A Survey of ARX-based Symmetric-key Primitives

Nur Fasihah Mohd Esa[1], Shekh Faisal Abdul-Latip[1] and Mohd Rizuan Baharon[1]

[1]INSFORNET Centre for Advanced Computing Technology,
Fakulti Teknologi Maklumat dan Komunikasi, Universiti Teknikal Malaysia Melaka

***Abstract*:** Addition Rotation XOR is suitable for fast implementation symmetric –key primitives, such as stream and block ciphers. This paper presents a review of several block and stream ciphers based on ARX construction followed by the discussion on the security analysis of symmetric key primitives where the best attack for every cipher was carried out. We benchmark the implementation on software and hardware platforms according to the evaluation metrics. Therefore, this paper aims at providing a reference for a better selection of ARX design strategy.

***Keywords*:** ARX, cryptography, cryptanalysis, design, stream ciphers, block ciphers.

1. Introduction

The rapid development of today's computing technology has made computer devices became smaller which in turn poses a challenge to their security aspects. Current trend of technologies, for example, Internet of Things (IoT) and cloud computing, such as smart devices, wearable systems or any social media are able to help humans to perform their daily activities conveniently and more efficiently [1], [2]. Therefore, there is a huge demand of lightweight ciphers that specifically target resource-constrained devices either in software or hardware implementation to overcome these challenges [3]. As reviewed by Amiruddin et al. in [87], IoT security are still lacking in performance measurements, and have risk of vulnerability to the attack.

Previous studies have emphasized security analysis of lightweight block ciphers that are based on Substitution-Permutation Network (SPN) and Feistel Network which basically have S-box function as their main nonlinear component. Although S-box design contributes to the robustness of cryptosystem, look-up table needs to be formed; hence, making it vulnerable to time-cache attack. Therefore, to achieve fast implementation lightweight block cipher, trends of ARX with fewer operations are widely used in cryptography. Given their flexibility and efficiency, ARX designs are expected to gain popularity. However, the ARX approach has not been analysed extensively and any general framework for analysing ARX functions will have impact on future cipher designs. Desktop computers easily support 32-bit words, and many newer architectures support 64-bit words, all of which lead to cheap and efficient processing. In addition, the use of addition modulo 2n reduces memory footprint that may otherwise be used for substitution box table lookups [4].

The ARX consists of only these three operations: addition modulo of 2^n, bit rotation, and bitwise XOR. Modular addition is the only source that brings non-linear and confusion properties, which has the same function as S-box. Meanwhile, bitwise XOR and bit rotation contribute to the linear mixing and diffusion properties [5]. As previously reported in the literature, ARX mostly performs in compact and fast software-oriented implementation. Nevertheless, the security properties are still not well studied in literature as compared to SPN and Feistel ciphers.

Observation of addition from [4]: First, addition modulo 2^n on the window can be approximated by addition modulo 2^w. Second, this addition gives a perfect approximation if the carry into the window is estimated correctly. The probability distribution of the carry is generated, depending on the probability of approximation correctness. The probability of the carry is independent of *w*; in fact, for uniformly distributed addends it is $\frac{1}{2} + 2^{-i-1}$, where $i \geq 0$ is the position of the least significant bit in the window. Thirdly, the probability of correctness for a random guess of the value of the window decreases exponentially with w; it is $\frac{1}{2^w}$. Hence, the bias of the approximation (the difference between the probability of correctness of our approximation and that of a random guess) increases with w.

There have been numerous symmetric-key ciphers by using ARX paradigm published in the previous literature. For example, the stream cipher Salsa20 [6] and ChaCha20, conventional block cipher, [7] IDEA [8], TEA [9], XTEA [10], and followed by lightweight block ciphers, namely HIGHT [11], SPECK [12], LEA [13], Chaskey [14], SPARX [15], and CHAM [16]. As reported recently in 2018 by Hatzivasilis et al. [17], SPECK and LEA are the most efficient in software environment on small processors among all ARX ciphers.

There are three observations reported by Bernstein in [7] with reference to the significance of ARX paradigm selection. First, ARX is very effective in getting rid of look-up table related to S-box design paradigm that is vulnerable to timing attacks. Hence, side channel attack can be resisted by using ARX construction. Second, the total number of operations in the encryption process can be reduced to the minimum and thereby permitting increase of speed in software implementation. Third, computer code that describes such an algorithm is very small, making this approach particularly attractive for lightweight block ciphers where memory requirements are the most difficult.

The ARX, based symmetric-keys initiated by IDEA conventional block cipher was designed by Massey and Lai in 1993. IDEA is a modification structure from the Improved Proposed Encryption Standard (IPES). The purpose of IPES design was to replace DES cipher due to the controversial issue when DES became practically insecure because of its small key size of 56 bits and computational power increased [5].

Most of the studies in previous literature discussed the

efficiency of ARX construction towards software implementation. How about the efficiency of ARX symmetric performance in hardware oriented? Is ARX paradigm efficient in hardware?

- How far the security of ARX function has been studied?
- What are the most common attacks in ARX ciphers?
- What is the relation between the securities of stream and block ARX construction?
- Is there any continuation of future work of cryptanalysis tools against ARX designs?

1.1 Our Contribution

1 We provided the literature survey on symmetric keys included the structure component, and the weakness of the ciphers respectively.
2 We presented the literature of cryptanalysis and summarized the best attack on the ciphers.
3 We benchmarked the evaluation of performance of ARX block ciphers according to hardware and software platform.

1.2 Organization of this paper

This paper is structured as follows: Section 2 and Section 3 represent the discussion of the studies and characteristics of ARX-based stream and block cipher, respectively. The summary of security analysis from literature is described in Section 3. The benchmarking of implementation in software and hardware environments for ARX ciphers is reported in Section 5. Lastly, some conclusions and future works are drawn in Section 6.

2. The Studied Symmetric-key ARX-based Cryptography

This section provided the literature of the construction of symmetric-keys ARX based from the previous work. The main component of ARX symmetric-keys are block size, key size, and the number of iteration rounds.

2.1 The Description ARX Stream Ciphers

This subsection briefly discusses the description of stream cipher based on ARX construction. Salsa and ChaCha have a quarter round, respectively, which produce diffusion and confusion property.

2.1.1 Salsa20

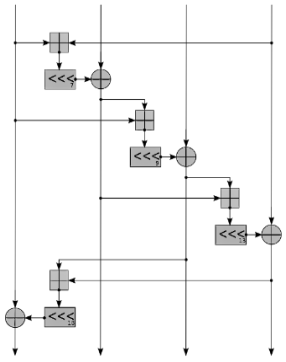

Figure 1. Salsa20 quarter round function

Salsa [6] is a 256-bit stream cipher introduced by Bernstein in 2005 with the aim of making it considerably faster than AES and implemented more easily with better security. It was designed as a chain of three simple operations on 32-bit words of ARX. Salsa came in three variants: Salsa20, Salsa12 and Salsa8, of 20, 12 and 8 rounds, respectively. Salsa20 was selected as a candidate of eStream [18]. Salsa is not only effectively in software environment but also has reasonable performance on hardware. Figure 1 shows the quarter round of Salsa20 built on pseudorandom function based on ARX operations.

2.1.2 ChaCha20

Later, after Salsa was introduced, Bernstein presented ChaCha [7], a 256-bit stream cipher modification from Salsa20, which was specifically designed to improve the amount of diffusion per round, and thus enhancing cryptanalysis resistance while maintaining and improving Salsa's performance. RFC 1654 provides further details regarding the implementation and security considerations [19]. Based on its latest achievement, ChaCha has received renewed recognition as the standard process for inclusion of cipher suites based on ChaCha20-Poly1305 AEAD (i.e. ChaCha20 for symmetric encryption and Poly1305 for authentication) in TLS 1.3 has nearly concluded [20]. Figure 2 shows the quarter round of ChaCha built on pseudorandom function based on ARX operations.

