AES Variants Secure against Related-Key Differential and Boomerang Attacks

Jiali Choy, Aileen Zhang, Khoongming Khoo, Matt Henricksen, Axel Poschmann

AES Variants Secure Against Related-Key Differential and Boomerang Attacks*

Jiali Choy[1], Aileen Zhang[1], Khoongming Khoo[1], Matt Henricksen[2] and Axel Poschmann[3]

[1] DSO National Laboratories
20 Science Park Drive, Singapore 118230
{cjiali, zyinghui, kkhoongm}@dso.org.sg
[2] Institute for Infocomm Research,
A*STAR, Singapore
mhenricksen@i2r.a-star.edu.sg
[3] Division of Mathematical Sciences, School of Physical and Mathematical Sciences
Nanyang Technological University, Singapore
aposchmann@ntu.edu.sg

Abstract. In this paper, we present a framework for protection against the recent related-key differential and boomerang attacks on AES by Biryukov et al. Then we study an alternative AES key schedule proposed by May et al. at ACISP 2002 as a possible candidate to protect against these related key attacks. We find that there exist equivalent keys for this key schedule and in response, we propose an improvement to overcome this weakness. We proceed to prove, using our framework, that our improved May et al.'s key schedule is secure against related-key differential and boomerang attacks. Since May et al.'s key schedule is not on-the-fly (which is a requirement for some hardware implementations), we propose an on-the-fly AES key schedule that is resistant against related-key differential and boomerang attacks.

Keywords: Related-key attacks, differential cryptanalysis, boomerang attacks, AES key schedule.

1 Introduction

In [4], Biryukov et al. launched the first known key-recovery attack on AES-256. It is a related-key differential attack that exploits a differential characteristic path of high probability, where we allow both the plaintext and key to have non-zero differentials. The attack has a time/data complexity of 2^{131} and requires 2^{65} memory in addition to 2^{35} related key pairs. Later in [3], Biryukov and Khovratovich used a shortened version of the related-key differential characteristic of [4] to construct a distinguisher for the related-key boomerang attack on AES-256.

* The extended version of this paper shall appear on the eprint archive.

This allowed the authors to avoid the majority of the active S-boxes in the differential characteristic of [4], which resulted in a much improved attack with time and data complexity of $2^{99.5}$, requiring 2^{77} memory and just 4 related keys. A similar approach was used to derive a related-key boomerang attack on AES-192 with data complexity 2^{123}, time complexity 2^{176}, and memory complexity 2^{152} in addition to 4 related keys. This is also the first known key-recovery attack on full AES-192. However, we need adaptive ciphertext decryption for the attack of [3] whereas only chosen plaintext encryption is needed for the attack of [4]. In our paper, we present a framework for practical resistance against Biryukov et al.'s related-key differential attacks from [3, 4].

The structure of the AES-256 cipher is still very secure as the best non related-key attacks can work up to at most 8 of the 14 rounds [9]. The more devastating related key attacks [3, 4] exploits the high linearity in the AES-256 key schedule. If we look only at a key schedule differential characteristic path, it is possible to find paths which involve only one active S-box. Thus a key point in securing AES against the latest related-key differential/boomerang attacks is to make the key schedule more nonlinear, so that any related-key differential path would involve more active S-boxes in the subkey differences[4].

1.1 Our Contribution

We design two new AES variants to protect against the related-key attacks of [3, 4] by making the AES key schedule more nonlinear, while keeping the main AES cipher the same. Thus we retain the strong security of AES against non related-key attacks.

Construction 1: In Section 3, we consider the possibility of using an alternative AES key schedule by May et al. [11]. This key schedule was shown to have good statistical properties while achieving the strong property of round key irreversibility and resistance against previously known related-key attacks [1]. However, we show in Section 3.1 that there are pairs of equivalent keys that produce the same encryption functions. We propose an improvement of their key schedule in Section 3.2 that avoids this weakness. Based on our framework, we prove in Sections 3.3 and 3.4 that the improved May et al.'s key schedules for AES have practical resistance against related-key differential and boomerang attacks. This key schedule is also secure against related-cipher attacks and slide attacks, and has round key irreversibility.

Construction 2: However, our improved May et al.'s design uses three AES rounds to derive a subkey. This is too expensive for hardware implementation, which requires on-the-fly key schedule. In Section 4, we propose an on-the-fly key schedule design for AES-128, AES-192, and AES-256, where the time needed to derive each round key is no more than the computation of 1.25 (amortized) AES

[4] Our observations also correspond with those made by Kim et al. in Section 2.6 of [10] where it is mentioned that if the key schedule of the cipher is complex enough and does not have "good" differential properties, then the number of keys required for the attack becomes infeasibly large.

rounds. Furthermore, we prove that this new key schedule has practical resistance against related-key differential and boomerang attacks. This key schedule is also secure against related-cipher attacks and slide attacks, and has partial round key irreversibility.

2 Framework for Protection Against Related Key Differential and Boomerang Attacks

2.1 Some Definitions and Notation

We first define some notation and concepts which form the basis of differential attacks.

Given a block cipher, the plaintext, secret key and ciphertext are denoted by P, K and C respectively. The encryption and decryption processes are denoted by $C = E_K(P)$ and $P = E_K^{-1}(C)$ respectively. We denote the input of the first round by P_0, and the output of the i^{th} round by P_i, $i = 1, \ldots, NR$, where NR is the number of rounds. Similarly, we write K_i, $i = 0, \ldots, m$, for the $m + 1$ subkeys generated by the key schedule.

To launch a differential attack, one attempts to find a pair of differences in the plaintext and ciphertext that occur with high probability. This usually involves finding a sequence of round inputs and outputs that occur with high probability. We write ΔP and ΔC to denote a plaintext and ciphertext difference respectively, and ΔP_i to denote the difference in the round output of round i. A *differential characteristic* refers to a sequence of input differences to the rounds

$$(\Delta P_0 \longrightarrow \Delta P_1 \longrightarrow \cdots \longrightarrow \Delta P_{NR})$$

We abbreviate the above expression to $(\Delta P \xrightarrow{dc} \Delta C)$.

