Tweaking AES

Ivica Nikolić

University of Luxembourg

Abstract. In this paper we present a tweak for the key schedule of AES in a form of a few additional basic operations such as rotations and S-boxes. This leads to a new cipher, which we call xAES, and which is resistant against the latest related-key differential attacks found in AES. xAES has a speed benchmark close to the one of AES even in the applications which use a frequent change of the master key.

Keywords: AES, tweak, key schedule.

1 Introduction

The Advanced Encryption Standard (AES) [6] is a block cipher adopted by NIST [15]. Eight years after the adoption, AES is widely used for commercial and governmental purposes while being implemented in both software and hardware. It is an elegant design and a very efficient cipher.

Recently, a few cryptanalytical results were obtained regarding the security resistance of AES [4,3]. It was shown that AES-192 and AES-256, i.e. the versions of AES with 192 and 256 key bits, do not have the ideal security level in the framework where related-key attacks are permitted. Despite the fact that so far these attacks are only theoretical and require a computational power beyond our reach, finding an efficient fix for AES that will produce a cipher that is ideal by the cryptographic standards, seems a good open problem.

In this paper we propose such fix. Since the recent attacks are mostly based on the property of the key schedule of AES, we tweak only this part of the cipher, while keeping intact the round function. We introduce only a few additional operations in the key schedule which result in a cipher that is: 1)resistant against related-key differential attacks, 2)has a speed close to the speed of AES.

The rest of the paper is structured as follows. In section 2 we give a brief facts on efficiency and security of AES. In section 3, we present our tweak for the key schedule. First, we focus on choosing a tweak that will produce a secure and efficient key schedule, then we prove the resistance of the new cipher against related-key differential attacks and finally we give theoretical and empirical estimates of its efficiency. In section 4 we conclude.

2 Efficiency and Security of AES

The block cipher AES has 128-bit state and supports three key sizes: 128, 192, and 256 bits. It is a byte oriented cipher and depending on the key size it has

A. Biryukov, G. Gong, and D.R. Stinson (Eds.): SAC 2010, LNCS 6544, pp. 198–210, 2011.

10 rounds for 128-bit keys, 12 rounds for 192-bit keys, and 14 rounds for 256-bit keys. In each round of AES the state, which can be seen as 4x4 matrix of bytes, undergoes four basic transformations:

1. SubBytes - bytewise application of S-boxes,
2. ShiftRows - cyclic shift of each row of the state matrix on some amount,
3. MixColumns - columnwise matrix multiplication,
4. AddRoundKey – xor of the subkey to the state.

The choice of these transformations permits to implement a round of AES as a combination of simple table lookups and xors (See [6]) leading to significant performance benefits. Moreover, Intel has announced that a new AES instruction set[11], called AES-NI, will be introduced in the new processors. Among other, the new instructions will significantly increase the efficiency of the round function of AES. Yet, no special instructions will be available to perform the key schedule routine.

Security analysis of AES has been the target of many cryptographic papers. The initial analysis was done by the submitters of Rijndael [6]. Using the property of the MixColumns transform, which is based on maximum distance separable code, the designers proved that *in the fixed-key model, differential characteristics exist only for a reduced number of rounds*. Hence the further analysis of AES was mainly focused either on other (non-differential) fixed-key attacks, or on related-key differential attacks.

In the fixed-key model, square attacks on 6 rounds of AES [6], boomerang attack on 6 rounds [2], collisions attacks on 7-8 rounds of AES [10,7], impossible differentials on 7-8 rounds of AES [13] and partial sum attacks [8] on 7,8,9 rounds of AES-128,-192,-256 respectively, were obtained.

In the related-key model, until recently, the best attacks were the boomerang and rectangle attacks found up to 10 rounds of AES-192 and 10 rounds of AES-256 [1,12,9]. The first attack on full-round AES-256 was given in [4]. In the paper, the authors present a related-key differential characteristic on all 14 rounds of AES-256, which leads to a key-recovery attack. Since some of the differences of the characteristic are in the subkey bytes which afterwards go through S-boxes, the attack works only for a class of keys. The second attack, in a form of related-key boomerang attack, on full-round AES [3] focuses on both AES-192 and AES-256. The attack leads to a key-recovery and works for all keys.