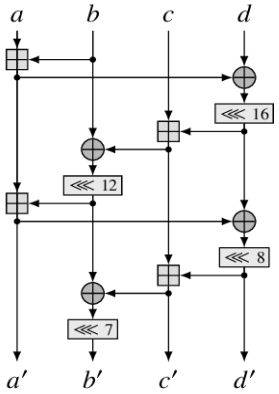

Figure 2. Chacha20 quarter-round function

2.2 The Description ARX Block Ciphers

This section briefly discusses the ARX block ciphers, such as IDEA, TEA, XTEA, HIGHT, SPECK, LEA, SPARX and CHAM. The operation of every block cipher was reviewed as well as the weaknesses of the structure that are vulnerable to cryptanalytic attack.

Table 1. List of symmetric-key ARX block ciphers

Block Cipher	Year	Block Size	Key Size	Structure	Rounds
IDEA	1991	64	128	Lai-Massey	8.5
TEA	1994	64	128	Feistel	64
XTEA	1997	64	128	Feistel	64
HIGHT	2006	64	128	GFN	32
SPECK	2013	32/48/64/96 /128	64/72/96/ 128/144/ 192/256	Feistel	22/23/26/ 27/28/29/ 32/33/34
LEA	2014	128	128/192/ 256	GFN	24/28/32
SPARX	2016	64/128	128/256	SPN	24/32/40
Cham	2017	64/128	128/256	GFN	80/80/96

2.2.1 IDEA block cipher

IDEA is a conventional symmetric block cipher introduced by Massey and Lai [21] that uses 64-bit blocks and 128-bit keys through 8.5 rounds of the output transformation (see Figure 3). The security of the cipher is enhanced by the composition of bitwise XOR, modulo addition and

Multiplications modulo with 16-bit unsigned integers in all data operations. IDEA is Lai-Massey scheme twice in parallel, with the other two parallel round functions intertwined to each other specifically for 16-bit per word that work well. To achieve an adequate diffusion, two sub-blocks are swapped after each round. For key schedule, each round requires six of 16-bit sub-keys, while the half-round uses 4; in total, 52 keys (48+4) for 8.5 rounds. For the first round, 6 sub-keys (K1-K6) were extracted from the master key of 128-bits and followed by the left rotation of 25 bits. Next, for further 8 sub-keys, they are extracted through the same process. This mechanism is repeated until all 52 sub-keys needed are generated [22]. The designer did not put any round constant in this structure. Therefore, it causes high vulnerability towards the weak keys, slide attack [23] and rotational attack [24][25].

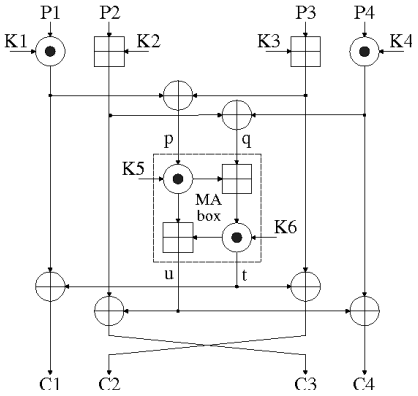

Figure 3 IDEA round function

2.2.2 *TEA Block Cipher*

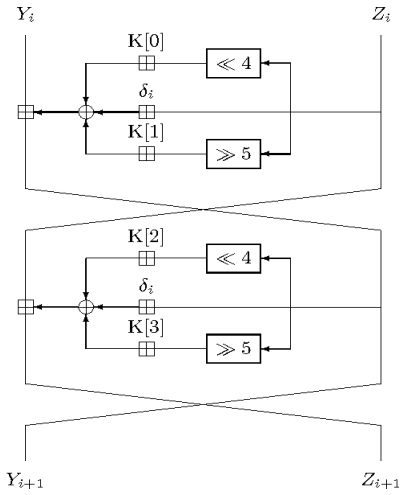

Figure 4. TEA Round function

TEA (Tiny Encryption Algorithm) was invented by Wheeler, Needham and Roger [9]. As illustrated in Figure 4, it operates on two 32-bit blocks (64-bit blocks) by using the 128-bit key. The structure is based on Feistel network with 64 rounds. Each round is implemented in pair of rounds that function as one round, with a totally simple key schedule when 128-bit master key splits into four 32-bit blocks and uses it repeatedly in successive rounds. TEA uses 0x9E3779B9 as a magic constant that is selected to be $\lfloor 2^{32}/\phi \rfloor$, to obstruct simple attacks based on the symmetry of the rounds, where ϕ is the golden ratio [26]. TEA has several weaknesses. TEA identifies an equivalent key, for example, each key is equivalent to three other keys. This means the effective key size of TEA is only 126 bits [27]. As a result, TEA is very inappropriate as a function of cryptographic hash. This vulnerability leads to the hacking of Microsoft's Xbox game console, where the cipher is used as a hash function [10]. The best attack TEA is related-key attack, requiring 2^{23} chosen plaintexts under the related key pairs, with 2^{32} time complexity [28]. XTEA design cipher is therefore proposed to improve the key schedule structure to correct the weaknesses in TEA.

2.2.3 *XTEA Block Cipher*

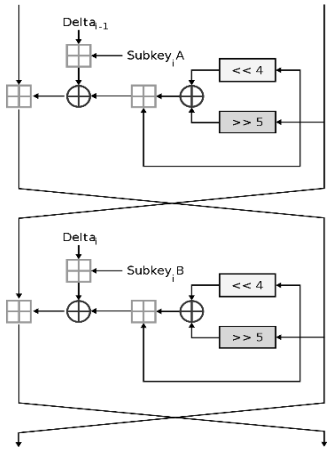

Figure 5 XTEA Round function

The extended TEA (XTEA) was pioneered by David Wheelerand Roger Needham from the Cambridge Computer Lab [25]. The algorithm was presented in a non-published technical report in 1997 (Needham & Wheeler, 1997). It is not subject to any patent [26]. Like TEA, XTEA is a block of 64-bit blocks of Feistel with 128-bit keys and 64 rounds reserved as shown in Figure 5. Some differences from the TEA are clear, including a relatively complex main table and restructuring of changes, XORs, and additions

2.2.4 *HIGHT Block Cipher*

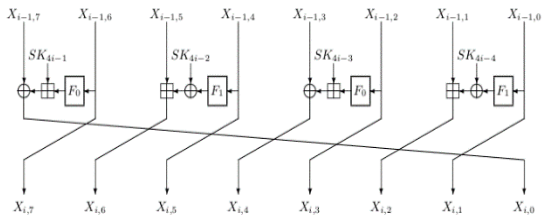

Figure 6. HIGHT Round function

HIGHT is a lightweight block cipher proposed by Hong et al. [11] in 2006 at CHES conference in Japan. HIGHT is designed specifically for low resource devices with 64-bit blocks and 128-bit key length. Figure 5 illustrated the structure of HIGHT block cipher. To have a high efficiency on hardware platform, HIGHT is designed based on GFN (8 branch Feistel Network) by 8-bit unsigned integer ARX operation with the iteration of 32 rounds. The XOR operation is sometimes replaced by a modular addition, whereas the two internal functions in Feistel network is replaced by the combination of XOR and left and right shift circular to achieve an optimum diffusion. To avoid direct retrieval from plaintext and ciphertext, HIGHT is injected by key whitening at the first and the last rounds. LFSR technique is used to generate a sequence of constant for randomness of sub-key enhancement and slide attack resistance. In contrast, LFSR design is typically bulky and requires more power for implementation [29].

2.2.5 SPECK Block Cipher

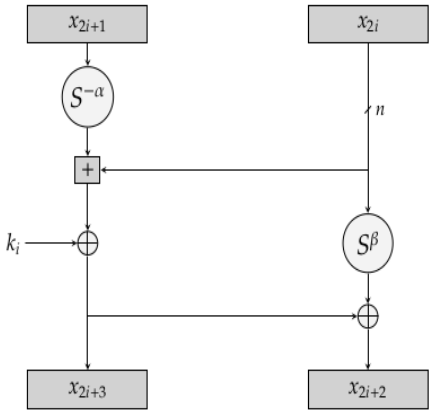

Figure 7 SPECK round function

The SPECK families block cipher was officially proposed by the NASA in 2013 [12] and designed by Beaulieu et al. with SIMON block cipher family. SPECK is designed to supply the security demand of enough flexibility in the era of IoT as well as to provide the resistance of related-keys attack [30][31]. The designer claimed that SPECK has an excellent efficiency in software and hardware platforms. In addition, the existing block ciphers are less flexible on variation of platform application. It uses Feistel structure with both branches amended by using bitwise XOR, addition and rotation for each round in both directions. For key schedule consideration, SPECK adopts round function (refer Figure 7) for its structure to allow reduction in code size and to improve the performance for software implementation, requiring on-the-fly round key generation. The choice of rotation constant is according to the addition counters for efficiency in software as compared to hardware implementation.

