Journal of Information Assurance and Security.
ISSN 1554-1010 Volume 7 (2012) pp. 52-60

AES: Current Security and Efficiency Analysis of its Alternatives

Herman Isa, Iskandar Bahari and Muhammad Reza Z'aba

Cryptography Lab, Advanced Analysis and Modeling (ADAM) Cluster,
MIMOS Berhad, Kuala Lumpur, Malaysia
herman.isa@mimos.my, iskandar.bahari@mimos.my, reza.zaba@mimos.my

Abstract: **The Advanced Encryption Standard (AES) has been in existence over the last 11 years. It was widely accepted as the de facto standard in many security-related applications such as SSL/TLS, Microsoft BitLocker Drive Encryption, Skype and many others. Recently in 2011, the AES was claimed to be theoretically broken in the single-key attack model using a new technique called biclique. Just two years before, in 2009, the AES with 192- and 256- bit keys were found to be theoretically broken in the related-key attack model. This paper reviews existing attacks on the AES and evaluates the efficiency of recent block cipher proposals as alternatives to the AES. These block ciphers were proposed to patch the AES against the related-key type of attack.**

Keywords: Cryptography, cryptanalysis, symmetric cipher, block cipher, efficiency

I. Introduction

The Advanced Encryption Standard (AES) [13] is a standard block cipher adopted by the National Institute of Standards and Technology (NIST) of the United States (US). It was defined as the Federal Information Processing Standards Publications (FIPS PUB) 197 [27]. The AES is also included as part of the International Organization for Standardization / International Electrotechnical Commission (ISO/IEC) standard for encryption algorithms [20]. The wide acceptance of the AES shows that the cipher is central in providing confidentiality of information.

The process of selecting the AES began in 1997. During this time, NIST issued an open call to select the AES to replace the then 20-year old Data Encryption Standard (DES) [26]. Fifteen candidates were accepted for evaluation. After much public scrutiny, Rijndael [12] was eventually selected as the AES in 2000. In 2003, the scope of the AES was broadened by the US government to include protection of classified information up to SECRET and TOP SECRET levels [11].

This paper reviews existing attacks on the AES and evaluates the efficiency of recently proposed block ciphers as alternatives to the AES[1]. These block ciphers were proposed to counter the AES against related-key type of attack. Furthermore, all the proposed block ciphers are identical except in the key schedule. In particular, the related-key type of attacks manage to theoretically break AES with 192- and 256-bit keys [5, 6, 7]. Another very recent attack, the biclique [8], is claimed to theoretically break the AES in the single-key attack model. Since the attack is very new, there are currently no proposals that modify the AES to counter this attack. Therefore, we do not review proposals that counter this attack in this paper.

This paper is organized as follows. Section II gives a brief description of the AES. Existing attacks on the AES are reviewed in Section III. Section IV discusses existing solutions to counter recent attacks on the AES. Section V explained about the efficiency analysis of each proposals and Section VI concludes the paper.

[1] This is a revised and extended version of paper presented in Information and Assurance Security (IAS) 2011[19]. We corrected some minor mistakes found in its efficiency analysis.

II. Description of the AES

The AES accepts 128 bits of plaintext and master key blocks of size 128, 192 or 256 bits. Let us denote the AES with these different key sizes as AES-128, AES-192 and AES-256, respectively. The 128-bit ciphertext block is produced after the plaintext block is processed by the round function a number of times. This number is 10, 12 and 14 for AES-128, AES-192 and AES-256, respectively. The plaintext, ciphertext and intermediate state blocks can be depicted as two-dimensional rectangular array of bytes with dimension 4×4. The master key can also be represented in this form but the number of rows is fixed to four. The number of columns equals the key length divided by 32.

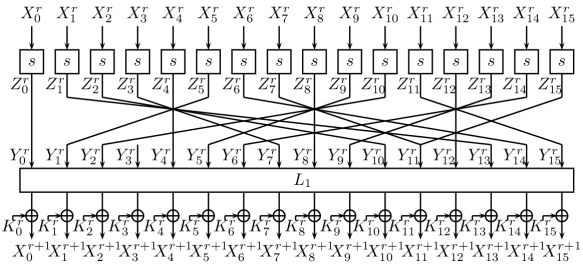

Figure. 1: Round function of the AES in Round r

A. Encryption Algorithm

The plaintext block $P = X_0^0|X_1^0|\ldots|X_{15}^0$ is formed from the concatenation of sixteen 8-bit words X_i^0. Let K^r de-

note the 128-bit subkey in round r derived from the master key K. The derivation of these subkeys is explained later in Section II-B. The round function is composed of a non-linear transformation S, linear transformations L_0 and L_1, and a key mixing transformation. These transformations refer to `SubBytes`, `ShiftRows` and `MixColumns`, respectively. The interested reader is referred to Daemen and Rijmen [12, 13] for a detailed treatment of these transformations. The encryption algorithm of the AES can be expressed by the following equations:

$$\begin{aligned} X^1 &= P \oplus K^0 \\ X^{r+1} &= L_1(L_0(S(X^r))) \oplus K^r, \quad r = 1, 2, \ldots, N_R - 1 \\ C &= L_0(S(X^{N_R})) \oplus K^{N_R} \end{aligned}$$

where $N_R \in \{10, 12, 14\}$. It can be observed that there is an additional key mixing transformation before the first round, and that the transformation L_1 is omitted in the last round. The round function of the AES is depicted in Figure 1.

Input: Master key $\hat{K}$ and array $W[4][4(N_R + 1)]$;
1 **for** $i = 0$ **to** *3* **do**
2 **for** $j = 0$ **to** $N_k - 1$ **do**
3 $W[i][j] = \hat{K}[i][j]$;
4 **end**
5 **end**
6 **for** $j = N_k$ **to** $4(N_R + 1) - 1$ **do**
7 **if** j *mod* $N_k == 0$ **then**
8 $W[0][j] = W[0][j - N_k] \oplus s(W[1][j - 1]) \oplus RC^{j/N_k}$;
9 **for** $i = 1$ **to** *3* **do**
10 $W[i][j] = W[i][j-N_k] \oplus s(W[(i+1) \bmod 4][j-1])$;
11 **end**
12 **else if** *(*j *mod* $N_k == 4$*) AND (AES-256)* **then**
13 **for** $i = 0$ **to** *3* **do**
14 $W[i][j] = W[i][j-N_k] \oplus s(W[(i+1) \bmod 4][j-1])$;
15 **end**
16 **else**
17 **for** $i = 0$ **to** *3* **do**
18 $W[i][j] = W[i][j - N_k] \oplus W[i][j - 1]$;
19 **end**
20 **end**
21 **end**
Output: Array W;

Algorithm 1: Key schedule for the AES.