Similarly, to launch a related-key differential attack, one attempts to find a set of differences for (P, K, C) that hold with high probability. We shall see that this can be done by finding a sequence of differences in the key and subkeys generated by the key schedule, and the plaintext and round outputs generated by the main cipher, such that these differences occur with high probability.

We first consider a differential characteristic in the key schedule alone. We denote a difference in the key by ΔK, and differences in the subkeys ΔK_i, $i = 0, \ldots, m$. We note that the subkeys are not necessarily derived sequentially from each other, so the concept of a differential characteristic 'path' may not exist in this sense. We therefore write $(\Delta K \xrightarrow{dc} \Delta K_0, \ldots, \Delta K_m)$ for a differential characteristic in the key schedule.

Now we consider a differential characteristic in the key schedule and the main cipher. We write this as

$$(\Delta K \xrightarrow{dc} \Delta K_0, \ldots, \Delta K_m,\ \Delta P_0 \longrightarrow \cdots \longrightarrow \Delta P_{NR})$$

which we abbreviate to $(\Delta P, \Delta K \xrightarrow{dc} \Delta C)$. We also define:

$$p_k = Prob(\Delta K \xrightarrow{dc} \Delta K_0, \ldots, \Delta K_m),$$
$$p_{c|k} = Prob(\Delta P \xrightarrow{dc} \Delta C | \Delta K \xrightarrow{dc} \Delta K_0, \ldots, \Delta K_m).$$

It is easy to see that $Prob(\Delta P, \Delta K \xrightarrow{dc} \Delta C) = p_k \times p_{c|k}$ by Bayes' Theorem.

2.2 Protection Against Related-Key Differential Attack of [4]

The attacker must run through p_k^{-1} key pairs on average in order to find one that satisfies the specified differential characteristic in the key schedule. For each key pair, the differential attack has complexity $O(p_{c|k}^{-1})$ and needs the same number of chosen plaintexts, so in total the complexity is $O((p_{c|k}p_k)^{-1})$. In the attack of [4], we have $p_k = 2^{-35}$, $p_{c|k} = 2^{-93}$, and with some computational overheads get an attack complexity of 2^{131}. The interested reader should refer to [4] for the details.

We can defend against this attack by having $p_k \times p_{c|k} < 2^{-NK}$ where NK is the key size of the cipher, for any related-key differential characteristic, i.e. no good distinguisher can be found that can be exploited in a related-key differential attack. The attack also cannot be applied if $p_{c|k} < 2^{-NB}$, where NB is the block size of the cipher, as there would be insufficient plaintexts to launch the attack.

2.3 Protection Against Related-Key Boomerang Attack of [3]

The main idea behind the boomerang attack [2, 15] is to use two short differential characteristics of high probabilities instead of one long differential characteristic of low probability. We assume that a block cipher $E : \{0,1\}^{NB} \times \{0,1\}^{NK} \rightarrow \{0,1\}^{NB}$ can be described as a composition of two sub-ciphers, i.e. $E = E_1 \circ E_0$. Here, NB and NK denote the block size and key size of the cipher respectively. Suppose we have a related-key differential characteristic $\alpha \rightarrow \beta$ of E_0 (excluding a couple of rounds at the beginning of the cipher) under a key difference ΔK_0 with probability p and another related-key differential characteristic $\gamma \rightarrow \delta$ for E_1 under key difference ΔK_1 with probability q. Here, $p = p_k \times p_{c|k}$ where p_k is the probability that the differential characteristic path in the key schedule corresponding to E_0 will be satisfied while $p_{c|k}$ is the probability that the differential characteristic path in the main cipher, $\alpha \rightarrow \beta$ in E_0, will be satisfied given that the key differential characteristic is satisfied. Likewise, $q = q_k \times q_{c|k}$ with similar definitions pertaining to E_1. The differential characteristic trails of E_0 and E_1 are called *upper* and *lower* trails respectively.

The related-key boomerang process involves four different unknown but related keys. The relation between the keys can be an arbitrary bijective function R chosen in advance by the attacker. A plaintext pair results in a *quartet* with probability p^2q^2 whereas for a random permutation, the probability of obtaining a good quartet is 2^{-NB}.

The attacker must run through $(p_k q_k)^{-2}$ quartets of related keys on average in order to find one that satisfies the specified differential characteristic in the key schedule. For each quartet, the attack has complexity $O((p_{c|k} q_{c|k})^{-2})$, so in total the complexity is $O(1/(p_{c|k} p_k))$. In the attack of [3], we have $(p_k q_k)^2 = 1$, $(p_{c|k} q_{c|k})^2 = 2^{-96}$, and with some computational overheads get an attack complexity of $2^{99.5}$. The interested reader should refer to [3] for the details.

We can defend against this attack by having $(p_k q_k)^2 (p_{c|k} q_{c|k})^2 < 2^{-NK}$ where NK is the key size of the cipher, for all decompositions of the cipher into two smaller sub-ciphers and for all differential characteristics for these sub-ciphers. This would mean that there do not exist any boomerang quartets of high probability that can be exploited. The attack also cannot be applied if $(p_{c|k} q_{c|k})^{-2} < 2^{-NB}$, where NB is the block size of the cipher, as there would be insufficient plaintexts to launch the attack.

3 Security of Improved May et al.'s AES Key Schedule Against Related-key Attack

To protect AES against related-key differential and boomerang attacks, one strategy is to use a strengthened key schedule with good differential properties. One possible candidate is an alternative key schedule for AES proposed by May et al. in [11] in 2002. At that time, there was already a 9-round related-key square attack [9] on AES-256 which exploited the slow diffusion of relatively few non-linear elements in the key schedule. May et al. wanted to design an efficient key schedule with more nonlinear components and better diffusion to defend against such attacks.