2.2.6 LEA Block Cipher

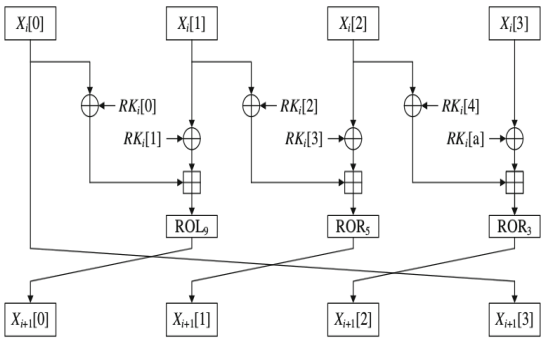

Figure 8. Round function of LEA

Hong et al. [13] proposed the LEA software-oriented lightweight block cipher in 2013. It has block size of 128 bits and key size of 128, 192, or 256 bits that consists of simple operations of ARX for 32-bit words through 24, 28, and 32 rounds, respectively. In addition, diffusion property is achieved by the swapping of word branch and bit rotation operation. Key schedule is generated by 192-bit round key sequence without mixing the words. LEA uses constant from the hexadecimal expression of $\sqrt{766995}$, where 76, 69, and 95 represent the ASCII codes for 'L', 'E', and 'A', respectively. Figure 8 represent the round function of LEA's structure.

2.2.7 SPARX Block Cipher

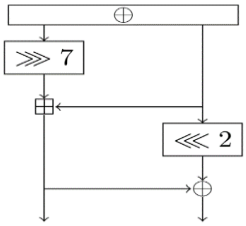

Figure 9. ARX-Box

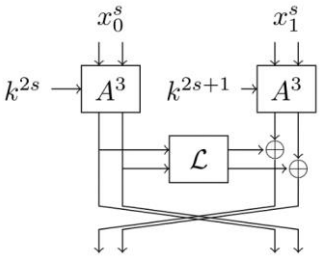

Figure 10. Round function of SPARX

Among all the above ciphers, SPARX lightweight block cipher is the borrowed cipher based on SPN structure. SPARX is designed by Dinu et al. [15] in 2016. Dinu et al., replacing ARX-box as non-linear function with a new cipher, namely Speckey (see Figure 9) that was introduced by Biryukov et al. [32]. Speckey is a modification of key addition to the full state of round instead of half-round of state from Speck-32 [12]. Biryukov et al. in [32] - proposed several ARX constructions, such as Speckey-32 and MARX-32 that are feasible to compute the exact maximum differential and linear probabilities over any number of rounds. Speckey-32 was chosen as ARX-box because of the fewer operations [15]. Besides, Dinu et al. implemented the technique of Long Trail Strategy to achieve a maximum diffusion in each linear layer. Dinu et al. reused the round functions component in the key schedule itself to limit the code size. The round function of SPARX are shown in Figure 10.

2.2.8 CHAM Block Cipher

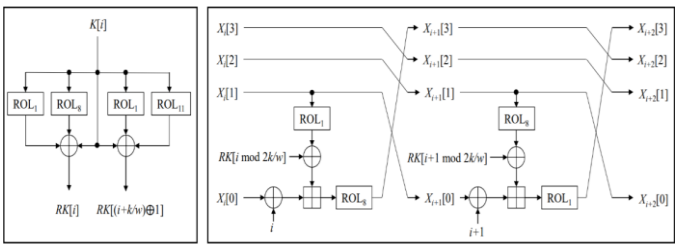

Figure 11. Key Schedule (left) and two consecutive round functions beginning with the even i-th round (right) for CHAM

CHAM lightweight block cipher that contributed to the improved structure is better than LEA by boosting up the level of stability for resource-constrained platform. The main goal of CHAM design is to have better efficiency than SIMON and SPECK in both hardware and software oriented IoT technologies. CHAM ARX based construction is designed by Koo et al. [16] which generalised 4-branch Feistel structure (see Figure 10). The family of CHAM consists of 64-bit and 128-bit block size while the key lengths consist of 128 and 256 bits through 80 and 96 rounds. CHAM uses an extremely simple structure of round

function and key schedule. To minimise the number of flip-flops in hardware implementation, the key schedule is implemented without updating any key states.

3. Security Analysis of the Symmetric-key ARX Structure

A large amount of works has been performed in the academy community to understand the security of the algorithms. This section discusses the review of security analysis among symmetric-keys cryptography.

3.1. Cryptanalysis of ARX Stream Cipher

As previously reported in the literature, the study on Salsa and ChaCha families cryptanalysis are rarely published by the researchers in the last 10 years thus, making it surprising to see the increased number of researchers who are interested in analysing the same in recent years, such as fault analysis [33], differential cryptanalysis and linear cryptanalysis [34].

3.1.1. Salsa20 and ChaCha20

The first result on Salsa20 cryptanalysis by [35] used this idea to move three rounds forward from the initial state and to go two rounds back from the final state for an attack on 5-round Salsa20. Later in [36], biased differentials till four forward rounds have been exploited and then the attack was moved backwards from the final state for four rounds to obtain an 8-round attack on Salsa20. In 2015, Maitra et al. [37] improved the complexity by revisiting the method that came from the idea of Probabilistic Neutral Bit (PNB) and determined a better choice of certain parameters. Later, Deepthi and Singh [38] revisited Maitra et al.'s [37] study where they found the mistake in one-bit change in the 8th and 9th word in the first round that resulted in valid initial state. Also, by applying the same concept of method by Maitra et al. [39], 128-key bit of Salsa20/7 was attacked within time 2^101and ChaCha7 within time 2^101.

Mouha et al. proposed an efficient toolkit to find the optimal charateristics for ARX ciphers application to Salsa20 [40]. Mouha et al. proved in their paper [41] that Salsa is secured against differential cryptanalysis until 15-round [41]. However, there is still a gap where there are no proofs of resistance of linear cryptanalysis in any literature. In addition, Mouha et al. suggested to increase the bound of provable of differential cryptanalysis regarding their untighten security findings. Aaraj et al. used MILP techniques to search for differential trail for two rounds of ChaCha at bit-level, and six rounds at word level [33]. In [42], Saho et al. proposed the algorithm to search for differential cryptanalysis applied on ChaCha stream cipher. However, the search algorithm also has some defects that need to be improved and optimised. Recently in 2017, Kumar et al. [33] presented the first practical differential fault attack on the ChaCha family.

3.2. Cryptanalysis of ARX Block Cipher

The literature review on cryptanalysis of ARX block cipher is discussed as below subsections. The summary of the best attack are presented in Table 2.

3.2.1. IDEA block cipher

Daemen et al. [8] have discovered 2^{51} differential weak keys on IDEA block cipher in 1993 using a test on membership of the weak key class, explaining that the test only needs two encryptions and solution of sixteen non-linear Boolean equations each of which has twelve variables. Soon after five years, Hawkes [9] improved the research when he developed 2^{63} differential weak keys for full-round cipher. Later, Wagner [10] searched 2^{63} weak keys by using cryptanalytic attack, namely Boomerang attack. Consequently, Sahu et al. [11] discussed the countermeasure of weak keys that is to insert any round constant to be XORed to all round keys. Sahu et al. reported work related regarding cryptanalytic attacks on the IDEA in literature until 2015 [43]. The authors concluded that IDEA is secure for confidentiality and is still relevant after 25 years as there has not been a single attack that could practically break it in full round of IDEA block cipher.