3 xAES – A New and Improved AES

In this section we present our proposal xAES. Although the security of the new version is our main concern, we would like to obtain an efficient primitive as well.

AES is widely implemented in both software and hardware. The round function is elegant with good security properties and it allows a fast implementation through a low number of table lookups and xors. The implementation is made even faster in software on the upcoming new Intel processors. Therefore, to gain

the necessary level of security for our proposal, we will focus on *improving the current key schedule of AES while keeping unchanged the round function.*

Let us define our main objectives. First, it is creating a new key schedule for AES such that no related-key differential characteristics exist on the full-round version of 128, 192, and 256 key sizes. We take a conservative approach and require that no such characteristics exist in any weak key class. Second, the new key schedule should be efficient – the speed of the new proposal should be comparable with the speed of AES. Note that when talking about the speed of a cipher, regarding the key agility we can go into two directions. One is to measure the efficiency of a cipher in *the encryption mode*, where the master key is fixed and the subkeys are computed once and used in all of the iterations. In this case, the efficiency of the key schedule is irrelevant[1] and the designer can spend a lot of computational power to produce the subkeys from the master key since the key setup is done only once. The second is when the cipher is used as an underlying primitive for other cryptographic constructions, e.g. hash functions. Then the master key is changed on every iteration, and sequentially, the subkeys have to be recomputed. In this case, the efficiency of the key schedule comes to a forefront and it has a significant impact on the efficiency of the whole cipher (the whole cryptographic construction). To measure the efficiency of a cipher, we take into account the second, more conservative direction, i.e. we measure the speed with the assumption that the master key is frequently changed – *a hash mode.*

Further, we present some ideas on possible approaches to build a secure and fast key schedule for AES. Then we present our proposal xAES, and give a security and efficiency analysis of the new cipher.

3.1 Methods to Improve the Key Schedule of AES

There are a few approaches to raise the security level of AES against related-key differential attacks. Further we present each approach and give an evaluation of its efficiency.

1. Increase the number of rounds. One can simply add a few more rounds at the top of the regular number of rounds of AES and obtain a cipher that is secure against related-key differential attacks. Note that the current key schedule of AES can easily produce a few more subkeys without the necessity of any substantial change. This approach has many positive sides, one of which is that the new cipher does not have to be reevaluated against the rest of the related-key non-differential attacks because the key schedule has not been changed. Yet, our second objective, efficiency, seems to suffer. Obviously each added round reduces the speed (in both the encryption and hash modes) by a factor of $\frac{1}{10}$, $\frac{1}{12}$, $\frac{1}{14}$ for AES-128, AES-192, and AES-256 respectively.
2. Create a key-schedule provably secure against differential attacks. One can design a key schedule full of S-box transformations and then prove the any related-key differential characteristic on the full number of rounds of the

[1] It is important only for encryption of short messages.

cipher, alone in the key schedule has a low probability because it has a high number of active S-boxes in the characteristic of the key schedule. This way, the related-key differential attacks on full rounds become impossible. A similar approach was used in [14] although the resistance of the key schedule was not formally proven. Again, with this approach, the designer meets our security objective, but might suffer a strong efficiency drawback in the hash mode due to the high number of S-boxes in the key schedule which may reduce the speed significantly.

3. Slightly change the current key schedule of AES, but keep unchanged the number of rounds. One can alter the key schedule by introducing additional (but small) number of S-boxes and/or other simple operations. These can be any operations that are sufficiently fast in software and hardware, e.g. ANDs, ORs, rotations, XORs, etc. This way the efficiency will not change significantly. Yet, in this approach, the proof of security against differential attacks is not trivial.

Further, we will use the third approach, i.e. we will introduce a small change in the current AES key schedule. This way we can easily meet our second objective – efficiency. To fulfill the security objective, we will analyze the resistance of the new cipher against related-key differential attacks using the tool for search of differential characteristics proposed in [5].