Their key schedule for AES-256 is shown below. Here, NR=14 is the number of rounds; a, b are 128-bit values derived from the Master Key $MK = MK_0|MK_1|\ldots|MK_{32}$, $a = a_0|a_1|\ldots|a_{15}$ (the MK_i and a_i are 8-bit values, and $|$ represents concatenation), r is the round number and K_r is the 128-bit round subkey for Round q. Each round subkey is the 128-bit output after the execution of three rounds of the cipher algorithm, using the master key (with the addition of different round constants) as both data and key input.

```
for r = 0 to NR
        for j = 0 to 15
                a_j = MK_j ⊕ S[r * 16 + j] ⊕ S[MK_{j+16}]
                b_j = MK_{j+16} ⊕ S[r * 16 + j] ⊕ S[MK_j]
        for i = 0 to 2
                SubBytes(a)
                ShiftRows(a)
                MixColumns(a)
                AddRoundKey(a, b)
        K_r = a
```

May et al.'s Key Schedule for AES-256

In [11], the authors conducted statistical tests to show that for their proposal, there is no bit leakage between round subkeys. Furthermore, each round key satisfies both the frequency and Strict Avalanche Criterion (SAC) tests, indicating

good pseudorandomness properties such as bit confusion and diffusion. The authors concluded that previously published attacks that exploit the key schedule such as the standard related-key attacks [1] will not work on their proposed key schedule.

Moreover, the key schedule achieves the property of round key irreversibility, by which we mean that given any subset of the round keys, it is hard to derive the remaining round keys. This forces an adversary to attack all the round keys. This is in contrast to the AES key schedule, which is reversible - given any two round keys, one can derive all the other round keys. The obvious countermeasures for preventing related-key attacks on AES, such as increasing the number of rounds and hashing the key before expanding it, also produce key schedules which are also reversible.

3.1 Equivalent Keys in May et al.'s Key Schedule

Despite the good cryptographic properties of May et al.'s key schedule as mentioned in the previous section, we shall show that their key schedule has equivalent keys as shown in the following proposition.

Proposition 1 *In May et al.'s key schedule for AES-256, there are* 2^{271} *equivalent key pairs* $\{(MK, MK') : MK \neq MK'\}$ *such that* $AES_{MK}(\cdot) = AES_{MK'}(\cdot)$*, i.e. they produce the same encryption output.*

Proof. Consider the 4-byte tuple $(MK_i, MK'_i, MK_{i+16}, MK'_{i+16})$ for each index i. We look for those that satisfy the equations:

$$MK_i \oplus MK'_i = S[MK_{i+16}] \oplus S[MK'_{i+16}],\ S[MK_i] \oplus S[MK'_i] = MK_{i+16} \oplus MK'_{i+16}. \tag{1}$$

By a computer simulation, there are 65644 tuples $(MK_i, MK'_i, MK_{i+16}, MK'_{i+16})$ that satisfy equation (1). In that case,

$$\Delta a_i = \Delta MK_i \oplus \Delta S[MK_{i+16}] = 0 \text{ and } \Delta b_i = \Delta MK_{i+16} \oplus \Delta S[MK_i] = 0$$

Thus for $i = 0, 1, 2, \ldots, 15$, if we let $(MK_i, MK'_i, MK_{i+16}, MK'_{i+16})$ satisfy equation (1) for $s \geq 1$ of the indices i and let $(MK_i, MK_{i+16}) = (MK'_i, MK'_{i+16})$ for the rest of the $16 - s$ indices, we will have $\Delta a = 0 = \Delta b$. From the definition of May et al.'s key schedule, this implies the subkeys derived from MK and MK' are the same and they will produce the same encryption output. The number of such equivalent key pairs are given by:

$$\sum_{s=1}^{16} \binom{16}{s} \times 65644^s \times (256^2)^{16-s} \approx 2^{272}.$$

When (MK, MK') is an equivalent key pair, (MK', MK) is also an equivalent key pair. Thus we divide the total number of equivalent key pairs by 2 to get 2^{271}.

□

3.2 An Improved May et al.'s Key Schedule

From Section 3.1, we have seen that the three AES rounds used to generate each round key in May et al.'s key schedule help to ensure that the round keys have good statistical properties and attain round key irreversibility. The problem is the initialization of a and b which allows an adversary to force a, b to have zero differential by choosing an appropriate pair of related secret keys.

Below, we propose an improved version of the May et al.'s key schedule. Basically, we simplify the initialization of a, b so that each byte of a and b only depends on one instead of two bytes of the secret key. This prevents an adversary from using the technique in Proposition 1 to force Δa and Δb to be zero. We also make use of key-length-dependent counters $keylen$ to defend against the related-cipher attack [16], which was first applied to the alternative AES key schedule proposed by May et al. In the algorithm shown, $keylen - 1$ refers to the key length of the cipher (minus 1) encoded as a byte. A more detailed explanation of the related-cipher attack can be found in [16].

This key schedule, as with the original key schedule by May et al., has the property of round key irreversibility.

Next, we shall show in the following section that the improved May et al.'s key schedule can protect AES against related-key differential and boomerang attacks.

```
for r = 0 to NR
    for j = 0 to 15
        a_j = S[MK_j] ⊕ S[r * 16 + j] ⊕ (keylen − 1)
        b_j = S[MK_{j+16}] ⊕ S[r * 16 + j] ⊕ (keylen − 1)
    for i = 0 to 2
        SubBytes(a)
        ShiftRows(a)
        MixColumns(a)
        AddRoundKey(a, b)
    K_r = a
```

Improved May et al.'s Key Schedule AES-256

3.3 Improved May et al.'s Key Schedule is Secure Against Related-Key Differential Attack

Our aim in this section is to study the security of our improved May et al.'s key schedule against the related-key differential attack which was recently used by Biryukov et al. [4, 3, 5] to attack full-round AES-256.