3.2.2. TEA and XTEA

There are several cryptanalytic results for TEA and XTEA. TEA is attacked by full round related key cryptanalysis due to the exploitation of its extremely simple key schedule [27]. Shepherd [44] reported three equivalent keys that have been found in each round, hence, reducing the effectiveness of key space from 128 bits to 126 bits. Despite this minor change regarding brute-force attack, even a strong algorithm is highly exposed to weak implementation because of this incident. For a single key attack setting, Moon et al. (2002) [10] presented impossible differential cryptanalysis of 14-round XTEA and 11-round TEA based on 12-round and 12-round impossible differential, respectively. Hong et al. [28] proposed the work of truncated differential cryptanalysis that breaks TEA in reduced round of 17 and XTEA for reduced round of 23 with the complexity of $2^{123.73}$ and $2^{120.65}$, respectively. Sekar et al. proposed meet-in-the-middle attack on 23 rounds of XTEA with complexity 2^{117} [45]. Bogdanov and Wang (2012) [46] presented techniques of zero-correlation linear cryptanalysis which breaks 23 round of TEA and 27 rounds of XTEA, by using the whole code book. Chen et al. [47] later improved the impossible cryptanalysis of 17-round TEA and 23-round XTEA based on 14 rounds and 13 rounds of impossible differential, respectively. In 2014, Biryukov and Velichkov [42] proposed the first differential full trail with 18-round TEA block cipher. Hong et al. 2003 found full trail for 14-round XTEA with probability of $2^{60.76}$ [28].

3.2.3. HIGHT

Azimi et al. proposed reduced rounds from the 27-round impossible differential attack [48]. Meanwhile, Chen et al. (2012) [47] proposed techniques to improve the speed of exhaustive search of impossible differential characteristics and enhance the result from 26 rounds by Ozen et al. [49] to 27 rounds. Related key on 31-round HIGHT utilised 22-round related-key impossible differential. Also, full round Biclique attack was proposed in [50].

3.2.4. SPECK

Dinur [51] showed that an r-round differential distinguisher yields at least an (r + m)-round attack, where m is the number of words of key. For Speck, there is also a slight multipath effect for differences, and so in some cases, an additional round can be gained, or the data requirement can be reduced, as noted by Song et al. [52]. Additional rounds added to obtain a security buffer like AES-128, which is ample at 30%. (See, for example, [53]; the best current attack on AES-128 in the standard attack model is on seven of its

10 rounds.) For example, consider Speck 128/128. As noted above, difference paths extend through 20 rounds; the probability drops below 2−128 at 21 rounds. By using the results of Dinur, the 20-round path results in an attack on $r + m = 20+2 = 22$ rounds of Speck128/128. To get a 30% margin with respect to this attack would require 31 rounds, and the stepping for Speck128/128 was set at 32 rounds. (Moreover, because Speck128/192 and Speck128/256 have one and two more words of key, respectively, we set the stepping at 32 + 1 = 33 for 12 Speck128/192 and 32 + 2 = 34 for Speck128/256.) Dinur's idea was extended by [52], where it showed that (r + m + 1)-round attacks are possible. The result is an attack on 23 out of the 32 rounds, which is currently the best attack on Speck128/128. This leaves Speck128/128 with a 28% security margin, like AES-128.

The best linear paths are notably weaker than the best difference paths, with squared correlations dropping below 2−block size in fewer rounds that is necessary for the difference path probabilities. This is in line with what was found (through non-exhaustive searches) in [54]. In [55], it is proven that for Speck32, Speck48, and Speck64, the squared correlations fall below 2−block size in 10, 11, and 14 rounds, respectively. The linear paths tend to exhibit a stronger multipath effect, but the best linear attacks for Speck are still worse in every case than the best differential attacks.

For Speck64, 6-round impossible differentials have been found by using MILP techniques [56]. Any resulting attack would not be competitive with the best current attack on Speck64/128, for example, which is a differential attack on 20 of its 27 rounds [52]. Zero correlation distinguishers on Speck, like the impossible differentials, only appear to get through a handful of rounds.

3.2.5. LEA

Zhang et al. [57] proposed 1st zero correlation attack on 1-round LEA128, 13-round LEA-192 and 14-round LEA-256, (distinguishing attack, not key recovery attack). Song et al. [52] improved the attack of differential by two more rounds from the previous work [13], LEA-128, from 12 rounds to 14 rounds and LEA-192 and LEA-256 1st work in [52], 14 rounds and 15 rounds respectively. Based on the literature, further cryptanalysis could be applied on LEA block cipher.

3.2.6. SPARX

Ankele and List [58] presented truncated differential cryptanalysis on reduced round of SPARX 64/128 until 16 rounds and used single differential characteristics, for the first part of the 14-round distinguisher and truncated the second part of the distinguisher. The designers of SPARX-64, Dinu et al. claimed that SPARX is resistant to differential cryptanalysis for 15 rounds. However, according to the literature in [59], Ankele and Kolbl stated that by considering the differential effect of SPARX-64, also in comparison with SPECK-64, it seems there exist differentials with more than 15 rounds with data complexity by using less than the full codebook. Tolba et al. proposed zero linear correlation attack [60] of SPARX-128 where 24- and 25-round zero correlation distinguishers are used to launch key recovery attacks against 28, 29 rounds (7, 7.25 out of 10 steps) of SPARX-128/256 and 26 rounds (6.5 out of 8 steps) of SPARX-128/128. Tolba et al. [61] claimed their techniques are the first third party attacks against SPARX-128/128 and SPARX-128/256.

3.2.7. CHAM

CHAM lightweight block cipher was recently published in 2018 [16]. Therefore, not much of cryptanalysis is discovered yet in the literature. The designers, Koo et al. claimed in their paper that CHAM is resistant to many attacks such as differential cryptanalysis, linear cryptanalysis, boomerang cryptanalysis, rotational-XOR-differential cryptanalysis, etc. The best attack for CHAM 64/128, 128/128, 128/256 are 4-rounds related key boomerang cryptanalysis, 47-round boomerang cryptanalysis, and 47-round related key (boomerang cryptanalysis) and boomerang cryptanalysis, respectively.

Table 2. Summary of best attack in ARX ciphers

Block Cipher	Best Attack	Rounds
IDEA	Related-key attack	Full round
TEA	zero-correlation linear cryptanalysis	23
XTEA	zero-correlation linear cryptanalysis	27
HIGHT	Biclique attack	Full round
SPECK	Differential cryptanalysis	14
SPARX	Truncated Differential cryptanalysis	16
CHAM 64/128	Related key (Boomerang Cryptanalysis)	41
CHAM 128/128	Boomerang Cryptanalysis	47
CHAM 128/256	Related key (Boomerang Cryptanalysis) and Boomerang	47

4. Automated tools of selected cryptanalysis against ARX Design

This section provided the selections of cryptanalysis techniques against ARX designs construction. The literature review on the automated tools of selected cryptanalysis against ARX Designs are discussed below**.**

4.1. Differential and Linear Cryptanalysis

Differential Cryptanalysis was initiated by Biham and Shamir [62]. The previous studies have emphasized the techniques of analysis of differential characteristics and differential cryptanalysis based on ARX design. The works are categorized into three approaches: bottom-up, top-down and approximation-based techniques. We briefly desribe the categories below.

4.1.1. Bottom-up Techniques

This category is by far the largest and encompasses methods for the (automatic) construction of differential and linear trails in ARX. Arguably the first such techniques dated back to the collisions of the MD and SHA families of hash functions by Wang et al. [63]–[65]. While these results were reportedly developed by hand, subsequent methods were proposed for the fully automatic construction of differential paths in ARX, all of which were applied to augmented ARX designs, for instance, SHA1, SHA2, MD4 and MD5. In [66] a method was proposed for the automatic construction of differential trails in pure ARX designs and applied to the hash function Skein. While many of the mentioned techniques are general and potentially applicable to any ARX primitive, all of them were applied exclusively to hash functions. To fill the gap, [67] the threshold search method was proposed for searching of differential trails in ARX ciphers, such as TEA, XTEA and Speck. This method was subsequently extended to the case of differentials in [68]. Most recently, in 2015, two new techniques for automatic

search for linear trails have been proposed. One has been applied to Speck [69], while the other is dedicated to authenticated encryption schemes [70]. In 2016, Biryukov et al. [32] improved the search of best trail for both linear and differential cryptanalysis from the adaptation of Matsui Algorithm based on branch and bound strategy.