B. Key schedule

The master key $K = K_0|K_1|\ldots|K_{m_k-1}$ is formed from the concatenation of m_k 8-bit words. The value m_k is 16 for AES-128, $m_k = 24$ for AES-192 and $m_k = 32$ for AES-256 key. The key schedule is described as follows. First, the master key K is put into a two-dimensional array $\hat{K}[\cdot][\cdot]$ consisting of 4 rows and $N_k = m_k/4$ columns. The setup is performed as follows: $\hat{K}[i \bmod 4][\lfloor i/4 \rfloor] = K_i$ for $i = 0, 1, \ldots, N_k - 1$. Next, an array $W[\cdot][\cdot]$ consisting of 4 rows and $4(N_R + 1)$ columns is initialized. The key array $\hat{K}$ is expanded into the array W by Algorithm 1. Then, byte i of the 128-bit subkey in round r, denoted K_i^r, is derived as follows.

$$K_i^r = W[i \bmod 4][4r + \lfloor i/4 \rfloor] \quad i = 0, 1, \ldots, 15.$$

Figure 2 illustrate the operations that occurs in Algorithm 1.

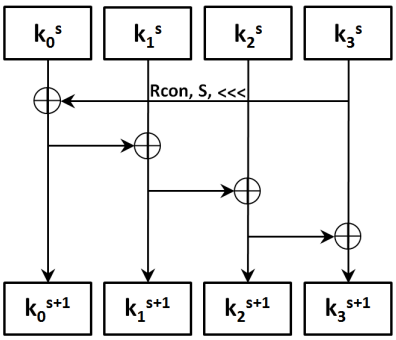

Figure. 2: The key schedule of AES

III. Attacks on the AES

As a standard, the AES block cipher has been subjected to great public scrutiny, resulting in a number of attacks.
Some of these attacks are given in Table 1. The table is divided into three main rows. The first main row contains attacks for the AES-128, the second main row lists attacks for AES-192 and the last main row shows attacks for the AES-256. Previous cryptanalysis on the AES can be categorized into single- and related- (or multiple-) key attacks.
In the single-key scenario, several attacks have been mounted on the AES-128 reduced to five [3], six [9, 3, 15], seven [18, 32, 23] and the full 10 rounds [8]. For AES-192, there are attacks on seven [24, 18, 15, 29, 32, 14, 23], eight [15] and the full 12 rounds [8]. For AES-256, there are attacks on seven [29, 32, 14, 23], eight [15, 32, 14, 23] and the full 14 rounds [8]. The biclique [8], which is based on the meet-in-the-middle attack, is currently the best[2] attack on all key-size variants of the AES. However, the biclique attack is still a theoretical approach since its computational complexity is very large (Refer to Table 1).
In the related-key scenario, there has been a significant progress in the cryptanalysis of both the AES-192 and AES-256. This is due to the slower diffusion property in the key schedules for these variants compared to the AES-128. For the 12-round of AES-192, related-key type attacks manage to penetrate seven [21], eight [21, 33, 2], nine [1] and ten [22] rounds. Similarly, for the 14-round of AES-256, related-key attacks can be launched on nine [15, 16, 4] and ten [1, 22, 16, 4] rounds. Recently, Biryukov et al. demonstrated attacks on the full AES-192 and AES-256 using related-key amplified boomerang and boomerang, respectively [5, 6]. Both attacks use only four related keys and are the best existing attacks on AES-192 and AES-256 (in the related-key attack model).

IV. How to Counter the Related-Key Attacks

As mentioned in Section III, there are many cryptanalytic attacks against the AES. It can be noted that related-key type of attacks are particularly effective in attacking AES-192 and AES-256. This is mainly due to the lack of nonlinearity in the key schedule of the AES. Therefore, a few alternative key schedules have been proposed to counter the related-key

[2]Here, the best attack refers to an attack that manages to penetrate the most number of block cipher rounds.

Table 1: Summary of existing attacks on the AES

AES	Round	Attack	# of keys	Complexity		
				Data	Time	Memory
128	6	Partial sums [15]	1	$6 \cdot 2^{32}$CP	2^{44}	n/a
	7	Collision [18]	1	2^{32}CP	2^{128}	2^{80}
	7	Partial sums [15]	1	$2^{127.9}$CP	2^{120}	n/a
	7	Imp. diff. [23]	1	$2^{112.2}$CP	$2^{117.2}$	n/a
	8	Biclique [8]	1	$2^{126.33}$CP	$2^{124.97}$	2^{102}
	8	Biclique [8]	1	2^{127}CP	$2^{125.64}$	2^{32}
	8	Biclique [8]	1	2^{88}CP	$2^{125.34}$	2^{8}
	10	Biclique [8]	1	2^{88}CP	$2^{126.18}$	2^{8}
192	7	Collision [18]	1	2^{32}CP	2^{140}	2^{84}
	7	Integral [24]	1	2^{32}CP	2^{184}	2^{32}
	7	MitM [14]	1	2^{32}CP	2^{72}	2^{206}
	7	Imp. diff. [23]	1	$2^{113.8}$CP	$2^{118.8}$	n/a
	7	Imp. diff. [23]	1	$2^{91.2}$CP	$2^{139.2}$	n/a
	8	Partial sums [15]	1	$2^{127.9}$CP	2^{188}	2^{64}
	8	RK imp. diff. [21]	2	2^{88}RK-CP	2^{183}	n/a
	8	RK imp. diff. [33]	2	$2^{64.5}$RK-CP	2^{177}	2^{69}
	8	RK imp. diff. [33]	2	2^{88}RK-CP	2^{153}	n/a
	8	RK imp. diff. [33]	2	2^{112}RK-CP	2^{136}	n/a
	8	RK diff.-lin. [34]	2	2^{118}RK-CP	2^{165}	n/a
	9	Biclique [8]	1	2^{80}CP	$2^{188.8}$	2^{8}
	10	RK rectangle [22]	64	2^{124}RK-CP	2^{183}	n/a
	12	RK amp. boom. [5, 6]	4	2^{123}RK-CP	2^{176}	2^{152}
	12	Biclique [8]	1	2^{80}CP	$2^{189.74}$	2^{8}
256	7	Collision [18]	1	2^{32}CP	2^{140}	2^{84}
	7	Integral [24]	1	2^{32}CP	2^{200}	2^{32}
	8	Imp. diff. [23]	1	$2^{111.1}$CP	$2^{227.8}$	n/a
	8	Imp. diff. [23]	1	$2^{89.1}$CP	$2^{229.7}$	n/a
	8	Partial sums [15]	1	$2^{127.9}$CP	2^{204}	2^{104}
	8	MitM [14]	1	2^{32}CP	2^{200}	2^{206}
	9	RK [15]	256	2^{85}RK-CP	2^{224}	2^{32}
	9	Biclique [8]	1	2^{120}CP	$2^{253.1}$	2^{8}
	9	Biclique [8]	1	2^{120}CP	$2^{251.92}$	2^{8}
	10	RK rectangle [22]	64	$2^{113.9}$RK-CP	$2^{172.8}$	n/a
	14	RK differential [7]	2^{35}	2^{131}RK-CP	2^{131}	2^{65}
	14	RK boomerang [5, 6]	4	$2^{99.5}$RK-CP	$2^{99.5}$	2^{77}
	14	Biclique [8]	1	2^{40}CP	$2^{254.42}$	2^{8}

attacks. In this section, we review these proposals.