3.2 Specification of xAES

Now we can give a complete specification of our proposal xAES. Similarly to AES, xAES supports three key sizes: 128,192,256, denoted as xAES-128, xAES-192, and xAES-256 respectively. Although in [5] it was proven that no differential characteristic exist on the full round AES-128, we introduce xAES-128 to have a complete family of ciphers supporting the standard key sizes of 128,192, and 256 bits. The number of internal rounds in xAES for different key sizes is the same as the number of rounds in AES, i.e. 10 rounds for xAES-128, 12 rounds for xAES-192, and 14 rounds for xAES-256. Each internal round is defined same as in AES – through the four transformations SubBytes, ShiftRows, MixColumns and AddRoundKey. The only difference between AES and xAES is in the key schedule. Yet, the difference is small. In short, for obtaining each next column of the new subkey, xAES *always* uses rotation by one byte up of the previous subkey column, while AES uses a rotation only when obtaining the subkey column with an index multiple of N_k ($N_k = 4, 6, 8$ for AES-128,-192,-256). Let us give a formal definition of the new key schedules. We assume that the master key K is given as an array $K[4][N_k]$ and the key schedule produces a subkey array $W[4][4(N_r + 1)]$. The s-th subkey is given by the columns $4 \cdot s$ to $4 \cdot (s + 1) - 1$ of W. The round constant $RC[i][j]$ is the same as in AES.

The key schedule of xAES-128 is defined as follows. Let $K[4][4]$ be the master key. Then the subkey array $W[4][44]$ is defined as:

$$W[i][j] = \begin{cases} K[i][j], & \text{if } j < 4 \\ S(W[i-1 \bmod 4][j-1]) \oplus W[i][j-4] \oplus RC[i][j/4], & \text{if } j \bmod 4 == 0 \\ W[i-1 \bmod 4][j-1] \oplus W[i][j-4], & \text{otherwise} \end{cases}$$

The key schedule of xAES-192 is defined as follows. Let $K[4][6]$ be the master key. Then the subkey array $W[4][52]$ is defined as:

$$W[i][j] = \begin{cases} K[i][j], & \text{if } j < 6 \\ S(W[i-1 \bmod 4][j-1]) \oplus W[i][j-6] \oplus RC[i][j/4], & \text{if } j \bmod 6 == 0 \\ S(W[i-1 \bmod 4][j-1]) \oplus W[i][j-6], & \text{if } j \bmod 6 == 3 \\ W[i-1 \bmod 4][j-1] \oplus W[i][j-6], & \text{otherwise} \end{cases}$$

The key schedule of xAES-256 is defined as follows. Let $K[4][8]$ be the master key. Then the subkey array $W[4][60]$ is defined as:

$$W[i][j] = \begin{cases} K[i][j], & \text{if } j < 8 \\ S(W[i-1 \bmod 4][j-1]) \oplus W[i][j-8] \oplus RC[i][j/4], & \text{if } j \bmod 8 == 0 \\ S(W[i-1 \bmod 4][j-1]) \oplus W[i][j-8], & \text{if } j \bmod 8 == 4 \\ W[i-1 \bmod 4][j-1] \oplus W[i][j-8], & \text{otherwise} \end{cases}$$

To clearly understand the idea of the additional operations, in Fig. 1 we give a pictorial representation of how the $s+1$-th subkey is obtained from the s-th subkey, for all three key schedules of xAES as well as for the key schedule of AES-256 so the reader can compare the changes.

Besides the additional rotations, the difference between AES and xAES is in the 192-bit key version. We introduce additional layer of S-boxes to meet our security level.

3.3 Security Analysis of xAES

For a security analysis of xAES first we will focus on proving the resistance of xAES against related-key differential attacks. Then we will provide an evaluation of the security against other attacks.