4.1.2. Top-down Techniques

Top down techniques are considered of cipher that causes this technique to be better than the bottom-down technique that builds only one round trail at a time. More precisely, the cipher can be represented either as a Boolean equation system or as an integer mixture of inequality system. Each solution to the system corresponds to a valid trail. In the first case, the Boolean equations are transformed into a conjunctive normal form (CNF) formula, whose satisfying assignment/s is/are found with a SAT solver. In the second case, the problem of searching for 6 Alex Biryukov, Vesselin Velichkov, and Yann Le Corre trails is effectively transformed into a mixed-integer linear problem (MILP) that is usually solved by MILP's dedicated solver who uses branching and bound techniques by linear programming. The SAT solver approach has been used to find the best differential trails for several rounds of stream cipher Salsa20 and for proving security bounds for the authenticated encryption cipher NORX. As for the MILP-based methods, up to now they have been successful mainly in the analysis of S-box designs [71], [72]. The only applications of MILP to ARX available are the results on the augmented ARX cipher Simon [72] and a very recent paper [54] on Speck appearing in this volume of FSE'16.

4.1.3. Approximation-based Techniques

In both top-down and bottom-up approaches, complex techniques for an existing algorithm analysis were initiated. On the other hand, in approximation-based techniques, the problem is changing: a new primitive is developed so that it can easily be analysed by design. The main idea is to replace the non-linear component of ARX – the modular addition – by a simpler non-linear approximation that can be efficiently and accurately analysed with existing methods. A design based on this strategy is the authenticated encryption scheme NORX [73]. In it, the addition operation is replaced by the first-order approximation $a \oplus b \oplus (a \wedge b) \ll 1 \approx a \boxplus b$, which effectively limits the carry propagation to a sliding window of 2 bits. The latter significantly facilitates the analysis of the scheme and makes it hardware efficient.

4.2. Rotational Cryptanalysis

Rotation cryptanalysis is a selected plaintext cryptanalytic technique proposed by Khovratovich et al. in [25]. In cryptanalysis rotation, the enemy requests the encryption of a plaintexts pair, in which one plaintext is obtained through another cycle of cycles. This is done under two related keys, which are also spin pairs. Khovratovich et al. mentioned that the rotational relation between the two inputs is maintained with probability through ARX operation. A proposed response to rotary cryptanalysis is for a constant dependent on the XOR round, which crawls the probability of propagation.

Adrian et al. [74] suggested in papers to enhance ArxPy by finding all the features that share the difference of RX inputs and RX outputs of the optimum feature differences to achieve a better estimate of the probability. Additionally, work can be extended by using a different offset spin from 1 and is injected via modular addition instead of XOR

5. Implementation Evaluation

Benchmarking was conducted on 19 hardware and 33 software implementations of ARX block ciphers. The implementations were evaluated based on the metrics and the fair comparison approach detailed in Section 2. In hardware, 0.09, 0.13, 0.18, and 0.35 µm technologies were used. The software implementations deployed 8-, 16- and 32-bit microcontrollers. Software Software implementation performance metrics and its definition briefly explained in Table 3.

Table 3. Software implementation performance metrics

Metric Definition	Definition
Code size (bytes)	Memory size to store the cipher code and constants. Typically, resides in flash memory
RAM size (bytes)	Memory size to store the intermediate state during the execution of the cipher code.
Cycles/byte	Number of cycles to encrypt
Throughput	Number of encrypted bits per seconds (Kbps)
Combined Metric	(Code size X Cycle count) / Block size

5.1 Software Implementation

Table 4 (refer page 11) illustrates the best software implementations according to individual metrics. IDEA exhibited high latency in software (more than 10,000 cycles). The most compact implementations (in terms of code size) were reported for SPECK, HIGHT and XTEA (less than 0.5KB). The implementations with the higher throughput are those of SPECK, LEA, IDEA, and HIGHT (more than 85 Kbps). CHAM, SPECK, SIMON, and HIGHT achieved the lowest latency (less than 3000 cycles). Based on the criteria adopted in this work, SPECK and CHAM evaluations were best, as compared to all the other proposals for many key sizes. SPECK also offers efficient encryption/decryption implementations with low cost.

5.2. Hardware Implementation

Based on Table 5 (refer page 11), the proposals for the ciphers IDEA and LEA exceed the boundary of 3000GE and therefore, not considered efficient. None of the ARX ciphers consumes excessive energy per bit and produce low hardware efficiency. TEA consumes high energy per bit. HIGHT has higher power requirements than other ciphers. SPECK produces compact implementations with the lowest power requirements (less than 1µW). XTEA consumes low power (around 2.5µW). In addition, SPECK and CHAM consume low energy per bit (less than 10 µJ per bit). Based on the criteria adopted in this work, SPECK and CHAM exhibited low latency and efficient implementations, achieved good overall status, and consumed low energy but were newly proposed ciphers. The most compact ciphers that require around 1000GE and provided key sizes between 80- and 256-bits are SPECK and CHAM that performed the best.

6. Conclusion

This paper discusses the study and the benchmark of symmetric-keys based on ARX construction in which the literature on the security and implementation of software and hardware platform were reviewed. It is believed that there is still work needed on cryptanalysis of ARX based design especially in differential, linear and rotational cryptanalysis.

In terms of efficient implementation, CHAM and SPECK shine in both domains and HIGHT in hardware, while the rest of the ciphers perform reasonably well in both domains and do not excel in any of them.

7. Future Works

- Adrian et al. [74] suggested in his paper that ArxPy need to be improved by finding all the characteristics that share the input RX difference and the output RX difference of optimal characteristics to achive better approximation of the probability. Also, the work could be extended by using the rotational offset that differs from 1, and the constant injected through modular addition instead of XOR.
- Rotational Cryptanalysis to be applied on ARX design cipher other than SPECK.
- More cryptanalysis need be explored in Salsa and ChaCha stream cipher.
- The study of ARX function need to be extended such as the improvement of carry pattern selection for different input distributions and the study of other properties of linearity to improve bias in ARX construction.
- The design approach of ARX construction that can have the same security level as the AES block cipher.
- Mathematical background of ARX function also needs attention.

8. Acknowledgement

This research paper is supported by Short Term Grant (PJP Grant) numbered PJP/2019/FTMK(2B)/S01673 funded by the Universiti Teknikal Malaysia Melaka (UTeM), Malaysia.

References

[1] M. Conti, A, Dehghantanha, K. Franke, and S. Watson, "Internet of Things security and forensics: Challenges and opportunities," Future Generation Computer Systems, Vol. 78, No 2, pp. 544-546, 2018.

[2] C. Stergiou, K. E. Psannis, B. G, Kim, B. Gupta, "Secure integration of IoT and cloud computing" Future Generation Computer Systems, Vol 78, No 3, pp. 964-975, 2018.

[3] C. Pei, Y. Xiao, W. Liang, X. Han, "Trade-off of security and performance of lightweight block ciphers in Industrial Wireless Sensor Networks". EURASIP Journal on Wireless Communications and Networking, p. 117, 2018.

[4] K. A. McKay, P. L. Vora, "Analysis of ARX Functions: Pseudo-linear Methods for Approximation, Differentials, and Evaluating Diffusion," IACR Cryptology ePrint Archive, p. 895, 2014.

[5] V. Velichkov, N. Mouha, C. De Canniere, B. Preneel, "The additive differential probability of ARX," In International Workshop on Fast Software Encryption, Springer, Berlin, Heidelberg, pp. 342-358, 2011

[6] D. J. Bernstein, "The Salsa20 family of stream ciphers," In New stream cipher. Springer, Berlin, Heidelberg, pp. 84-97, 2008.

[7] D. J. Bernstein, "ChaCha, a variant of Salsa20," In Workshop Record of State of the Art of Stream Ciphers (SASC), Vol. 8, pp. 3-5, 2008.

[8] W. Meier, "On the security of the IDEA block cipher," Workshop on the Theory and Application of of Cryptographic Techniques, Springer, Berlin, Heidelberg, Vol. 765, pp. 371-385, 1993.