A. Proposal 1: May et al. (meAES)

In 2002, May et al. propose an alternative to the key schedule of the AES [25]. Let us denote this proposal as *meAES*. The objectives are to meet several desired properties for the key schedule such as collision-resistant one-way function, minimal mutual information and efficient implementation. In particular, the first property is to attain round key *irreversibility*. This means that given any subset of the round subkeys, it is hard to derive the remaining round subkeys. Note that meAES was proposed before the latest related-key attacks that managed to theoretically break the full AES-192 and AES-256 in 2009 [5, 6].

The meAES key schedule is given in Algorithm 2. Let $a = a_0|a_1|\dots|a_{15}$ and $b = b_0|b_1|\dots|b_{15}$ denote 128-bit values derived from the master key[3] $K = K_0|K_1|\dots|K_{m_k-1}$. Each 128-bit round subkey K_r is obtained after the execution of three rounds of the AES round function, (i.e. `SubBytes`, `ShiftRows`, `MixColumns` and `AddRoundKey`), using a and b as inputs. As shown in Figure 3, all these additional operations compared to the original AES Key Schedule were being highlighted.

In [25], the authors conducted statistical tests to show that their proposed key schedule does not contain any bit leakage between round subkeys and achieved bit confusion and diffusion by satisfying frequency and Strict Avalanche Criterion (SAC) tests.

[3] In May et al., the master key is denoted as MK [25].

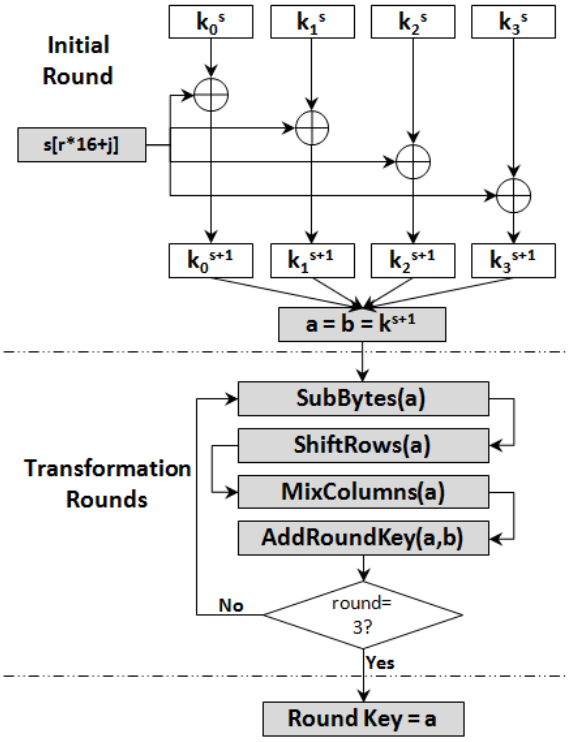

Figure. 3: The key schedule of meAES

for $r = 0$ **to** N_R **do**
  for $j = 0$ **to** 15 **do**
    if *AES-128* **then**
      $a_j = b_j = K_j \oplus S[r * 16 + j]$;
    else if *AES-192* **then**
      $a_j = K_j \oplus S[r * 16 + j] \oplus S[K_{j+8}]$;
      $b_j = K_{j+8} \oplus S[r * 16 + j] \oplus S[K_j]$;
    else if *AES-256* **then**
      $a_j = K_j \oplus S[r * 16 + j] \oplus S[K_{j+16}]$;
      $b_j = K_{j+16} \oplus S[r * 16 + j] \oplus S[K_j]$;
    end
  end
  for $i = 0$ **to** 2 **do**
    `SubBytes`(a);
    `ShiftRows`(a);
    `MixColumns`(a);
    `AddRoundKey`(a, b);
  end
  $K^r = a$
end

Algorithm 2: meAES Key Schedule.

B. Proposal 2: Nikolić (xAES)

In 2010, Nikolić proposes a new key schedule called xAES [28]. The xAES is identical to the AES except for the key schedule. Let xAES-128, xAES-192 and xAES-256 denote xAES with 128-, 192- and 256-bit key, respectively. In the original key schedule of AES (refer to Section II-B), to obtain the values of a column in the W array, if the column index is a multiple of N_k, then the previous column is first rotated one byte up. In xAES, the rotation is done for all columns as

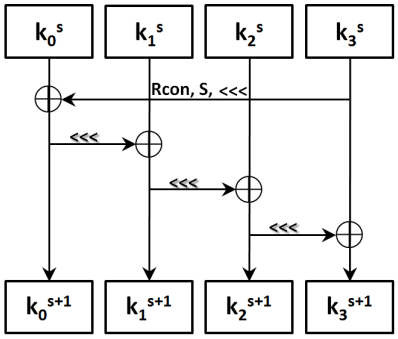

Figure. 4: The key schedule of xAES

being illustrate in Figure 4. Furthermore, xAES has additional application of S-boxes for the key schedule of xAES-192. Nikolić claims that full round of all xAES variants have resistance against related-key differential attacks and all fixed-key attacks. Algorithm 3 outlines the key schedule for xAES.

Input: Master key $\hat{K}$ and array $W[4][4(N_R+1)]$;
for $i = 0$ **to** *3* **do**
  for $j = 0$ **to** $N_k - 1$ **do**
    $W[i][j] = \hat{K}[i][j]$;
  end
end
for $j = N_k$ **to** $4(N_R+1) - 1$ **do**
  if j *mod* $N_k == 0$ **then**
    $W[0][j] = W[0][j-N_k] \oplus s(W[1][j-1]) \oplus RC^{j/N_k}$;
    for $i = 1$ **to** *3* **do**
      $W[i][j] = W[i][j-N_k] \oplus s(W[(i+1) \bmod 4][j-1])$;
    end
  else if *(*j *mod* $N_k == N_k/2$*) AND (xAES-192 OR xAES-256)* **then**
    for $i = 0$ **to** *3* **do**
      $W[i][j] = W[i][j-N_k] \oplus s(W[(i+1) \bmod 4][j-1])$;
    end
  else
    for $i = 0$ **to** *3* **do**
      $W[i][j] = W[i][j-N_k] \oplus W[(i+1) \bmod 4][j-1]$;
    end
  end
end
Output: Array W;

Algorithm 3: The xAES Key Schedule.