We have the following technical lemma which will be used to prove the main results of this section later on.

Lemma 1. *For any round subkey generation using the key schedule proposal described above, if we have a pair of master keys with nonzero difference, then the differential characteristic path either has at least four active S-boxes, or it has at least three active S-boxes and an additional four active S-boxes resulting from the generation of a and b.*

The proof of this lemma can be found in Appendix A of this paper. Based on the above result, we may deduce the following corollary.

Corollary 1. *If we have a pair of master keys with nonzero difference, then our improved May et al.'s key schedule for AES-256 has at least* 43 *active S-boxes involved in the generation of* 13 *subkeys.*

Proof. By Lemma 1, the differential characteristic path for each subkey generation has at least three active S-boxes, and if there exists a subkey whose differential characteristic path produces only three active S-boxes occurs, then there will be four additional S-boxes involved in generating a and b. This would give at least $13 \times 3 + 4 = 43$ S-boxes in total. On the other hand, if the differential characteristic path for each subkeys produces at least four active S-boxes, there are at least $13 \times 4 > 43$ S-boxes in total. □

In the attacks on AES, we always consider an $(NR-2)$-round attack involving $NR-1$ subkeys, in keeping with [4] where the attack is based on an $(NR-2)$-round related-key differential characteristic.

Theorem 1. *AES-256 using our improved May et al.'s key schedule is resistant to related-key differential attack.*

Proof. We apply Corollary 1 for AES-256 assuming an $NR-2$ round attack, i.e. a 12-round attack which involves 13 subkeys. Since each active S-box has probability at most 2^{-6}, this gives a probability of at most $p_k \times p_{c|k} = (2^{-6})^{43} \times p_{c|k} = 2^{-258} \times p_{c|k} < 2^{-256}$. Therefore, we may conclude that AES-256 with the strengthened key schedule is indeed resistant to related-key differential attacks. □

May et al. also proposed alternative key schedules for AES-128 and AES-192. The key schedules proposed by May et al. in [11] for AES-128 and AES-192 are largely the same as that for AES-256, except that a and b are generated in slightly different ways: for $r = 0$ to NR, $j = 0$ to 15.

(1) For AES-128: $a_j = b_j = MK_j \oplus S[r * 16 + j]$.
(2) For AES-192: $a_j = MK_j \oplus S[r * 16 + j] \oplus S[MK_{j+8}]; b_j = MK_{j+8} \oplus S[r * 16 + j] \oplus S[MK_j]$.

It is easy to see that equivalent keys similar to those in Proposition 1 exist for May et al.'s key schedule for AES-192. Thus we propose a similar improvement to May et al.'s key schedule for AES-192 below. As before, we also tweaked the key schedules a bit by introducing key-length-dependent counters *keylen* for protection against the related-cipher attack [16].

(1) Improvement for AES-128: $a_j, b_j = MK_j \oplus S[r * 16 + j] \oplus (keylen - 1)$.
(2) Improvement for AES-192: $a_j = MK_j \oplus S[r * 16 + j] \oplus (keylen - 1); b_j = MK_{j+8} \oplus S[r * 16 + j] \oplus (keylen - 1)$.

Based on the above description of the schedules for AES-128 and AES-192, it is easy to deduce the following corollary from the proof of Lemma 1.

Corollary 2. *If we have a pair of master keys with nonzero difference in our proposed improvement of the key schedule of [11] for AES-128 and AES-192, then there are at least* 3 *active S-boxes involved in the generation of each subkey.*

Corollary 2 allows us to prove Theorem 2.

Theorem 2. *AES-128 and AES-192 using the key schedule of [11] for AES-128 and our improvement for AES-192 are also resistant to related-key differential attack.*

Proof. In our improved key schedule for AES-128 and AES-192, we see that if a pair of master keys has non-zero difference, then one of Δa or Δb is non-zero. Thus we can use the fact that every differential characteristic path of a round subkey generation has at least 3 active S-boxes (excluding the generation of a and b) from the proof of Lemma 1.

For AES-128, an 8-round attack involves 9 subkeys. Assuming that each active S-box has probability 2^{-6}, this gives a probability of at least $p_k \times p_{c|k} = (2^{-6})^{(9\times 3)} \times p_{c|k} = 2^{-162} \times p_{c|k} < 2^{-128}$.

Similarly for AES-192, a 10-round attack involves 11 subkeys. The differential characteristic probability is at least $p_k \times p_{c|k} = (2^{-6})^{(11\times 3)} \times p_{c|k} = 2^{-198} \times p_{c|k} < 2^{-192}$.

Therefore, AES-128 and AES-192 with the strengthened key schedule are resistant to differential related-key attacks. □

Remark 1. Our proofs show that the original May et al.'s key schedule is also resistant against related-key differential attack. This is because if a pair of keys is an equivalent weak key pair, then they produce the same roundkeys and we have normal differential attack instead of related-key differential attack. If they are not an equivalent key pair, then Δa or Δb is non-zero and we can apply[5] Lemma 1 to prove Theorem 1 for May et al.'s key schedule.

3.4 Improved May et al.'s Key Schedule is Secure Against Related-Key Boomerang Attack

We consider an arbitrary decomposition of AES, with our improved key schedule, into two smaller sub-ciphers. The generation of the subkeys by the key schedule will be split between the two sub-ciphers. Since the subkeys are independently generated, $p_k q_k$ is simply the product of the probabilities that the differential characteristics hold for the generation of each subkey.

We note that it may be possible to bypass one subkey for the round at which the cipher is split into two using a boomerang switch. Furthermore, we assume that two rounds at the start can be ignored by not specifying the differences in the differential trail (as in [3], where one round at the start is ignored). Hence, for AES-128, we consider the generation of 7 subkeys; for AES-192, 9; and for AES-256, 11.