[9] D.J. Wheeler, R. M. Needham, "TEA, a tiny encryption algorithm," In International Workshop on Fast Software Encryption, Springer, Berlin, Heidelberg, pp. 363-366, 1994.

[10] D. Moon, K. Hwang, W. Lee, S. Lee, J. Lim, "Impossible differential cryptanalysis of reduced round XTEA and TEA," International Workshop on Fast Software Encryption. Springer, Berlin, Heidelberg, pp. 49-60, 2002.

[11] D. Hong, J. Sung, S. Hong, J. Lim, S. Lee, B.S. Koo, C. Lee, D. Chang, J. Lee, K. Jeong, H. Kim, "HIGHT: A new block cipher suitable for low-resource device," In International Workshop on Cryptographic Hardware and Embedded Systems, Springer, Berlin, Heidelberg, pp. 46-59, 2006.

[12] R. Beaulieu, D. Shors, J. Smith, S. Treatman-Clark, B. Weeks, L. Wingers, "The SIMON and SPECK lightweight block ciphers," In Proceedings of the 52nd Annual Design Automation Conference, ACM, p. 175, 2015.

[13] D. Hong, J. K. Lee, D. C. Kim, D. Kwon, K. H. Ryu, D. G. Lee, "LEA: A 128-bit block cipher for fast encryption on common processors," In International Workshop on Information Security Applications, Springer, Cham, pp. 3-27, 2013.

[14] N. Mouha, B. Mennink, A. Van Herrewege, D. Watanabe, B. Preneel, I. Verbauwhede, "Chaskey: an efficient MAC algorithm for 32-bit microcontrollers," In International Conference on Selected Areas in Cryptography, Springer, Cham, pp. 306-323, 2014.

[15] D. Dinu, L. Perrin, A. Udovenko, V. Velichkov, J. Großschädl, A. Biryukov, "Design strategies for ARX with provable bounds: Sparx and LAX," In International Conference on the Theory and Application of Cryptology and Information Security. Springer, Berlin, Heidelberg, pp. 484-513, 2016.

[16] B. Koo, D. Roh, H. Kim, Y. Jung, D. G. Lee, D. Kwon, "CHAM: a family of lightweight block ciphers for resource-constrained devices" In International Conference on Information Security and Cryptology, Springer, Cham, pp. 3-25, 2017.

[17] G. Hatzivasilis, K. Fysarakis, I. Papaefstathiou, C. Manifavas, C, "A review of lightweight block ciphers" Journal of Cryptographic Engineering, Vol. 8, No. 2, pp.141-184, 2018.

[18] S. Babbage, C. Canniere, A. Canteaut, C. Cid, H. Gilbert, T. Johansson, M. Parker, B. Preneel, V. Rijmen, M. Robshaw, 2 "The eSTREAM portfolio," eSTREAM, ECRYPT Stream Cipher Project, pp.1-6, 2008.

[19] Y. Nir, A. Langley, "ChaCha20 and Poly1305 for IETF Protocols," RFC 7539 (Informational), Internet Engineering Task Force, 2015.

[20] E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.3," Vol. 5246, pp. 1–137, 2017.

[21] X. Lai, J. L. Massey, "A proposal for a new block encryption standard," In Workshop on the Theory and Application of Cryptographic Techniques. Springer, Berlin, Heidelberg, pp. 389-404, 1990.

[22] H. K. Sahu, V. Jadhav, S. Sonavane, and R. K. Sharma, "Cryptanalytic Attacks on International Data Encryption Algorithm Block Cipher," Defence Science Journal, Vol. 66, No. 6, pp 582-589, 2016.

[23] A. Biryukov, D. Wagner, "Slide attacks," In International Workshop on Fast Software Encryption, Springer, Berlin, Heidelberg, pp. 245-259, 1999.

[24] D. Khovratovich, I. Nikolić, "Rotational cryptanalysis of ARX," In International Workshop on Fast Software Encryption, Springer, Berlin, Heidelberg, pp. 333-346, 2010.

[25] D. Khovratovich, I. Nikolić, J. Pieprzyk, P. Sokołowski, R. Steinfeld, "Rotational cryptanalysis of ARX revisited," In *International Workshop on Fast Software Encryption*. Springer, Berlin, Heidelberg, pp. 519-536, 2015.

[26] J. C. Hernandez and P. Isasi, "Finding efficient distinguishers for cryptographic mappings, with an application to the block cipher TEA," Computational Intelligence, Vol 20, No 3, pp. 517-525, 2004.

[27] J. Kelsey, B. Schneier, D. Wagner, "Related-key cryptanalysis of 3-way, biham-des, cast, des-x, newdes, rc2, and tea," In International Conference on Information and

Communications Security, Springer, Berlin, Heidelberg, pp. 233-246, 1997

[28] S. Hong, D. Hong, Y. Ko, D. Chang, W. Lee, S. Lee, "Differential Cryptanalysis of TEA and XTEA," In International Conference on Information Security and Cryptology, Springer, Berlin, Heidelberg. pp. 402-41, 2004.

[29] B. J. Mohd, T. Hayajneh, Z. A. Khalaf, K. M. Ahmad Yousef, "Modeling and optimization of the lightweight HIGHT block cipher design with FPGA implementation," Security and Communication Networks, Vol 9, No. 13, pp. 2200-2216, 2016.

[30] R. Beaulieu, D. Shors, J. Smith, S. Treatman-Clark, B. Weeks, L. Wingers, "SIMON and SPECK: Block Ciphers for the Internet of Things," IACR Cryptology ePrint Archive, p.585, 2015.

[31] R. Beaulieu, D. Shors, J. Smith, S. Treatman-Clark, B. Weeks, L. Wingers, "Notes on the design and analysis of SIMON and SPECK," IACR Cryptology ePrint Archive, 2017, p. 560, 2017.

[32] A. Biryukov, V. Velichkov, and Y. Le Corre, "Automatic search for the best trails in ARX: application to block cipher speck" In International Conference on Fast Software Encryption, Springer, Berlin, Heidelberg, Vol. 9783, pp. 289-310, 2016.

[33] S. D. Kumar, S.D. Patranabis, J. Breier, D. Mukhopadhyay, S. Bhasin, A. Chattopadhyay, A. Baksi, "A practical fault attack on ARX-like ciphers with a case study on ChaCha20," In Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC), IEEE, pp. 33-40, 2017.

[34] N. Aaraj, F. Caullery, and M. Manzano, "MILP-aided Cryptanalysis of Round Reduced ChaCha," IACR Cryptology ePrint Archive, p. 1163, 2017.

[35] P. Crowley, "Truncated differential cryptanalysis of five rounds of Salsa20," The State of the Art of Stream Ciphers SASC, pp.198-202, 2006.

[36] J. P. Aumasson, S. Fischer, S. Khazaei, W. Meier, and C. Rechberger, "New features of Latin dances: analysis of Salsa, ChaCha, and Rumba," In International Workshop on Fast Software Encryption, Springer, Berlin, Heidelberg, pp. 470-488, 2008.

[37] S. Maitra, G. Paul, and W. Meier, "Salsa20 Cryptanalysis: New Moves and Revisiting Old Styles." IACR Cryptology ePrint Archive 2015, p. 217, 2015.

[38] K. K. Deepthi, K. Singh, "Cryptanalysis of Salsa and ChaCha: Revisited," In International Conference on Mobile Networks and Management, Springer, Cham, pp. 324-338, 2017.

[39] S. Maitra, "Chosen IV cryptanalysis on reduced round ChaCha and Salsa," Discrete Applied Mathematics, Vol. 208, pp.88-97, 2016.

[40] N. Mouha, B. Preneel, "Towards finding optimal differential characteristics for ARX: Application to Salsa20," 2013.

[41] N. Mouha, B. Preneel, "A Proof that the ARX Cipher Salsa20 is Secure against Differential Cryptanalysis," IACR Cryptology ePrint Archive, p. 328, 2013.

[42] P. Shao, G. Zhang, and M. Li, "Automatic search for differential characteristics in ARX ciphers," In 10th International Conference on Natural Computation (ICNC), IEEE, pp. 1009-1013, 2014.