Related-key Differential Attacks in xAES. Let us give a formal definition of an r-round related-key differential characteristic for a cipher. Let $E_K(P)$ be an R-round block cipher, where P is the plaintext and K is the master key. The key schedule function $KS(K)$ of the cipher $E_K(P)$ given key K, produces a set of (round) subkeys $K_0, \ldots, K_R$, i.e. $KS(K) = (K_0, \ldots, K_R)$. Let S_i be the value of state of $E_K(P)$ at the beginning of round i and $S_0 = P$ and let $E^1_{K_i}(S_i)$ be one round of the cipher. Let $p_K^{\Delta^K, \Delta_i^K}$ be probability that a given difference Δ^K in the master key K produces a set of differences $\Delta_i^K, i = 0, \ldots, R$ in the subkeys K_i, i.e.

$$p_K^{\Delta^K, \Delta_i^K} = \mathbf{P}(KS(K) \oplus KS(K \oplus \Delta^K) = (\Delta_0^K, \ldots, \Delta_R^K)) \tag{1}$$

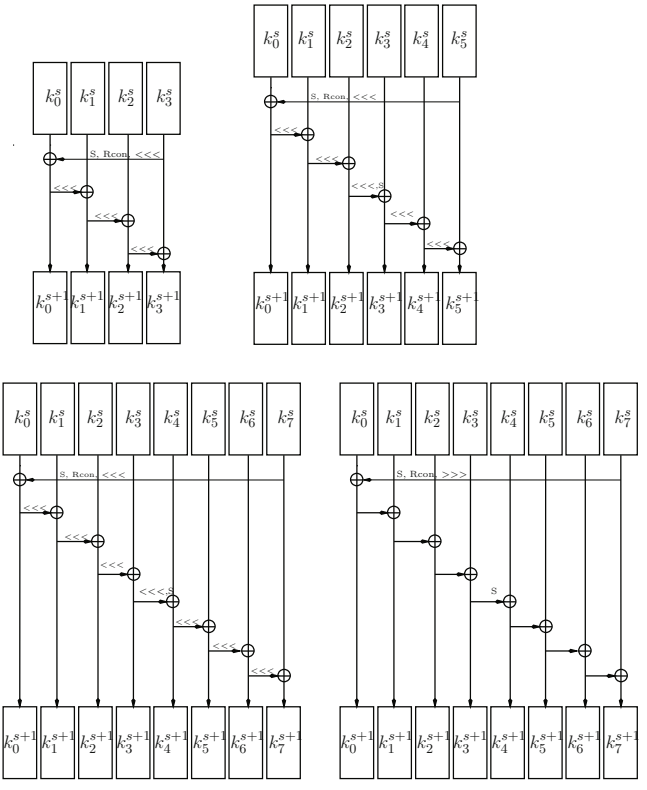

Fig. 1. One subkey round for xAES-128 (top left), xAES-192 (top right), xAES-256 (bottom-left), and AES-256 (bottom-right). Each k_i^j is a subkey column – 4 bytes. $\lll$ ($\ggg$) stand for a word rotation on 8 bits to the left(right), S for S-box, and Rcon for xor of the round constant.

Let $p_i^{\Delta_i^K,\Delta_i^S,\Delta_{i+1}^S}$ be the probability that a difference Δ_i^S in the state S_i and Δ_i^K in the subkey K_i produces difference Δ_{i+1}^S in the state S_{i+1}, i.e.

$$p_i^{\Delta_i^K,\Delta_i^S,\Delta_{i+1}^S} = \mathbf{P}(E^1_{K_i\oplus\Delta_i^K}(S_i\oplus\Delta_i^S)\oplus E^1_{K_i}(S_i)=\Delta_{i+1}^S), i=0,\ldots,R-1 \quad (2)$$

For a cipher $E_K(P)$ an r-round related-key differential characteristic is defined as a set $(\Delta_i^S, \Delta_i^K, p_i^{\Delta_i^K,\Delta_i^S,\Delta_{i+1}^S}, p_K^{\Delta^K,\Delta_i^K}), i=0,\ldots,r$ where (1),(2) are satisfied. The probability of this characteristic is given as:

$$p = p_K^{\Delta^K,\Delta_i^K}\cdot\prod_{0\le i<r} p_i^{\Delta_i^K,\Delta_i^S,\Delta_{i+1}^S} \quad (3)$$