C. Proposal 3: Improved May et al.'s (imeAES)

In 2011, Choy et al. [10] proposed an improved version of the meAES key schedule. Let us denote this as *imeAES*. The improvement is done to patch a minor weakness found in the latter's key schedule.

Choy et al. show that the meAES key schedule has equivalent keys that produce the same encryption output. The weakness is due to the initialization of a and b. In May et al.'s key schedule, an adversary is able to force a and b to have zero differential by choosing an appropriate pair of related secret key [10].

Choy et al.'s improved key schedule is given in Algorithm 4. The designers modify the initialization of a and b such that each byte of a and b only depends on one byte instead of two bytes from the secret key. They also introduce key-length-dependent constant $keylen$ to defend against the related-cipher attack [31]. Here, $keylen$ denotes the key length of

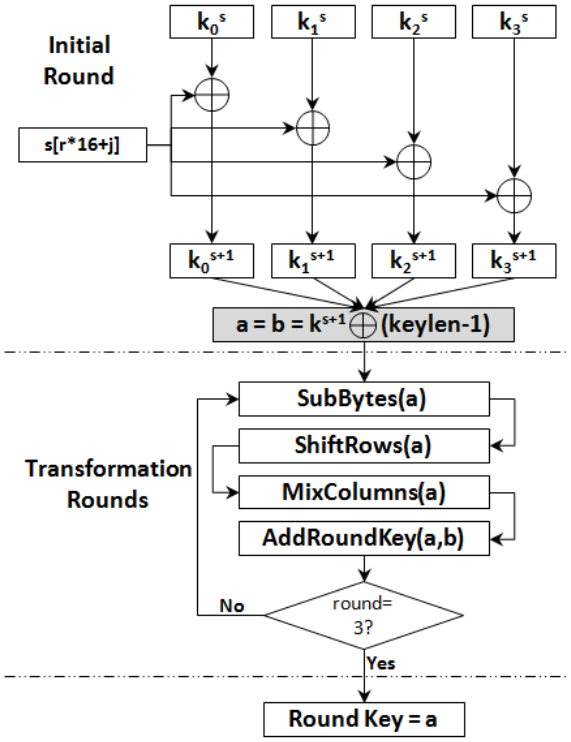

Figure. 5: The key schedule of improved meAES

the cipher encoded as byte. Same as the original version by May et al, this improved key schedule also has the property of round key irreversibility. Since the structure is same as meAES, Figure 5 highlight the part that was being improved for imeAES key schedule.

for $r = 0$ **to** N_R **do**
  for $j = 0$ **to** 15 **do**
    if *AES-128* **then**
      $a_j = b_j = K_j \oplus S[r*16+j] \oplus (keylen-1)$;
    else if *AES-192* **then**
      $a_j = K_j \oplus S[r*16+j] \oplus (keylen-1)$;
      $b_j = K_{j+8} \oplus S[r*16+j] \oplus (keylen-1)$;
    else if *AES-256* **then**
      $a_j = S[K_j] \oplus S[r*16+j] \oplus (keylen-1)$;
      $b_j = S[K_{j+16}] \oplus S[r*16+j] \oplus (keylen-1)$;
    end
  end
  for $i = 0$ **to** 2 **do**
    `SubBytes`(a);
    `ShiftRows`(a);
    `MixColumns`(a);
    `AddRoundKey`(a,b);
  end
  $K^r = a$
end

Algorithm 4: The imeAES Key Schedule

D. Proposal 4: Choy et al. (ceAES)

Apart from improving the meAES key schedule, Choy et al. also propose a new key schedule for the AES [10]. Let us denote this proposal as *ceAES*. Similar to May et al.'s proposal, Choy et al.'s key schedule uses transformation from the AES round function. See Figure 6 for the illustration. The proposal only provides partial round key irreversibility. This is

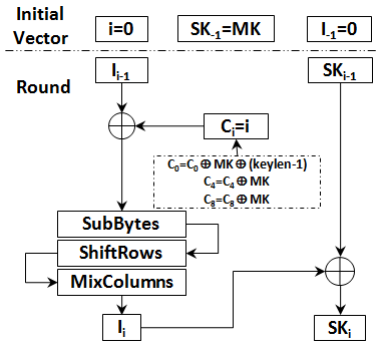

Figure. 6: The key schedule of ceAES

because, given some combinations of three or more round subkeys, it may be possible to obtain the remaining round subkeys. If, however, given two round subkeys, it is hard to obtain the remaining round subkeys.

The proposed key schedules of Choy et al. are given in Algorithm 5 and Algorithm 6 for 128-, and 256- and 192-bit keys, respectively. Note that all values for K_i where $i < 0$ are discarded. Furthermore, both C_j and $keylen$ are encoded as 128-bit strings.

V. Efficiency Analysis

In this section, we measure the efficiency (speed) of the four proposed key schedules. The focus is only on the key schedule and we omit the efficiency of encryption. This is because in all the proposals, the encryption and decryption algorithms are the same as the AES. In AES, the operations used on **bytes** include XOR, rotation, multiplication and table lookup.

Let us assume that the cost of performing an XOR operation between two bytes is the same as the cost of rotating one byte. Furthermore, let us denote these operations as **O**, multiplication as **M** and table lookup as **L**. Therefore, the cost of performing an XOR between two bytes, or rotating one byte, equals 1**O**. The cost of performing a table lookup for one byte is 1**L** and the cost of multiplying two bytes is 1**M**.

Using the previous notation, the cost of performing the `SubBytes` transformation is 16**L** because one has to perform 16 table lookups which correspond to the 16 S-boxes. Each S-box is assumed to be implemented using a table lookup of 256 entries.

Recall from Section II that the AES intermediate state block can be represented as a two-dimensional array of bytes with dimension 4×4. In `ShiftRows`, the second row of the state block is rotated by one byte (1**O**), the third row by two bytes (2**O**) and the last row by 3 bytes (3**O**). Therefore, `ShiftRows` costs $1 + 2 + 3 = 6$**O**.

The `MixColumns` transformation involves the application of a 4×4 Maximal Distance Separable (MDS) matrix to four column vectors of the AES intermediate state block. The application of the matrix to one column vector involves $4 \times 4 = 16$ byte multiplications and $3 \times 4 = 12$ XORs (between two bytes). The AES intermediate state block has four column vectors. Therefore, `MixColumns` costs $16 \times 4 = 64$**M** and $12 \times 4 = 48$**O**.

The `AddRoundKey` transformation performs XOR of all the 16 bytes of the AES intermediate state block with the 16 bytes of the 128-bit round subkey. Therefore, there are 16 XOR operations and `AddRoundKey` costs 16**O**.