[5] We can apply Lemma 1 because both the improved and original May et al.'s key schedule uses the same 3-round AES structure to generate each roundkey.

By Lemma 1 and Corollary 2, we see that there are at least 3 active S-boxes involved in the generation of each subkey for all three versions of AES. For t subkeys, the product of the probabilities that the differential characteristics hold for each subkey is $(2^{-6})^{3t}$. Since $p_k q_k \leq 2^{-18t}$, $(p_k q_k)^2 < 2^{-NK}$ holds if $(2^{-18t})^2 < 2^{-NK}$.

If $t \geq 4$, we have $(p_k q_k)^2 < 2^{-128}$; if $t \geq 6$, we have $(p_k q_k)^2 < 2^{-192}$; and if $t \geq 8$, we have $(p_k q_k)^2 < 2^{-256}$. For AES-128, $t = 7$; for AES-192, $t = 9$; and for AES-256, $t = 11$. Hence, for AES with the strengthened key schedule, for any decomposition into two sub-ciphers, there does not exist a boomerang quartet of high probability which can be exploited. Therefore, we have proved that AES-128, AES-192 and AES-256 using the strengthened key schedule of [11] are resistant to related-key boomerang attack.

By a reasoning similar to Remark 1, the original May et al.'s key schedule is also resistant against related-key boomerang attack.

4 A New On-the-fly Key Schedule for AES Secure Against Related-Key Differential and Boomerang Attacks

We present here a new key schedule for AES that offers several advantages over both the original key schedule (security against related-key differential and boomerang attacks) and that proposed by May et al. [11] (better efficiency).

The key schedule shown below generates fifteen 128-bit round keys $K_i, 0 \leq i \leq NR = 14$ from a 256-bit master key or thirteen 128-bit round keys $K_i, 0 \leq i \leq NR = 12$ from a 192-bit master key. Round key K_i is used in the i^{th} round of encryption. For a 256-bit master key, one subkey SK_0 is not converted into a usable round key, and for a 192-bit master key, three subkeys $SK_{0..2}$ are not converted into usable round keys.

Here, C_j denote 128-bit strings which are initialized by equating them to integers j encoded as 128-bit strings. $keylen - 1$ refers to the key length of the cipher (minus 1) also encoded as a 128-bit string. $1R\_AES(x)$ refers to one round of unkeyed AES with the plaintext x. The $AddRoundKey$ operation in the AES round can be omitted or provided with a null key.

These proposed key schedules for AES-192 and AES-256 are partially irreversible, by which we mean that, given two round keys, it is hard to derive the rest of the round keys. However, given certain combinations of three or more round keys, it may be possible to derive the rest of the round keys. For example, if we have SK_i, SK_{i+1} for $i = 1$ or 2, as well as SK_4, then we can obtain $K1$ from SK_i, SK_{i+1}, and then we either have, or can compute SK_3, and then use SK_4 to get $K2$. In this sense, this proposed key schedule is weaker than the original and improved key schedules by May et al. Nonetheless, partial irreversibility is a desirable property which is lacking in the original AES-192 and AES-256 key schedules.

```
if AES-192
        f = 1
if AES-256
        f = 2

for j = 0 to 15
        K1_j = MK_j
        K2_j = MK_{j+(8*f)}

for j = 0 to 15
        C_j = j
C_0 = C_0 ⊕ K1 ⊕ (keylen − 1)
C_4 = C_4 ⊕ K2
C_8 = C_8 ⊕ K1
C_12 = C_12 ⊕ K2

SK_{-1} = K1, I_{-1} = 0
for i = 0 to 15
        I_i = 1R_AES(I_{i-1} ⊕ C_i)
        SK_i = I_i ⊕ SK_{i-1}
         if AES-192
                K_{i-3} = SK_i
         if AES-256
                K_{i-1} = SK_i
```

New key schedule proposal for 192-bit and 256-bit keys

The following key schedule shown generates eleven 128-bit round keys $K_i, 0 \leq i \leq NR = 10$ from a 128-bit master key. One subkey SK_0 is not converted into a usable round key.

```
for j = 0 to 11
        C_j = j
C_0 = C_0 ⊕ MK ⊕ (keylen − 1)
C_4 = C_4 ⊕ MK
C_8 = C_8 ⊕ MK

SK_{-1} = MK, I_{-1} = 0
for i = 0 to 11
        I_i = 1R_AES(I_{i-1} ⊕ C_i)
        SK_i = I_i ⊕ SK_{i-1}
        K_{i-1} = SK_i
```

New key schedule proposal for 128-bit keys

Our proposed key schedule for AES-128 is also partially irreversible in that at least two round keys are needed to derive the rest of the round keys, and only certain combinations of keys can work. In contrast, the original AES-128 key schedule requires only one round key to derive all the other round keys.

Theorem 3. *AES-128, AES-192 and AES-256 with the key schedules proposed in this section are resistant against related-key differential and boomerang attacks.*

Proof. In this proof, we use the fact that the differential characteristic probability of four consecutive AES rounds is bounded by 2^{-150} [8, page 33]. This result holds only when the input differential is non-zero and encryption is under a fixed key, i.e. the subkey differentials are zero.

For an attacker to control the round key differences ΔK_i to launch a related-key attack, he would need to control the output differential ΔI_i of the key

schedule internal state. Thus we need to prove that the differential probability of this internal state is low enough to prevent related-key differential and boomerang attacks. Since we are considering related key differential attack, we assume $\Delta MK \neq 0$.

Key schedule for AES-128: $\Delta MK \neq 0$ implies the input differential to the first four AES rounds of the internal state is non-zero. Therefore the differential characteristic probability of the key schedule internal state I_i is bounded by 2^{-150}.