To prove the resistance of xAES against related-key differential attacks, we will use the technique provided in [5]. In the paper, the authors specify a tool for

Algorithm 1. Search for RK differential characteristic

```
FirstRound()
{
for all ΔP do
  for all ΔS_1 do
    ΔK_0 = ΔP ⊕ ΔS_1
    Call NextRound(ΔS_1, ΔK_0, W(ΔS_1) + W(ΔK_0), 2)
  end for
end for
}

NextRound(ΔS_old, ΔK_old, w, r)
{
S̃ ← StateRound(ΔS_old)
for all S_new | W(S_new) + w + W_{n−r} ≤ W̃_n do
  ΔK_new ← ΔS̃ ⊕ ΔS_new
  if ΔK_new == SubkeyRound(ΔK_old) then
    if r == n then
      W̃_n ← w + W(ΔS_new) + W(ΔK_new)
    else
      Call NextRound(ΔS_new, ΔK_new, w + W(ΔS_new) + W(ΔK_new), r + 1)
    end if
  end if
end for
}
```

search of the best related-key differential characteristics in byte-oriented block ciphers. Depending on the key schedule of the analyzed cipher, the authors give three versions of the tool. Since, the key schedule of xAES is almost the same as the one in AES, we will use the same version as did the authors when analyzing AES. That is the version 2 of the tool. On Alg. 1 we give a short description of the tool in pseudo code. Given the weights W_i (which are actually the number of active S-boxes) of the best differential characteristics (the best is considered the characteristic that holds with highest probability) for the first $n-1$ rounds, and some weight $\tilde{W}_n$ of the n-round characteristic, the tool produces the best characteristic for n rounds. It starts with the procedure FirstRound, which fixes a certain difference ΔS_1 in the state at the begining of round 1, and a difference ΔK_0 in the whitening key. For each pair $(\Delta S_1, \Delta K_0)$ NextRound is called. This procedure, under certain weight conditions, extends the characteristic for an additional round as follows. First, from ΔS_{old} produces the difference $\Delta\tilde{S}$ in the state at the end of the round (just before the subkey addition). Then it takes all possible differences ΔS_{new} at the begining of the next round, and produces the difference ΔK_{new} in the subkey as an xor of $\Delta\tilde{S}$ and ΔS_{new} (recall the subkey addition defined in AES). Finally, it checks if ΔK_{new} can be obtained from ΔK_{old}, that is it checks if the difference in the following subkey can be produced from the difference in the previous subkey. The conditions on

the weights ($W(S_{new}) + w + W_{n-r} \leq \tilde{W}_n$) are introduced to stop extending unnecessarily some characteristics – the number of active Sboxes in the first r rounds ($W(S_{new}) + w$) plus the minimal number of active Sboxes in the rest $n-r$ rounds (W_{n-r}) should not exceed the number of active Sboxes of the best known characteristic ($\tilde{W}_n$)[2].

Since the only difference between AES and xAES is in the key schedule, to implement the tool for xAES, we have to find method that checks $\Delta K_{new} == SubkeyRound(\Delta K_{old})$, i.e. if the difference in the next subkey can be obtained from the difference of the previous subkey. Taking into consideration that the subkeys columns are produced one by one (see Fig. 1), this problem is reduced to the problem of checking if the differences in three subkey columns $k^i_j, k^i_{j+1}, k^{i-1}_{j+1}$ (where k^i_j, k^i_{j+1} are the j and $j+1$-th columns of the i-th subkey, and k^{i-1}_{j+1} is the $j+1$-th column of the $i-1$-th subkey) are related as:

$$(\Delta k^i_j) \lll 8 \oplus \Delta k^i_{j+1} = \Delta k^{i-1}_{j+1}. \tag{4}$$