One round of the AES therefore costs $6 + 48 + 16 = 70$**O**, 64**M** and 16**L**. The efficiency figures of all the four proposed key schedules are compiled in Table 2. The table illustrates the efficiency of the key schedule using 128, 192 and 256-bit keys. The subsequent sections describe how the numbers are obtained.

for $j = 0$ **to** 11 **do**
  $C_j = j$
end
$C_0 = C_0 \oplus K \oplus (keylen - 1)$;
$C_4 = C_4 \oplus K$;
$C_8 = C_8 \oplus K$;
$SK_{-1} = K$;
$I_{-1} = 0$;
for $i = 0$ **to** 11 **do**
  $I_i = I_{i-1} \oplus C_i$;
  `SubBytes`(I_i);
  `ShiftRows`(I_i);
  `MixColumns`(I_i);
  $SK_i = I_i \oplus SK_{i-1}$;
  $K_{i-1} = SK_i$;
end

Algorithm 5: The ceAES Key Schedule for 128-bit keys

if *AES-192* **then** $f = 1$;
if *AES-256* **then** $f = 2$;
for $j = 0$ **to** 15 **do**
  $K1_j = K_j$;
  $K2_j = K_{j+8*f}$
end
for $j = 0$ **to** 15 **do**
  $C_j = j$
end
$C_0 = C_0 \oplus K1 \oplus (keylen - 1)$;
$C_4 = C_4 \oplus K2$;
$C_8 = C_8 \oplus K1$;
$C_{12} = C_{12} \oplus K2$;
$SK_{-1} = K1, I_{-1} = 0$;
for $i = 0$ **to** 15 **do**
  $I_i = I_{i-1} \oplus C_i$;
  `SubBytes`(I_i);
  `ShiftRows`(I_i);
  `MixColumns`(I_i);
  $SK_i = I_i \oplus SK_{i-1}$;
  if *AES-192* **then** $K_{i-3} = SK_i$;
  if *AES-256* **then** $K_{i-1} = SK_i$;
end

Algorithm 6: The ceAES Key Schedule for 192- and 256- bit keys

A. AES

In the original AES, the subkey generation basically involves two sets of operations, i.e. if the round subkey column is a multiple of N_k, or not. Refer to Algorithm 1.

For AES-128, if the round subkey column is a multiple of $N_k = 4$, then there are $2 + 3 = 5$ XORs, 4 rotations and 4 table lookups (9**O** + 4**L**). For the remaining three columns, there are $3 \times 4 = 12$ XORs (12**O**) and no rotations. Therefore, to produce one subkey, $9 + 12 = 21$**O** and 4**L** are re-

quired. In total, $10 \times 21 = 210\mathbf{O}$ and $10 \times 4 = 40\mathbf{L}$ are involved in the key schedule of AES-128.
For AES-192, if the round subkey column is a multiple of $N_k = 6$, then there are $2 + 3 = 5$ XORs, 4 rotations and 4 table lookups ($9\mathbf{O} + 4\mathbf{L}$). For the remaining five columns, there are $5 \times 4 = 20$ XORs ($20\mathbf{O}$). For a block of 6 columns, $9+20 = 29\mathbf{O}$ and $4\mathbf{L}$ are required. Note that the total number of columns in the array W is 52. The previous operations are performed 8 times which produce $8 \times 6 = 48$ columns. The remaining $52-48 = 6$ columns represent the first round subkey of AES-192. In total, $8\times29 = 232\mathbf{O}$ and $8\times4 = 32\mathbf{L}$ are involved in the key schedule of AES-192.
For AES-256, if the round subkey column is a multiple of $N_k = 8$, then there are $2 + 3 = 5$ XORs, 4 rotations and 4 table lookups ($9\mathbf{O}+4\mathbf{L}$). If the round subkey column's index modulo $N_k = 8$ equals 4, then there are 4 XORs ($4\mathbf{O}$) and 4 table lookups ($4\mathbf{L}$). For the remaining $8 - 2 = 6$ columns, there are $6 \times 4 = 24$ XORs ($24\mathbf{O}$). For a block of 8 columns, $9+4+24 = 37\mathbf{O}$ and $4+4 = 8\mathbf{L}$. Note that the total number of columns in the array W is 60. The previous operations are performed 7 times which produce $7 \times 8 = 56$ columns. The remaining $60-56 = 4$ columns represent the first round subkey of AES-256. In total, $7\times37 = 259\mathbf{O}$ and $7\times8 = 56\mathbf{L}$ are involved in the key schedule of AES-256.

B. meAES

For all key sizes, the generation of any round subkey values involved three iterations of the AES round function. As discussed above, one round function of AES involved $70\mathbf{O}$, $64\mathbf{M}$ and $16\mathbf{L}$. Therefore for three iterations in meAES will involves $3 \times 70 = 210\mathbf{O}$, $3 \times 64 = 192\mathbf{M}$ and $3 \times 16 = 48\mathbf{L}$. Refer to Algorithm 2.
For meAES-128, there are 16 XORs and 16 table lookups ($16\mathbf{O} + 16\mathbf{L}$). Thus, for one round subkey, $210+16 = 226\mathbf{O}$, $192\mathbf{M}$ and $48 + 16 = 64\mathbf{L}$ are involved. Therefore in total, meAES-128 which have 11 rounds will require $11 \times 226 = 2486\mathbf{O}$, $11 \times 192 = 2112\mathbf{M}$ and $11 \times 64 = 704\mathbf{L}$ for its operations.
For meAES-192, there are $2 \times 2 \times 16 = 64$ XORs and $4 \times 16 = 64$ table lookups ($64\mathbf{O} + 64\mathbf{L}$) for the initialization of a and b. Then, for one round subkey, $210 + 64 = 274\mathbf{O}$, $192\mathbf{M}$ and $48 + 64 = 112\mathbf{L}$ are involved. So in total, for the 13 rounds of meAES-192, will require $13 \times 274 = 3562\mathbf{O}$, $13 \times 192 = 2496\mathbf{M}$ and $13 \times 112 = 1456\mathbf{L}$.
The key schedule for meAES-256 has the same number of computations as for meAES-192 to produce one round subkey. Thus for total of 15 rounds, meAES-256 will require, $15 \times 274 = 4110\mathbf{O}$, $15 \times 192 = 2880\mathbf{M}$ and $15 \times 112 = 1680\mathbf{L}$.