Key schedule for AES-192, AES-256: $\Delta MK \neq 0$ implies $\Delta(K1, K2) \neq 0$. Thus, we consider the three cases $\Delta K1 \neq 0$, $\Delta K2 = 0$; $\Delta K1 = 0$, $\Delta K2 \neq 0$ and $\Delta K1 \neq 0$, $\Delta K2 \neq 0$.

When $\Delta K1 \neq 0$, $\Delta K2 = 0$, the first round corresponding to internal state I_0 will have a non-zero differential input $\Delta K1$. Rounds 2 to 8 corresponding to I_1 to I_7 will have zero input key differences. Thus the differential characteristic probability of these eight rounds, and consequently, of the entire key schedule internal state I_i is at most $(2^{-150})^2 = 2^{-300}$.

When $\Delta K1 = 0$, $\Delta K2 \neq 0$, the first four rounds corresponding to I_0 to I_3 will have zero differential characteristic probability since there is a zero input difference and no input key differences for all four rounds. The fifth round corresponding to I_4 will have a non-zero differential input $\Delta K2$. Following this, rounds 6 to 12 corresponding to internal state I_5 to I_{11} will have zero input key differences. Thus the differential characteristic probability of these eight rounds, and consequently, of the entire key schedule internal state I_i is at most $(2^{-150})^2 = 2^{-300}$.

When $\Delta K1 \neq 0$, $\Delta K2 \neq 0$, the first round corresponding to internal state I_0 will have a non-zero differential input $\Delta K1$ while rounds 2 to 4 corresponding to I_1 to I_3 will have zero input key differences. This gives a differential characteristic probability of at most 2^{-150} for the first four rounds. The differential output after these four rounds is ΔI_3. If $\Delta I_3 \oplus \Delta K2 \neq 0$, then we have a non-zero differential input to the next four AES rounds corresponding to I_4 to I_7. Since rounds 6 to 8 corresponding to internal states I_5 to I_8 have zero key differences, this gives a differential characteristic probability of at most 2^{-150} for rounds 5 to 8. If $\Delta I_3 \oplus \Delta K2 = 0$, then there is no differential characteristic probability associated with rounds 5 to 8. But $\Delta K1$ will be a non-zero differential input to the next four AES rounds while rounds 10 to 12 corresponding to internal states I_8 to I_{11} have zero input key differences. This gives a differential characteristic probability of at most 2^{-150} for rounds 9 to 12. In both cases, the differential characteristic probability of the key schedule internal state I_i is at most $(2^{-150})^2 = 2^{-300}$.

For protection against related-key boomerang attack, when we split the cipher into two sub-ciphers E_0, E_1, the corresponding internal state I_i of the key schedule for one of the sub-cipher will contain 4 unkeyed AES rounds with a non-zero input differential. This means one of p_k or q_k is bounded by 2^{-150} and that $(p_k q_k)^2 \leq 2^{-300}$. Thus our cipher is secure against related-key boomerang attack.

□

For protection against other attacks on the key schedule, the use of round counters defeats slide attacks [6, 7]. As in the case for the improved May et al.'s key schedule, the use of key-length-dependant counters *keylen* defeats the related-cipher attack [16].

The key schedule offers better efficiency than the proposal by May et al. which invokes three AES rounds and a few S-box lookups per round key. Our key schedule proposal invokes at most an (amortized) 1.25 AES rounds per round key, making it more suitable for hardware implementation. If two AES round functions are implemented in parallel, it is three times as fast as the May et al. key schedule to encrypt; or if a single AES round function is implemented, it is twice as fast.

4.1 Hardware Implementation

Usually hardware implementations of encryption algorithms are optimized for high throughput, i.e. first for speed and then for area. If we look on these typically round-based architectures, our proposed new AES key schedule introduces only minor timing overheads compared to the original AES key schedule. Then for the encryption of one block with a 128-bit key 11 clock cycles are required (compared to 10 clock cycles for standard AES) and for 192-bit and 256-bit keys we need 15 clock cycles compared to 12 and 14 clock cycles, respectively. Note that for AES-256 -which suffers most from recent related key attacks and needs to be fixed most urgently- this is an overhead of only 7%.

At the same time, the similarity of the key schedule and the data path allows a better time-area trade-off and thus more flexibility for implementation. A designer can choose to implement both data paths (as described above, variant A) or to share resources between them (variant B). The latter variant B allows to save area at the cost of additional clock cycles (21 for AES-128, 25 for AES-192 and 29 for AES-256). The proposal by May *et al.* invokes three AES rounds and a few S-box lookups per round key. Therefore it cannot compute the round keys on-the-fly and will never achieve the same speed as the standard AES or our proposal, regardless of the hardware spent. Using a shared data path (variant B), our proposal is twice as fast as the proposal by May *et al.*, and using two separate data paths (variant A) it is three times faster.

Also the area overhead of our proposal is very moderate as the following estimations, which are based on the 180 nm *UMCL18G212D3* standard-cell library from UMC [14], indicate. In a round-based implementation, we need two 128-bit XOR gates to add MK and SK (600 GE) and a 4-bit XOR gate to add C_i (10 GE). Depending on the key length we need a 7-bit XOR gate (17 GE) or an 8-bit XOR gate (19 GE) and for AES-192 and AES-256 we also need a 128-bit MUX (342 GE). Finally a 128-bit AND gate (170 GE) is required to handle the proper addition of MK and the variables I and SK need to be stored in flip-flops (1536 GE). If the master key is never changed, it can be hardwired and requires no gates. Otherwise we have an additional storage overhead of 768 GE for AES-128, 1152 GE for AES-192, and 1536 GE for AES-256. For variant A an additional complete round of AES is required. Since the gate count for an

AES round depends on a wide variety of design choices, an estimation of the total overhead for variant A is difficult. We therefore concentrate on variant B, which only needs an additional 128-bit AND gate (170 GE). For variant B our proposal introduces an overhead of 2505 GE with a hardwired MK and 3270 GE with a flexible MK for AES-128. For AES-192 it sums up to 2850-4000 GE and for AES-256 to 2850-4385 GE.