[43] H. K. Sahu, V. Jadhav, S. Sonavane, and R. K. Sharma, "Cryptanalytic Attacks on IDEA Block Cipher," Defence Science Journal, Vol. 66, No. 6, p. 582, 2016.

[44] S. J. Shepherd, "The tiny encryption algorithm," Cryptologia, Vol 31, No 3, pp. 233-245, 2007.

[45] G. Sekar, N. Mouha, V. Velichkov, and B. Preneel, "Meet-in-the-middle attacks on reduced-round XTEA," In Cryptographers' Track at the RSA Conference, Springer, Berlin, Heidelberg, pp. 250-267, 2011.

[46] A. Bogdanov, M. Wang, "Zero correlation linear cryptanalysis with reduced data complexity," In International Workshop on Fast Software Encryption, Springer, Berlin, Heidelberg, (pp. 29-48, 2012.

[47] J. Chen, M. Wang, B. Preneel, "Impossible differential cryptanalysis of the lightweight block ciphers TEA, XTEA and HIGHT," In International Conference on Cryptology in Africa, Springer, Berlin, Heidelberg, pp. 117-137, 2012.

[48] S. A. Azimi, S. Ahmadi, Z. Ahmadian, J. Mohajeri, M. R. Aref, "Improved impossible differential and biclique cryptanalysis of HIGHT," International Journal of Communication Systems, Vol. 31, No.1, p. e3382, 2018.

[49] O. Özen, K. VarIcI, C. Tezcan, Ç. Kocair, "Lightweight block ciphers revisited: Cryptanalysis of reduced round PRESENT and HIGHT," In Australasian Conference on Information Security and Privacy, Springer, Berlin, Heidelberg, pp. 90-107, 2009.

[50] D. Hong, B. Koo, D. Kwon, "Biclique attack on the full HIGHT,"In International Conference on Information Security and Cryptology, Springer, Berlin, Heidelberg, pp. 365-374, 2011.

[51] I. Dinur, "Improved differential cryptanalysis of round-reduced speck," In International Conference on Selected Areas in Cryptography, Springer, Cham, pp. 147-164, 2014.

[52] L. Song, Z. Huang, Q. Yang, "Automatic differential analysis of ARX block ciphers with application to SPECK and LEA," In Australasian Conference on Information Security and Privacy, Springer, Cham, pp. 379-394, 2016.

[53] P. Derbez, P. A. Fouque, J. Jean, "Improved key recovery attacks on reduced-round AES in the single-key setting," In Annual International Conference on the Theory and Applications of Cryptographic Techniques, Springer, Berlin, Heidelberg, pp. 371-387, 2013.

[54] K. Fu, M. Wang, Y. Guo, S. Sun, L. Hu, "MILP-based automatic search algorithms for differential and linear trails for speck," In International Conference on Fast Software Encryption, Springer, Berlin, Heidelberg, 2016.

[55] Y. Liu, Q. Wang, V. Rijmen, "Automatic search of linear trails in ARX with applications to SPECK and Chaskey," In International Conference on Applied Cryptography and Network Security, Springer, Cham, pp. 485-499, 2016.

[56] H. Lee, H. Kang, D. Hong, J. Sung, S. Hong, "New Impossible Differential Characteristic of SPECK-64 using MILP," IACR Cryptology ePrint Archive, p.1137, 2016.

[57] K. Zhang, J. Guan, B. Hu, "Zero correlation linear cryptanalysis on LEA family ciphers," Journal of Communications, Vol. 11, No. 7, pp. 677–685, 2016.

[58] R. Ankele, E. List, "Differential cryptanalysis of round-reduced Sparx-64/128," In International Conference on Applied Cryptography and Network Security, Springer, Cham, pp. 459-475, 2018.

[59] R. Ankele, S. Kölbl, "Mind the Gap - A Closer Look at the Security of Block Ciphers against Differential Cryptanalysis," In International Conference on Selected Areas in Cryptography, Springer, Cham, pp. 163-190, 2018.

[60] M. Tolba, A. Abdelkhalek, A. M. Youssef, "Multidimensional zero-correlation linear cryptanalysis of reduced round SPARX-128," In International Conference on Selected Areas in Cryptography, Springer, Cham, pp. 423-441, 2017.

[61] M. Elsheikh, M. Tolba, A. M. Youssef, "Impossible Differential Attack on Reduced Round SPARX-128/256," IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, Vol. 101, No. 4, pp. 731-733. 2018.

[62] E. Biham, A. Shamir, "Differential cryptanalysis of DES-like cryptosystems," Journal of CRYPTOLOGY, Vol 4, No. 1, pp. 3-72, 1991.

[63] X. Wang, X. Lai, D. Feng, H. Chen, X. Yu, "Cryptanalysis of the Hash Functions MD4 and RIPEMD," In Annual international conference on the theory and applications of cryptographic techniques, Springer, Berlin, Heidelberg, pp. 1-18, 2005.

[64] X. Wang, Y. L. Yin, H. Yu, "Finding Collisions in the Full SHA-1," In Annual international cryptology conference, Springer, Berlin, Heidelberg, pp. 17-36, 2005.
[65] X. Wang, H. Yu, "How to break MD5 and other hash functions," In Annual international conference on the theory and applications of cryptographic techniques, Springer, Berlin, Heidelberg, pp. 19-35, 2005.
[66] G. Leurent, "Construction of differential characteristics in ARX designs application to Skein," In Annual Cryptology Conference, Springer, Berlin, Heidelberg, pp. 241-258, 2013.
[67] A. Biryukov and V. Velichkov, "Automatic Search for Differential Trails in ARX Ciphers," In International Workshop on Fast Software Encryption, Springer, Berlin, Heidelberg, pp. 546-570, 2014.
[68] A. Biryukov, A. Roy, and V. Velichkov, "Differential analysis of block ciphers Simon and Speck," In International Workshop on Fast Software Encryption, pp. 546-570. Springer, Berlin, Heidelberg, 2014.
[69] Y. Yao, B. Zhang, and W. Wu, "Automatic search for linear trails of the SPECK family," In International Conference on Information Security, Springer, Cham, pp. 158-176, 2015.
[70] C. Dobraunig, M. Eichlseder, and F. Mendel, "Heuristic tool for linear cryptanalysis with applications to CAESAR candidates," In International Conference on the Theory and Application of Cryptology and Information Security, Springer, Berlin, Heidelberg, pp. 490-509, 2015.
[71] N. Mouha, Q. Wang, D. Gu, B. Preneel, "Differential and linear cryptanalysis using mixed-integer linear programming," In International Conference on Information Security and Cryptology, Springer, Berlin, Heidelberg, pp. 57-76, 2011.
[72] S. Sun, L. Hu, P. Wang, K. Qiao, X. Ma, L. Song, "Automatic security evaluation and (related-key) differential characteristic search: application to SIMON, PRESENT, LBlock, DES (L) and other bit-oriented block ciphers," In International Conference on the Theory and Application of Cryptology and Information Security, Springer, Berlin, Heidelberg, pp. 158-178, 2014.
[73] J. P. Aumasson, P. Jovanovic, S. Neves, "NORX: parallel and scalable AEAD," In European Symposium on Research in Computer Security, Springer, Cham, pp. 19-36, 2014.
[74] A. Ranea, Y. Liu, T. Ashur, "An easy-to-use tool for rotational-XOR cryptanalysis of ARX block ciphers," In Proceedings of the Romanian Academy, Series A, Vol. 18, No. 3, pp. 307-316, 2017.
[75] T. Eisenbarth, Z. Gong, T. Güneysu, S. Heyse, S. Indesteege, S. Kerckhof, F. Koeune, T. Nad, T. Plos, F. Regazzoni, F. X. Standaert, "Compact implementation and performance evaluation of block ciphers in ATtiny devices," In International Conference on Cryptology in Africa, Springer, Berlin, Heidelberg, pp. 172-187, 2012.
[76] T. Eisenbarth, S. Kumar, C. Paar, A. Poschmann, L. Uhsadel, "A survey of lightweight-cryptography implementations," In IEEE Design & Test of Computers, Vol. 24, No. 6, pp. 522-533, 2007.
[77] T. Plos, H. Groß, M. Feldhofer, "Implementation of symmetric algorithms on a synthesizable 8-bit microcontroller targeting passive RFID tags," In International Workshop on Selected Areas in Cryptography, Springer, Berlin, Heidelberg, pp. 114-129, 2010.
[78] Z. Gong, S. Nikova, Y. W. Law, "KLEIN: a new family of lightweight block ciphers," In International Workshop on Radio Frequency Identification: Security and Privacy Issues, Springer, Berlin, Heidelberg, pp. 1-18, 2011.
[79] D. Dinu, Y. Le. Corre, D. Khovratovich, L. Perrin, J. Großschädl, A. Biryukov, "Triathlon of lightweight block ciphers for the internet of things," In Journal of Cryptographic Engineering, Vol. 9, No. 3, pp. 283-302, 2019.
[80] M. Cazorla, K. Marquet, M. Minier, "Survey and benchmark of lightweight block ciphers for wireless sensor networks," In 2013 International Conference on Security and Cryptography (SECRYPT), IEEE, pp. 1-6, 2013.
[81] J. P. Kaps, "Chai-tea, cryptographic hardware implementations of xtea," In International Conference on Cryptology in India, Springer, Berlin, Heidelberg, pp. 363-375, 2008.
[82] D. Lee, D. C. Kim, D. Kwon, H. Kim, "Efficient hardware implementation of the lightweight block encryption algorithm LEA," In Sensors, Vol. 14, No. 1, pp. 975-994, 2014.
[83] P. Kitsos, N. Sklavos, M. Parousi, A. N. Skodras, "A comparative study of hardware architectures for lightweight block ciphers," In Computers & Electrical Engineering, Vol. 38, No. 1, pp. 148-160, 2012.
[84] P. Israsena, S. Wongnamkum, "Hardware implementation of a TEA-based lightweight encryption for RFID security," In RFID Security, Springer, Boston, MA, pp. 417-433, 2008.
[85] T. Plos, C. Dobraunig, M. Hofinger, A. Oprisnik, C. Wiesmeier, J. Wiesmeier, "Compact hardware implementations of the block ciphers mCrypton, NOEKEON, and SEA," In International Conference on Cryptology in India, Springer, Berlin, Heidelberg, pp. 358-377, 2012.
[86] Y. I. Lim, J. H. Lee, Y. You, K. R. Cho, "Implementation of HIGHT cryptic circuit for RFID tag," In IEICE Electronics Express, Vol. 6, No. 4, pp. 180-186, 2009.
[87] A. Amiruddin, A.A.P Ratna, R.F. Sari, "Systematic review of internet of things security," International Journal of Communication Networks and Information Security, Vol. 11. No. 2, pp. 248-255, 2019.