C. xAES

The key schedule proposed for xAES is very similar to the AES. The difference is on the rotations used for every column and also for the key schedules for xAES-192. Refer to Algorithm 3.
For xAES-128, if the round subkey column is a multiple of $N_k = 4$, then there are 5 XORs, 4 rotations and 4 table lookups ($9\mathbf{O} + 4\mathbf{L}$). For the remaining three columns, there are 12 XORs and 12 rotations ($24\mathbf{O}$). Therefore, to produce one subkey, $9 + 24 = 33\mathbf{O}$ and $4\mathbf{L}$ are required. In total, $10 \times 33 = 330\mathbf{O}$ and $10 \times 4 = 40\mathbf{L}$ are involved in the key schedule of xAES-128.
For xAES-192, if the round subkey column is a multiple of $N_k = 6$, then there are 5 XORs, 4 rotations and 4 table lookups ($9\mathbf{O} + 4\mathbf{L}$). If the round subkey column's index modulo $N_k = 6$ equals 3, then there are 4 XORs, 4 rotations (4+4=8$\mathbf{O}$) and 4 table lookups ($4\mathbf{L}$). For the remaining $6 - 2 = 4$ columns, there are $4 \times 4 = 16$ XORs and 16 rotations ($32\mathbf{O}$). For a block of 6 columns, $9 + 8 + 32 = 49\mathbf{O}$ and $4 + 4 = 8\mathbf{L}$ are required. Note that the total number of columns in the array W is 52. The previous operations are performed 8 times which produce $8 \times 6 = 48$ columns. In total, $8 \times 49 = 392\mathbf{O}$ and $8 \times 8 = 64\mathbf{L}$ are involved in the key schedule of xAES-192.
For xAES-256, if the round subkey column is a multiple of $N_k = 8$, then there are 5 XORs, 4 rotations and 4 table lookups ($9\mathbf{O} + 4\mathbf{L}$). If the round subkey column's index modulo $N_k = 8$ equals 4, then there are 4 XORs, 4 rotations ($8\mathbf{O}$) and 4 table lookups ($4\mathbf{L}$). For the remaining $8 - 2 = 6$ columns, there are $6 \times 4 = 24$ XORs and 24 rotations ($48\mathbf{O}$). For a block of 8 columns, $9+8+48 = 65\mathbf{O}$ and $4+4 = 8\mathbf{L}$. Note that the total number of columns in the array W is 60. The previous operations are performed 7 times which produce $7 \times 8 = 56$ columns. In total, $7 \times 65 = 455\mathbf{O}$ and $7 \times 8 = 56\mathbf{L}$ are involved in the key schedule of xAES-256.

D. imeAES

The modification done by Choy et al. to the meAES key schedule is minor. Therefore, the efficiency analysis for the improved key schedule is almost similar to the original. Refer to Algorithm 4.
For imeAES-128, 16 byte XORs with a constant are added in the initialization of a and b to produce one round subkey. Therefore the $\mathbf{O}$ operation from meAES-128 are now increased to become $226 + 16 = 242\mathbf{O}$, while the rest remain. Hence, for total of 11 rounds of imeAES-128, we require $242 \times 11 = 2662\mathbf{O}$, $2112\mathbf{M}$ and $704\mathbf{L}$.
For imeAES-192, $16 \times 2 = 32$ table lookups are removed and replaced with a constant to produce one round subkey. Therefore, the number from $\mathbf{L}$ operation in meAES-192 is now being affected. i.e. $112 - 13 = 99\mathbf{L}$ for one round subkey, and the rest same. Therefore, in total of 13 rounds, $3562\mathbf{O}$, $2496\mathbf{M}$ and $13 \times 99 = 1287\mathbf{L}$ are required for imeAES-192.
While for imeAES-256, neither addition nor removal operation being made, only the change of variables were implemented. Thus the efficiency figures remain as the original May et al.'s proposal, i.e. $4110\mathbf{O}$, $2880\mathbf{M}$ and $1680\mathbf{L}$ are required for imeAES-256.

E. ceAES

For ceAES-128, before producing round subkeys[4], $4 \times 16 = 64$ byte XORs are performed. In producing each round subkey, the AES round function excluding the `AddRoundKey` is iterated once, in addition with $2 \times 16 = 32$ XORs. Therefore, for one round subkey, ceAES-128 which include `SubBytes`, `ShiftRows`, `MixColumns` and 2 other X-

[4]Here, Choy et al. encoded constant C_j, master key K and constant $(keylen - 1)$ as 128-bit string. i.e 16-byte string.

Table 2: Comparison of the efficiency of AES key schedule with other existing proposals

	128-bit key size	192-bit key size	256-bit key size
AES	210**O** + 0 **M** + 40**L**	232**O** + 0 **M** + 32**L**	259**O** + 0 **M** + 56**L**
meAES	2486**O** + 2112**M** + 704**L**	3562**O** + 2496**M** + 1456**L**	4110**O** + 2880**M** + 1680**L**
xAES	330**O** + 0 **M** + 40**L**	392**O** + 0 **M** + 64**L**	455**O** + 0 **M** + 56**L**
imeAES	2662**O** + 2112**M** + 704**L**	3562**O** + 2496**M** + 1287**L**	4110**O** + 2880**M** + 1680**L**
ceAES	1096**O** + 768**M** + 192**L**	1456**O** + 1024**M** + 256**L**	1456**O** + 1024**M** + 256**L**

ORs will require $48 + 6 + 32 = 86$**O**, 64**M** and 16**L**. Thus, in total, $64 + 12 \times 86 = 1096$**O**, $12 \times 64 = 768$**M** and $12 \times 16 = 192$**L** are needed for ceAES-128 key schedule.
For both ceAES-192 and ceAES-256, prior to producing the round subkeys, $5 \times 16 = 80$ XORs are performed. While in producing each round subkey, the AES round function excluding the `AddRoundKey` is iterated once, in addition with $2 \times 16 = 32$ XORs. This operation is same as implemented for ceAES-128, thus make the efficiency for each subkey round remain. i.e. 86**O**, 64**M** and 16**L**. In total of 16 rounds, $80 + 16 \times 86 = 1456$**O**, $16 \times 64 = 1024$**M** and $16 \times 16 = 256$**L** are needed.

F. Remarks

A summary of the efficiency figures for the AES and the new proposals are given in Table 2. We roughly assume that an XOR and a rotation operation (denoted by **O**) are faster than a table lookup (denoted by **L**). We further assume that a table lookup is faster than a multiplication operation (denoted by **M**).
As mentioned early in this section, the key schedule of the AES consists of mostly linear transformations. Firstly, the nonlinear transformation is only applied on columns which index is a multiple of N_k. The remaining columns are applied with linear transformations. Other than this, from Table 2, it can be noted that the key schedule for AES-192 has less number of table lookups (i.e. nonlinear transformation) than AES-128. Furthermore, for AES-192 and AES-256, the attacker has more control in determining the difference between master keys.
Among all the proposals, it is obvious that the original AES key schedule is faster than the other four proposals. The second fastest is the xAES key schedule, followed by ceAES, meAES and lastly imeAES. The degration of performance of these new key schedule proposals is due to the existence of additional transformations to the existing AES key schedule. This is a common security and performance trade-off. When a higher level of security is to be achieved, some performance may be sacrificed in the process.
In response to the recent related-key attacks on the AES, it may be logical to replace the original key schedule of the AES with a new one. However, this may impact many security applications that use the AES due to the changes that needed to be performed. Furthermore, due to the theoretical nature of the attacks, the related-key attacks may not have practical implications to the security of the AES. However, further developments in cryptanalysis may result in a near-practical attack on the AES. If this is the case, then the AES key schedule may be replaced, or we may see another AES competition.
In terms of efficiency alone, the xAES key schedule is the best option to replace the original AES key schedule. This is due to the minimal changes performed over the original key schedule. The designer of xAES has shown that the proposed key schedule is resistant against the latest related-key attacks. However, since the key schedule was proposed in 2010, further cryptanalysis by the cryptographic community is required in order to ascertain its security.
Related-key types of attacks are not entirely theoretical in nature. The Wired Equivalent Privacy (WEP), which is a protocol used to protect 802.11 wireless networks, was shown to be broken in practice using this type of attack. In one instance, Fluhrer, Mantin and Shamir managed to show a practical related key attack on the WEP protocol [17]. The attack is due to the way the protocol generates subsequent keys for the RC4 stream cipher. Using Fluhrer, Mantin and Shamir's findings, Stubblefield Ioannidis and Rubin managed to recover the key used in a WEP-protected network using off-the-shelf hardware and software[30].