To put these overhead figures into perspective, please note that a typical throughput-optimized co-processor implementation of AES-128 requires tens of thousands of GE: Satoh *et al.* report such an implementation on a 0.11 μm technology with 54,000 GE [13], while the implementation of Pramstaller *et al.* on a 0.6 μm technology requires 85,000 GE [12].

Acknowledgements

We would like to thank the anonymous reviewers of our previous paper submission for their valuable comments.

References

1. E. Biham, "New Types of Cryptanalytic Attacks using Related Keys", Advances in Cryptology - EUROCRYPT 1993, LNCS 765, pp. 398-409, Springer-Verlag, 1993.
2. E. Biham, O. Dunkelman, and N. Keller, "Related-Key Boomerang and Rectangle Attacks", Eurocrypt 2005, LNCS 3494, pp. 507-525, Springer, 2005.
3. A. Biryukov and D. Khovratovich, "Related-Key Cryptanalysis of the Full AES-192 and AES-256", Asiacrypt 2009, LNCS 5912, pp. 1-18, Springer-Verlag, 2009.
4. A. Biryukov, D. Khovratovich, and I. Nikolic, "Distinguisher and Related-Key Attack on the Full AES-256", Crypto 2009, LNCS 5677, pp. 231-249, Springer-Verlag, 2009.
5. A. Biryukov, O. Dunkelman, N. Keller, D. Khovratovich, and A. Shamir, "Key Recovery Attacks of Practical Complexity on AES Variant with Up To 10 Rounds", IACR eprint server, 2009/374 July 2009, `http://eprint.iacr.org/2009/374`.
6. A. Biryukov and D. Wagner, "Slide Attacks", FSE 1999, LNCS 1636, pp. 245-259, Springer, 1999.
7. A. Biryukov and D. Wagner, "Advanced Slide Attacks", Eurocrypt 2000, LNCS 1807, pp. 589-606, Springer, 2000.
8. J. Daemen and V. Rijmen, "Rijndael", First Advanced Encryption Standard Conference, August 1998, `http://csrc.nist.gov/encryption/aes/`.
9. N. Ferguson, J. Kelsey, S. Lucks, B. Schneier, M. Stay, D. Wagner, and D. Whiting, "Improved Cryptanalysis of Rijndael", FSE 2000, LNCS 1978, pp. 213-230, Springer-Verlag, 2000.
10. J. Kim, S. Hong, B. Preneel, E. Biham, O. Dunkelman, and N. Keller, "Related-Key Boomerang and Rectangle Attacks", IACR eprint server, 2010/019 Jan 2010, `http://eprint.iacr.org/2010/019`.
11. L. May, M. Henricksen, W. Millan, G. Carter, and E. Dawson, "Strengthening the Key Schedule of the AES", ACISP 2002, LNCS 2384, pp. 226-240, Springer-Verlag, 2002.

12. N. Pramstaller, S. Mangard, S. Dominikus and J. Wolkerstorfer, "Efficient AES Implementations on ASICs and FPGAs", AES Conference, pp. 98-112, 2004.
13. A. Satoh, S. Morioka and S. Munetoh, "A Compact Rijndael Hardware Architecture with S-Box Optimization", ASIACRYPT 2001, LNCS 2248, pp. 239-254, Springer-Verlag, 2001.
14. Virtual Silicon Inc. 0.18 μm VIP Standard Cell Library Tape Out Ready, Part Number: UMCL18G212T3, Process: UMC Logic 0.18 μm Generic II Technology: 0.18μm, July 2004.
15. D. Wagner, "The Boomerang Attack", FSE 1999, LNCS 1636, pp. 156-170, Springer, Heidelberg, 1999.
16. H. Wu, "Related-Cipher Attacks", ICICS 2002, LNCS 2513, pp. 447-455, Springer, 2002.

A Proof of Lemma 1

A.1 Notation

Referring to Figure 1 in Appendix B, for $i = 0, 1, 2$, let $\Delta a_0^{(i)}$, $\Delta a_1^{(i)}$, and $\Delta a_2^{(i)}$ be the input differences to the SubBytes, MixColumns, and AddRoundKey operations respectively in the i^{th} round of the subkey generation. Also, let Δb be the difference in b at each round. Therefore, $\Delta a_0^{(0)}$ is the data input difference to the subkey generation function and Δb is the key input difference to each round of the subkey generation function, where $(\Delta a_0^{(0)})_j = \Delta a_j = \Delta S(MK_j)$ and $(\Delta b)_j = \Delta S(MK_{j+16})$. The output difference $\Delta a_0^{(3)}$ is the difference in the round subkey.

We make a few observations about these differences.

(1) After applying the SubBytes operation to the state, the positions of the active bytes are unchanged. The ShiftRows operation preserves the number of active bytes, so the input difference to the SubBytes operation $\Delta a_0^{(i)}$ and the output difference of the ShiftRows operation $\Delta a_1^{(i)}$ have the same number of active bytes.
(2) Furthermore, if $\Delta a_1^{(i)}$ has one active column, and it contains more than one active byte, ShiftRows^{-1} spreads them to different columns, so $\Delta a_0^{(i)}$ must have more than one active column, each containing one active byte.
(3) $\Delta a_2^{(i)} = \text{MixColumns}(\Delta a_1^{(i)})$. The MixColumns function is maximal distance separable, so its branch number is 5. Thus t active bytes in one column of $\Delta a_1^{(i)}$ spread to at least $5 - t$ active bytes in the same column of $\Delta a_2^{(i)}$. In particular, one active byte in $\Delta a_0^{(i)}$ gives one active byte in $\Delta a_1^{(i)}$ which spreads to one column of at least four active bytes in $\Delta a_2^{(i)}$.
(4) The AddRoundKey operation gives $\Delta a_0^{(i+1)} = \Delta a_2^{(i)} \oplus \Delta b$.