Table 4. Software Implementation of ARX block ciphers.

Cipher	Key Size (bits)	Block size (bits)	ROM	RAM	Latency	Energy	Throughput at 4 MHz	Efficiency
8-bit microcontrollers								
IDEA [75]	128	64	836	232	8250/22792	34.3	31/11	37.08
IDEA [75]	128	64	596	0	2700/15393	10.8	94.8/16	159.06
TEA[75]	128	64	648	24	7408 / 7539	30.3	34.5 / 33.9	53.24
TEA[76]	128	64	1140	0	6271 / 6299	34.3	40.8 / 40.6	35.78
XTEA[77]	128	64	504	-	17514 / 19936	70.0	14.6 / 12.8	28.96
XTEA[77]	128	64	820	-	7786 / 8928	31.1	32.8 / 28.6	40.00
XTEA[77]	128	64	1246	-	7595 / 8735	30.3	33.7 / 29.3	27.04
SPECK [12]	128	128	396	0	1333 / -	5.3	384 / -	969.69
SPECK [12]	128	64	186	0	599 / -	2.3	427.5 / -	2298.38
HIGHT [76]	128	64	5672	0	2964 / 2964	11.8	86.3 / 86.3	15.21
HIGHT [78]	128	64	2510	117	7377 / 5844	29.5	34.7 / 43.8	13.82
HIGHT[79]	128	64	2608	342	87694 / 83464	350.7	2.91 /2.44	1.11
HIGHT [79]	128	64	1084	54	11399 / -	45.5	22.45 / -	20.71
HIGHT [79]	128	64	5718	47	6377/ -	25.5	40.14 /-	7.01
SPECK [12]	96	96	276	0	887 / -	3.5	433 / -	1568.84
SPECK[12]	96	64	152	108	1232/-	4.9	207.8/-	1367.10
SPECK [12]	96	64	182	0	577/-	2.3	444/-	2439.56
SPECK [79]	96	64	1692	300	121953 / 224635	487.8	2.09 / 1.13	1.23
SPECK [79]	96	64	572	49	14003 / -	56.0	18.28 / -	31.95
SPECK [15]	96	48	134	0	408 / -	1.6	470.5 / -	3511.19
16-bit microcontrollers								
IDEA [80]	128	64	3140	82	31402/163380	42.3	8.1/1.5	2.57
HIGHT [80]	128	64	3130	18	32372 / 32623	43.7	7.9 / 7.8	2.52
HIGHT [78]	128	64	2050	40	8620 / 8620	11.6	29.7 / 29.7	14.48
HIGHT [79]	128	64	2368	342	215107 / 209838	290.3	1.19 / 1.21	0.50
HIGHT [79]	128	64	980	62	26728 / -	36.0	9.57 / -	9.76
HIGHT [79]	128	64	13780	64	21882 / -	29.5	11.69 / -	0.84
SPECK[79]	96	64	1342	300	51621 / 45248	69.6	4.95 / 5.65	3.68
SPECK [79]	96	64	618	58	6054 / -	8.1	42.28 / -	68.41
32-bit microcontrollers								
LEA [13]	128	128	590	32	5321/-	-	97.8 / -	165.76
HIGHT [79]	128	64	2196	392	83157 / 91929	-	3.07 / 2.78	1.39
HIGHT [79]	128	64	1008	128	11602 /	-	22.06 / -	21.88
SPECK [79]	96	64	792	332	7665 / 12513	-	33.39 / 20.45	42.15
SPECK [79]	96	64	512	96	904 / -	-	283.18 / -	553.08

Table 5. Hardware implementations of ARX block cipher

Cipher	Key Size (bits)	Block size (bits)	Latency (Cycles/block)	Throughput at 100 KHz (Kbps)	Area (GE)	Efficiency (Kbps/KGE)	Power μW	Energy $\frac{\mu J}{(\text{bit}})$
0.18 μm technology								
TEA [17]	128	64	64	100	2355	42.46	3.53	35.32
0.13 μm technology								
XTEA [81]	128	64	32	200	2521	79.33	2.52	12.60
SPECK[12]	128	128	1058	12.1	1396	8.66	1.40	115.39
SPECK[12]	128	64	464	13.8	1127	12.24	1.12	81.70
LEA [82]	128	128	168	76.19	3826	19.91	3.82	50.22
LEA [82]	128	128	96	133.3	4296	31.02	4.30	32.22
LEA [82]	128	128	24	533.3	5426	98.28	5.42	10,17
SPECK [12]	96	48	400	12	884	13.57	0.88	73.67
SPECK [12]	96	64	441	14.5	984	14.73	0.98	67.80
SPECK [12]	96	64	29	220.7	1522	145.00	1.52	6.89

SPECK [12]	96	96	696	13.8	1134	12.16	1.13	82.22
CHAM [16]	64	128	-		665	-		
CHAM [16]	128	128	-		1,057			
CHAM [16]	128	256	-		1,180			
0.09 μm technology								
XTEA [83]	128	64	32	200	3490	57.30	2.44	12.21
0.35 μm technology								
TEA [84]	128	64	512	6.25	3872	1.61	7.00	557.56
XTEA [85]	128	64	705	9.08	2636	3.44	4.74	522.66
HIGHT [11]	128	64	34	188	3048	61.67	5.48	29.14
0.25 μm technology								
HIGHT(D) [86]	128	64	34	188	2608	72.08	4.70	24.93