VI. Conclusion

In this paper, we have given an overview of the state of security of the AES to date. In particular, we have reviewed existing attacks on the AES and existing block ciphers proposed in the literature to counter the relatey-key type of attacks. Therefore, all the proposals only tweaked the key scheduling algorithm of the AES. We then analyzed the efficiency of each of these block cipher proposals and ranked the ciphers in terms of efficiency.
To the best of our knowledge, the recent single-key and related-key type of attacks against the full AES are theoretical in nature. This is mainly due to the high complexities of the attacks. Furthermore, it is an open problem to simulate a related-key attack model in the real world. Therefore, at this moment, we believe that the AES still remains practically secure.

Acknowledgments

This research is partially supported by Human Capital Development (eHCD) under Ministry of Science, Technology and Innovation (MOSTI), Malaysia and as well as our organization, MIMOS Berhad.

References

[1] Eli Biham, Orr Dunkelman, and Nathan Keller. Related-Key Boomerang and Rectangle Attacks. In Ronald Cramer, editor, *Advances in Cryptology - EUROCRYPT 2005*, volume 3494 of *Lecture Notes in Computer Science*, pages 507–525. Springer-Verlag, 2005.

[2] Eli Biham, Orr Dunkelman, and Nathan Keller. Related-Key Impossible Differential Attacks on 8-Round AES-192. In David Pointcheval, editor, *Topics in Cryptology – CT-RSA 2006, The Cryptographer-*

s' Track at the RSA Conference 2006, volume 3860 of *Lecture Notes in Computer Science*, pages 21–33. Springer-Verlag, 2006.

[3] Alex Biryukov. The Boomerang Attack on 5 and 6-Round Reduced AES. In Hans Dobbertin, Vincent Rijmen, and Aleksandra Sowa, editors, *Advanced Encryption Standard - AES, 4th International Conference, AES 2004*, volume 3373 of *Lecture Notes in Computer Science*, pages 11–15. Springer-Verlag, 2004.

[4] Alex Biryukov, Orr Dunkelman, Nathan Keller, Dmitry Khovratovich, and Adi Shamir. Key Recovery Attacks of Practical Complexity on AES Variants With Up To 10 Rounds. Cryptology ePrint Archive, Report 2009/374, August 2009.

[5] Alex Biryukov and Dmitry Khovratovich. Related-key Cryptanalysis of the Full AES-192 and AES-256. Cryptology ePrint Archive, Report 2009/317, June 2009.

[6] Alex Biryukov and Dmitry Khovratovich. Related-key Cryptanalysis of the Full AES-192 and AES-256. In Mitsuru Matsui, editor, *Advances in Cryptology – ASIACRYPT 2009, 15th International Conference on the Theory and Application of Cryptology and Information Security*, volume 5912 of *Lecture Notes in Computer Science*, pages 1–18. Springer-Verlag, 2009.

[7] Alex Biryukov, Dmitry Khovratovich, and Ivica Nikolić. Distinguisher and Related-Key Attack on the Full AES-256. In Shai Halevi, editor, *Advances in Cryptology – CRYPTO 2009*, volume 5677 of *Lecture Notes in Computer Science*, pages 231–249. Springer-Verlag, 2009.

[8] Andrey Bogdanov, Dmitry Khovratovich, and Christian Rechberger. Biclique Cryptanalysis of the Full AES. In Dong Hoon Lee and Xiaoyun Wang, editors, *Advances in Cryptology - ASIACRYPT 2011 - 17th International Conference on the Theory and Application of Cryptology and Information Security*, volume 7073 of *Lecture Notes in Computer Science*, pages 344–371. Springer-Verlag, 2011.

[9] Jung Hee Cheon, MunJu Kim, Kwangjo Kim, Jung-Yeun Lee, and SungWoo Kang. Improved Impossible Differential Cryptanalysis of Rijndael and Crypton. In Kwangjo Kim, editor, *Information Security and Cryptology – ICISC 2001, 4th International Conference*, volume 2288 of *Lecture Notes in Computer Science*, pages 39–49. Springer-Verlag, 2002.

[10] Jiali Choy, Aileen Zhang, Khoongming Khoo, Matt Henricksen, and Axel Poschmann. AES Variants Secure against Related-Key Differential and Boomerang Attacks. In Claudio Agostino Ardagna and Jianying Zhou, editors, *Information Security Theory and Practice: Security and Privacy of Mobile Devices in Wireless Communication, WISTP 2011*, volume 6633 of *Lecture Notes in Computer Science*, pages 191–207. Springer-Verlag, 2011.

[11] Committee on National Security Systems. CNSS Policy No. 15, Fact Sheet No. 1 National Policy on the use of the Advanced Encryption Standard (AES) to Protect National Security Systems and National Security Information. Committee on National Security Systems, June 2003.

[12] Joan Daemen and Vincent Rijmen. AES proposal: Rijndael. NIST AES Proposal, 1998.

[13] Joan Daemen and Vincent Rijmen. *The Design of Rijndael, AES – The Advanced Encryption Standard*. Springer-Verlag, 2002.

[14] Hüseyin Demirci and Ali Aydın Selçuk. A Meet-in-the-Middle Attack on 8-Round AES. In Kaisa Nyberg, editor, *Fast Software Encryption: 15th International Workshop, FSE 2008*, volume 5086 of *Lecture Notes in Computer Science*, pages 116–126. Springer-Verlag, 2008.

[15] Niels Ferguson, John Kelsey, Stefan Lucks, Bruce Schneier, Mike Stay, David Wagner, and Doug Whiting. Improved Cryptanalysis of Rijndael. In Bruce Schneier, editor, *Fast Software Encryption: 7th International Workshop, FSE 2000*, volume 1978 of *Lecture Notes in Computer Science*, pages 213–230. Springer-Verlag, 2001.