A.2 Proof of Lemma 1

Proof. We denote n and m to be the number of nonzero bytes in Δa and Δb respectively, and we write k and l for the number of nonzero bytes in $\Delta a_0^{(1)}$

and $\Delta a_0^{(2)}$ respectively. Then the number of active S-boxes in the differential characteristic path is $n + k + l$. We also note that if $\Delta MK \neq 0$, then $\Delta(a, b) \neq (0, 0)$. We consider the various cases below.

(1) $n = 0$

We have $\Delta a = 0$, so $\Delta b \neq 0$ and $m \neq 0$. Since $\Delta a_0^{(1)} = \Delta b$, we must have $k = m$.

If $\Delta a_2^{(1)}$ has one active column, then it has at least $5 - m$ active bytes, and the active bytes of $\Delta a_0^{(1)} = \Delta b$ are all in different columns. Then $\Delta a_0^{(2)} = \Delta a_2^{(1)} \oplus \Delta b$ has at least $5-m-1$ active bytes, i.e. $l \geq 4-m$. Then $n+k+l \geq 0 + m + 4 - m = 4$.

If $\Delta a_2^{(1)}$ has more than one active column, then, $\Delta a_2^{(1)}$ has at least $8 - m$ active bytes. If $m \geq 4$, we have $n + k + l \geq 4$. If $m \leq 3$, then $\Delta a_0^{(2)} = \Delta a_2^{(1)} \oplus \Delta b$ has at least $8 - 2m$ active bytes, i.e. $l \geq 8 - 2m$, which gives $n + k + l \geq 0 + m + 8 - 2m = 8 - m \geq 5$.

(2) $k = 0$

We have $\Delta b = \Delta a_2^{(0)}$, so $m \geq 5 - n$. We also have $\Delta a_0^{(2)} = \Delta b$, so $l = m$. Then $n + k + l \geq n + 0 + 5 - n = 5$.

(3) $n \geq 1, k \geq 1, l = 0$

From $l = 0$ we have $\Delta b = \Delta a_2^{(1)} = \text{MixColumns}\left(\Delta a_1^{(1)}\right)$.

If $k = 1$, $\Delta a_0^{(1)} = \Delta a_2^{(0)} \oplus \Delta b$ has one active byte. We have $\Delta a_2^{(0)} = \text{MixColumns}\left(\Delta a_1^{(0)}\right)$, and we can write $\Delta a_0^{(1)} = \text{MixColumns}(\alpha)$, where α has four active bytes. Equating the two expressions for Δb, we get $\Delta a_2^{(1)} = \Delta a_0^{(1)} \oplus \Delta a_2^{(0)}$, and by the linearity of MixColumns we get $\Delta a_1^{(0)} = \Delta a_1^{(1)} \oplus \alpha$. Then $\Delta a_1^{(0)}$ has at least three active bytes, as does $\Delta a_0^{(0)}$, and so $n \geq 3$, giving $n + k + l \geq 3 + 1 + 0 = 4$.

If $k = 2$, $\Delta b = \Delta a_2^{(1)}$ has either one or two columns active. If it has two columns active, then all eight bytes in the two columns are active, and we also know that $\Delta a_0^{(1)}$ has two active bytes. If $\Delta a_2^{(1)}$ has one column active, then at least three of the bytes in that column active, and the two bytes of $\Delta a_0^{(1)}$ must be in different columns. Either way, $\Delta a_2^{(0)} = \Delta a_0^{(1)} \oplus \Delta b$ has at least two active columns, so we must have $n \geq 2$. Then $n+k+l \geq 2+2+0 = 4$.

If $k \geq 3$, then because $n \geq 1$, we have $n + k + l \geq 4$.

(4) $n \geq 1, k \geq 1, l \geq 1$

We either have $n = k = l = 1$, or $n + k + l \geq 4$. Assume $n = k = l = 1$. Then $n + k + l = 3$, and since $\Delta a_2^{(0)}$ has four active bytes and $\Delta a_0^{(1)}$ has one active byte, $\Delta b = \Delta a_2^{(0)} \oplus \Delta a_0^{(1)}$ has at least three active bytes, i.e. $m \geq 3$. Since $n + m \geq 4$, we have at least four active S-boxes from the generation of a and b.

□

B Figures

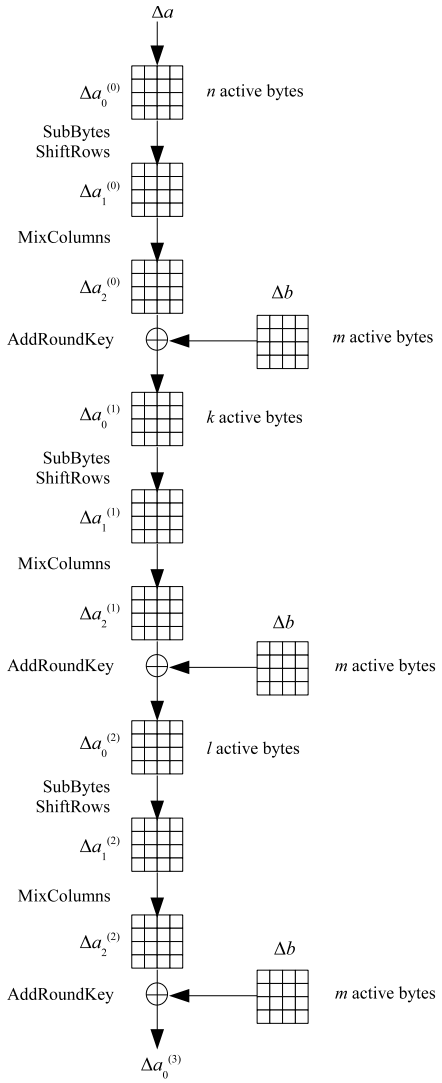

Fig. 1. Flow of differences for one round subkey generation