The authors of the tool proposed a so-called compact representation of a difference in the subkey column. This difference is described with an 8-bit vector $\tilde{c} = (\tilde{a}, \tilde{b}) = (a_1, \ldots, a_4, b_1, \ldots, b_4), a_i, b_i \in \{0, 1\}$. The compact $\tilde{c}$ describes (is a set of) all subkey column differences $\Delta\tilde{d} = (\Delta d_1, \Delta d_1, \Delta d_2, \Delta d_3), \Delta d_i \in Z_{2^8}$, such that:

$$\Delta\tilde{d}^T = MC \cdot (x_1, x_2, x_3, x_4)^T \oplus (y_1, y_2, y_3, y_4)^T = MC \cdot \tilde{x}^T \oplus \tilde{y}^T, \tag{5}$$

where MC is the matrix of the MixColumns transform, $x_i, y_i \in Z_{2^8}$ and $x_i > 0$ **iff** $a_i > 0$, $y_i > 0$ **iff** $b_i > 0$. Note that for example the vector $(\tilde{0}, \tilde{b})$ is a simple truncated representation of a difference. In our implementation we will use the same compact representation for the differences in the subkey columns. To check (4) we have to deal with the rotation of the key Δk^i_j since in the key schedule of AES there is no such rotation. Let us see how this rotation influences the compact representation. If the difference Δk^i_j in the subkey is $\Delta\tilde{d}$ (see (5)), then the rotation of this difference is:

$$\Delta\tilde{d}^T \lll 8 = (MC \cdot \tilde{x}^T \oplus \tilde{y}^T) \lll 8 = (MC \cdot \tilde{x}^T) \lll 8 \oplus \tilde{y}^T \lll 8 \tag{6}$$

The matrix MC has the property $(MC \cdot X) \lll 8 = MC \cdot (X \lll 8)$. Hence (6) can be rewritten as:

$$\Delta\tilde{d}^T \lll 8 = MC \cdot (\tilde{x}^T \lll 8) \oplus \tilde{y}^T \lll 8 \tag{7}$$

Therefore, if the initial difference Δk^i_j had a compact representation $\tilde{c_1} = (a_1, a_2, a_3, a_4, b_1, b_2, b_3, b_4)$ then the rotation of this difference $\Delta k^i_j \lll 8$ has

[2] If it exceeds than this r round characteristic cannot be extend to n rounds because the next $n-r$ rounds will have at least W_{n-r} active S-boxes, so the total weight will be more than $\tilde{W}_n$.

a compact representation $\tilde{c_1} = (a_2, a_3, a_4, a_1, b_2, b_3, b_4, b_1)$. Hence, all the differences in (4) have a compact representation and this condition can easily be checked[3].

We have implemented the tool in practice and we searched for the best *round-reduced* related-key differential characteristics. To prove the resistance of r-round xAES against related-key differential attacks, these characteristics have to have certain properties:

1. No two characteristics exist with probability $2^{-p_1}, 2^{-p_2}$ on r_1 and r_2 rounds such that $r_1 + r_2 \geq r - 2$ and $2p_1 + 2p_2 \leq k$, where k is the key size. This restriction was introduced to stop the boomerang attacks on the full r rounds. We assume that two rounds can be obtained for free by various techniques, but the rest $r_1 + r_2$ rounds are part of the boomerang.
2. No characteristics exist on $r/2$ rounds with probability higher than $2^{-k/2}$. Obviously, this was introduced to stop the related-key differential attacks on the full r-round cipher which always can be seen as a concatenation of two $r/2$-round ciphers.

Each characteristic found by the tool is presented in a compact form. To find the probability of a characteristic one has to count the number of active bytes, i.e. the position of the bytes with 1 that go through the S-boxes. For example, a characteristics with s active bytes has a probability at most $2^{-6 \cdot s}$ because the maximal differential propagation of an S-box in AES is 2^{-6}. In table Tbl. 1 we give the probabilities (in terms of active bytes) of the best related-key differential characteristics for xAES-128, xAES-192, and xAES-256[4]. Using these results we can prove the differential resistance of xAES.