[16] Ewan Fleischmann, Michael Gorski, and Stefan Lucks. Attacking 9 and 10 Rounds of AES-256. In Colin Boyd and Juan González, editors, *Information Security and Privacy, 14th Australasian Conference, ACISP 2009*, volume 5594 of *Lecture Notes in Computer Science*, pages 60–72. Springer-Verlag, 2009.

[17] Scott Fluhrer, Itsik Mantin, and Adi Shamir. Weaknesses in the Key Scheduling Algorithm of RC4. In Serge Vaudenay and Amr M. Youssef, editors, *Selected Areas in Cryptography: 8th Annual International Workshop, SAC 2001*, volume 2259 of *Lecture Notes in Computer Science*, pages 1–24. Springer-Verlag, 2001.

[18] Henri Gilbert and Marine Minier. A Collision Attack on 7 Rounds of Rijndael. In *The Third Advanced Encryption Standard Candidate Conference*, pages 230–241. NIST, 2000.

[19] Herman Isa, Iskandar Bahari, Hasibah Sufian, and Muhammad Reza Z'aba. AES: Current Security and Efficiency Analysis of its Alternatives. In *7th International Conference on Information Assurance and Security, IAS 2011*, pages 267–274. IEEE, 2011.

[20] ISO/IEC. Information technology - Security techniques - Encryption algorithms - Part 3: Block ciphers. ISO/IEC 18033-3, July 2005.

[21] Goce Jakimoski and Yvo Desmedt. Related-Key Differential Cryptanalysis of 192-bit Key AES Variants. In Mitsuru Matsui and Robert J. Zuccherato, editors, *Selected Areas in Cryptography, 10th Annual International Workshop, SAC 2003*, volume 3006 of *Lecture Notes in Computer Science*, pages 208–221. Springer-Verlag, 2004.

[22] Jongsung Kim, Seokhie Hong, and Bart Preneel. Related-Key Rectangle Attacks on Reduced AES-192 and AES-256. In Alex Biryukov, editor, *Fast Software Encryption: 14th International Workshop, FSE 2007*, volume 4593 of *Lecture Notes in Computer Science*, pages 225–241. Springer-Verlag, 2007.

[23] Jiqiang Lu, Orr Dunkelman, Nathan Keller, and Jongsung Kim. New Impossible Differential Attacks on AES. In Dipanwita R. Chowdhury, Vincent Rijmen, and Abhijit Das, editors, *Progress in Cryptology – INDOCRYPT 2008*, volume 5365 of *Lecture Notes in Computer Science*, pages 279–293. Springer-Verlag, 2008.

[24] Stefan Lucks. Attacking Seven Rounds of Rijndael under 192-bit and 256-bit Keys. In *The Third Advanced Encryption Standard Candidate Conference*, pages 215–229. National Institute of Standards and Technology, 2000.

[25] Lauren May, Matt Henricksen, William Millan, Gary Carter, and Ed Dawson. Strengthening the Key Schedule of the AES. In Lynn Margaret Batten and Jennifer Seberry, editors, *Information Security and Privacy: 7th Australasian Conference, ACISP 2002*, volume 2384 of *Lecture Notes in Computer Science*, pages 226–240. Springer-Verlag, 2002.

[26] National Bureau of Standards. Data Encryption Standard. Federal Information Processing Standards (FIPS) 46, 1977.

[27] National Institute of Standards and Technology. Advanced Encryption Standard. Federal Information Processing Standard (FIPS) 197, November 2001.

[28] Ivica Nikolić. Tweaking AES. In Alex Biryukov, Guang Gong, and Douglas R. Stinson, editors, *Selected Areas in Cryptography, 17th International Workshop, SAC 2010*, volume 6544 of *Lecture Notes in Computer Science*, pages 198–210. Springer-Verlag, 2011.

[29] Raphael C.-W. Phan. Impossible Differential Cryptanalysis of 7-round Advanced Encryption Standard (AES). *Information Processing Letters*, 91(1):33–38, July 2004.

[30] Adam Stubblefield, John Ioannidis, and Aviel D. Rubin. Using the Fluhrer, Mantin, and Shamir Attack to Break WEP. In *Proceedings of the Network and Distributed System Security Symposium, NDSS 2002*. The Internet Society, 2002.

[31] Hongjun Wu. Related-Cipher Attacks. In Robert H. Deng, Sihan Qing, Feng Bao, and Jianying Zhou, editors, *Information and Communications Security, 4th International Conference, ICICS 2002*, volume 2513 of *Lecture Notes in Computer Science*, pages 447–455. Springer-Verlag, 2002.

[32] Wentao Zhang, Wenling Wu, and Dengguo Feng. New Results on Impossible Differential Cryptanalysis of Reduced AES. In Kil-Hyun Nam and Gwangsoo Rhee, editors, *Information Security and Cryptology – ICISC 2007, 10th International Conference*, volume 4817 of *Lecture Notes in Computer Science*, pages 239–250. Springer-Verlag, 2007.

[33] Wentao Zhang, Wenling Wu, Lei Zhang, and Dengguo Feng. Improved Related-Key Impossible Differential Attacks on Reduced-Round AES-192. In Eli Biham and Amr M. Youssef, editors, *Selected Areas in Cryptography, 13th International Workshop, SAC 2006*, volume 4356 of *Lecture Notes in Computer Science*, pages 15–27. Springer-Verlag, 2007.

[34] Wentao Zhang, Lei Zhang, Wenling Wu, and Dengguo Feng. Related-Key Differential-Linear Attacks on Reduced AES-192. In K. Srinathan, C. Pandu Rangan, and Moti Yung, editors, *Progress in Cryptology – INDOCRYPT 2007, 8th International Conference on Cryptology in India*, volume 4859 of *Lecture Notes in Computer Science*, pages 73–85. Springer-Verlag, 2007.

Author Biographies

Herman Isa received his B.S. degree of science in industrial mathematics from University of Technology Malaysia (UTM), Malaysia in 2007. Currently he is a Researcher at MIMOS Berhad. His research interest includes data hiding, lightweight cryptography, block cipher and secure embedded system.

Iskandar Bahari received his B.S degree of mathematical science in 2006 and M.S. degree of computational and theoretical science in 2012, both from International Islamic University (IIUM), Malaysia. Currently, he is a Researcher at MIMOS Berhad. His research interest includes the analysis of block cipher, quantum cryptography and lattice-based cryptography.

Muhammad Reza Z'aba received the PhD in Information Technology (Information Security) at Queensland University of Technology, Australia in 2010. He is currently a Staff Researcher at MIMOS Berhad. His PhD thesis is on the linear relationship in block ciphers. His research interest includes cryptography particularly in the design and analysis of block ciphers.