284 of *Lecture Notes in Computer Science*, pages 1–20. Springer, 2006.
10. F. Chabaud and A. Joux. Differential collisions in SHA-0. In *CRYPTO'98*, volume 1462 of *Lecture Notes in Computer Science*, pages 56–71. Springer, 1998.
11. J. Chen and K. Jia. Improved related-key boomerang attacks on round-reduced Threefish-512. Cryptology ePrint Archive, Report 2009/526, 2009.
12. M. Daum. *Cryptanalysis of Hash Functions of the MD4-Family.* PhD thesis, Ruhr-Universität Bochum, May 2005.
13. O. Dunkelman, S. Indesteege, and N. Keller. A differential-linear attack on 12-round Serpent. In *INDOCRYPT'08*, volume 5365 of *Lecture Notes in Computer Science*, pages 308–321. Springer, 2008.

14. N. Ferguson, S. Lucks, B. Schneier, D. Whiting, M. Bellare, T. Kohno, J. Callas, and J. Walker. The Skein hash function family (2008). In *Submitted to SHA-3 Competition.*
15. J. Kelsey, B. Schneier, and D. Wagner. Related-key cryptanalysis of 3-WAY, Biham-DES, CAST, DES-X, NewDES, RC2, and TEA. In *ICICS'97*, volume 1334 of *Lecture Notes in Computer Science*, pages 233–246. Springer, 1997.
16. J. Kelsey, B. Schneier, and D. Wagner. Mod n cryptanalysis, with applications against RC5P and M6. In *FSE'99*, volume 1636 of *Lecture Notes in Computer Science*, pages 139–155, 1999.
17. J. Kim, S. Hong, and B. Preneel. Related-key rectangle attacks on reduced AES-192 and AES-256. In *FSE'07*, volume 4593 of *Lecture Notes in Computer Science*, pages 225–241. Springer, 2007.
18. L. R. Knudsen, K. Matusiewicz, and S. S. Thomsen. Observations on the Shabal keyed permutation, 2009.
19. G. Leander. Private communication, 2009.
20. S. Paul and B. Preneel. Solving systems of differential equations of addition. In *ACISP'05*, volume 3574 of *Lecture Notes in Computer Science*, pages 75–88. Springer, 2005.
21. F.-X. Standaert, G. Piret, N. Gershenfeld, and J.-J. Quisquater. SEA: A scalable encryption algorithm for small embedded applications. In *CARDIS'06*, volume 3928 of *Lecture Notes in Computer Science*, pages 222–236. Springer, 2006.
22. E. Tews, R.-P. Weinmann, and A. Pyshkin. Breaking 104 bit WEP in less than 60 seconds. In *WISA*, volume 4867 of *Lecture Notes in Computer Science*, pages 188–202. Springer, 2007.
23. X. Wang and H. Yu. How to break MD5 and other hash functions. In *EUROCRYPT'05*, volume 3494 of *Lecture Notes in Computer Science*, pages 19–35. Springer, 2005.

A On the Probabilities of Rotational Characteristics for Threefish

One part of the probabilities of the trails presented at Tbl. 4 is computed theoretically, and one part practically.

When the rotational pair does not have corrections, then we use the probability of addition defined by Lemma 1. In the original versions (with C_5) the rotation amount is 2, so the probability of addition is $2^{-1.676}$. When C_5 is absent then we use a rotation amount 1, and the probability of addition becomes $2^{-1.415}$. Two consecutive rounds of Threefish-256 have 4 MIX and each MIX has one addition. Hence, two rounds with no subkey additions have a probability of $2^{-6.6}$. Analogously, for Threefish-512 and Threefish-1024 we get $2^{-13.3}$ and $2^{-26.6}$, respectively. These numbers translate into $2^{-5.7}, 2^{-11.3}, 2^{-22.6}$ for the versions without C_5.

When there are corrections in the rotational pair, we find the probabilities of two rounds (one round + subkey addition + one round) experimentally. The probabilities for the round 1 (key addition + one regular round) are also computed experimentally.

We have used to following corrections:

Rounds	Threefish-256		Threefish-512		Threefish-1024
	original	without C_5	original	without C_5	original
1	-13.1	-10.5	-22.8	-21.6	-45.6
$2-3$	-6.6	-5.7	-13.3	-11.3	-26.7
$4-5$	-17.57	-12.56	-29.47	-25.92	-61.48
$6-7$	-6.6	-5.7	-13.3	-11.3	-26.7
$8-9$	-15.33	-13.95	-31.33	-22.68	-63.32
$10-11$	-6.6	-5.7	-13.3	-11.3	-26.7
$12-13$	-15.60	-12.05	-29.73	-27.99	-61.68
$14-15$	-6.6	-5.7	-13.3	-11.3	-26.7
$16-17$	-21.08	-14.17	-34.35	-25.81	-66.44
$18-19$	-6.6	-5.7	-13.3	-11.3	-26.7
$20-21$	-18.46	-14.6	-37.25	-29.43	-68.82
$22-23$	-6.6	-5.7	-13.3	-11.3	-26.7
$24-25$	-21.47	-17.41	-34.38	-26.89	-66.34
$26-27$	-6.6	-5.7	-13.3	-11.3	-26.7
$28-29$	-21.55	-13.44	-36	-25.61	-67.31
$30-31$	-6.6	-5.7	-13.3	-11.3	-26.7
$32-33$	-21.74	-16.64	-37.63	-26.74	-69.28
$34-35$	-6.6	-5.7	-13.3	-11.3	-26.7
$36-37$	-22.96	-17	-38.17	-25.12	-67.79
$38-39$	-6.6	-5.7	-13.3	-11.3	-26.7
$40-41$		-17.74	-36.24	-31.34	-69.64
$42-43$		-5.7	-6.6	-11.3	-26.7
$44-45$		-18.89		-30.60	-13.3
$46-47$		-5.7		-11.3	
$48-49$		-2.8		-33.19	
$50-51$				-11.3	
52				-5.7	
Total rounds	39	44	42	51.5	43.5
Total probability	-254.1	-251.4	-507	-505.5	-1014.5

Table 4. Probabilities for the rotational pairs of different versions of Threefish.

– for Threefish-256 without C_5 in the key: 7, 2, 2, 6; in the plaintext: 2,2,7,6;
– for Threefish-512 without C_5 in the key: 2, 1, 3, 1, 7, 1, 7, 3; in the plaintext: 7,1,6,1,2,1,2,3