In case of xAES-128, to have a valid differential characteristic[5], the number of active bytes in the differential attack should not exceed $\lfloor \frac{128}{6} \rfloor = 21$ (because the key size is 128 bits and the maximal differential propagation of the S-box

Table 1. The number of active S-boxes in the best round-reduced related-key differential characteristics in xAES-128, xAES-192, and xAES-256

rounds	$xAES-128$	$xAES-192$	$xAES-256$
2	1	0	0
3	5	1	1
4	10	4	3
5	> 11	9	7
6	> 11	> 16	13
7	> 11	> 16	18
8	> 11	> 16	> 21

[3] Note that now checking (4) is reduced to checking for the case where all three subkey columns have a compact representation, which was solved for AES.

[4] For example, the best differential characteristic for xAES-192 on 5 rounds has 9 active S-boxes – in Tbl. 1 the intersection of row 5 and column xAES-192 is 9.

[5] A valid characteristic has a probability higher than 2^{-128}.

is 2^{-6}). In a boomerang attack (recall that since xAES-128 has 10 rounds, we try to build a boomerang on $10 - 2 = 8$ rounds), at least one of characteristics (upper or lower) has to be on 4 rounds, for which the attacker has to pay at least $2 \cdot 10 = 20$ active S-boxes. Hence, he has only $21 - 20 = 1$ active S-box left which is insufficient for a boomerang on 8 rounds and therefore xAES-128 is resistant against boomerang-type attacks. Now let us try to build a differential characteristic on all 10 rounds. Note that the characteristic on 5 rounds has more than 11 active S-boxes, hence the characteristic on 10 rounds would have more than $2 \cdot 11 = 22$, and therefore no related-key differential characteristic exist on all 10 rounds of xAES-128.

In xAES-192, the number of active S-boxes in a valid differential characteristic is bounded by $\lfloor \frac{192}{6} \rfloor = 32$. For a boomerang attack, in case one of the characteristics has 6 or more rounds, the number of active S-boxes becomes greater than 32, hence the attack does not work. When both characteristics have 5 rounds, then this number is $2 \cdot 9 + 2 \cdot 9 = 36$ which is again higher than 32, and therefore the boomerang with 10 rounds has lower probability than 2^{-192}. Similarly, any differential characteristic on 12 rounds (which can be seen as $6 + 6$ rounds), has more than $16 + 16 = 32$ active S-boxes, hence its probability is less than 2^{-192}.

In xAES-256, the number of active S-boxes is bounded by $\lfloor \frac{256}{6} \rfloor = 42$. Regarding the boomerang attack, when one of the characteristics is on at least 8 rounds, the attacker only for this characteristic has to pay more then $2 \cdot 21 = 42$ active S-boxes. When the characteristics are on 7 and 5 rounds, then the attacker pays $2 \cdot 18 + 2 \cdot 7 = 50$, while when both are on 6 rounds, he pays $2 \cdot 13 + 2 \cdot 13 = 52$ active S-boxes. In each of these cases, the total probability of the boomerang is less than 2^{-256}. The best characteristic on 7 rounds has only 18 active S-boxes. Hence, we cannot trivially prove that it does not exist a characteristic on 14 rounds (because $18 + 18 = 36$ which is less than 42). That is why we have to take a different approach. Any characteristic on 14 rounds, is composed of two 7-round characteristics, one of which has to have no more than 21 S-boxes (otherwise the total sum will be more than 42). First, we build all characteristics on 7 rounds that have no more than 21 active S-box. Then, we try to extend upward and downward each such characteristic for 7 additional rounds. Our search did not find any good candidate, i.e. no characteristic on 7 rounds with at most 21 active S-box can be extended to a characteristic on 14 rounds (with no more that 42 active S-boxes) and therefore no related-key differential characteristic exists on 14 rounds in xAES-256. This concludes our proof for the related-key differential resistance of full-round xAES.

The search for the best related-key differential characteristics for different version of xAES, i.e. running the tool in practice and obtaining the results of Tbl. 1, required different amount of computational effort. While finding the probabilities and the actual values for the round-reduced characteristics in xAES-128 and xAES-192 was performed in a few hours, in xAES-256 the search required a few days on a single core. The search for all good characteristics on 7 rounds of xAES-256 with no more than 21 active S-boxes produced around 2^{15} candidates and required two weeks on 8 cores. Extending these 7-round characteristics to

14 rounds, as mentioned before, did not give any good candidate, and required a few days on a single core.

Resistance of xAES Against Other Attacks. The round function of xAES is the same as the one of AES. Therefore all fixed-key attacks on xAES, which are based on vulnerabilities of the round function and do not exploit any properties of the key schedule, have the same security margin as in AES, that is the number of attacked rounds cannot be increased in xAES. This includes all of the fixed-key attacks mentioned in section 2.

3.4 Efficiency of xAES

Let us compare the speed of xAES with the speed of AES in software. We assume that both use optimal implementation with table lookups and xors. First let us give a theoretical comparison of the efficiency. Recall that since the round functions of AES and xAES are identical, in an environment where the master key is fixed and the encrypted message is longer (encryption mode), their speed is the same. Now let assume that the master key is frequently changed (hash mode). One round of AES (and therefore of xAES) has 16 table lookups and 16 xors.

One out of 10 subkey rounds of AES-128 has 4 table lookups and 8 xors. In xAES one such round has 4 table lookups, 8 xors and 3 rotations. If we assume that rotations have the same cost as xors, then for encrypting 16 bytes, AES uses $10 \cdot 16 + 10 \cdot 4 = 200$ lookups and $10 \cdot 16 + 10 \cdot 8 = 240$ operations. In xAES these numbers are $10 \cdot 16 + 10 \cdot 4 = 200$ lookups and $10 \cdot 16 + 10 \cdot 4 + 10 \cdot 3 = 270$ operations.

In AES-192, one subkey round (out of 8) has 4 table lookups and 10 xors while in xAES-192 one subkey round has 8 table lookups, 10 xors and 5 rotations. Hence, for 16 bytes, AES-192 spends $12 \cdot 16 + 8 \cdot 4 = 224$ lookups and $12 \cdot 16 + 8 \cdot 10 = 272$ operations, while xAES-192 spends $12 \cdot 16 + 8 \cdot 8 = 256$ lookups and 312 operations.

Finally, AES-256 has 7 subkey rounds and in each it has 8 table lookups and 16 xors. xAES-256 has the same number of subkey rounds and in each uses 8 table lookups , 16 xors and 7 rotations. For 16 bytes, AES-256 uses $14 \cdot 16 + 7 \cdot 8 = 280$ lookups and $14 \cdot 16 + 7 \cdot 16 = 336$ operations, while xAES-256 uses $14 \cdot 16 + 7 \cdot 8 = 280$ lookups and $14 \cdot 16 + 7 \cdot 16 + 7 \cdot 7 = 385$ operations.

These numbers indicate that one can expect that xAES-128 is around 6%, xAES-192 is around 12.5%, and xAES-256 is 7% slower than AES-128, AES-192, and AES-256 respectively. The actual implementation results obtained based on optimal implementation on C, together with the theoretical estimates, are given in Tbl. 2. In the table, the "Hash mode" entries indicate the speed of xAES compared to the speed of AES where the master key is changed on every iteration (16 bytes of plaintext), while the "Encryption mode" entries compare the speed when the master key is fixed[6]. The difference between the theoretical estimate and the actual implementation comes from the fact that the modern

[6] For short messages in the encryption mode, refer to the speed of "Hash mode".

Table 2. Comparison of the speed of xAES with the speed of AES. The abbreviations "tb" and "op" stand for table lookups and operations, respectively.

xAES/AES			
	128	192	256
	Hash mode		
Theoretical	$\frac{200\text{tl}+240\text{op}}{200\text{tl}+270\text{op}} \approx \approx 94\%$	$\frac{224\text{tl}+272\text{op}}{256\text{tl}+312\text{op}} \approx \approx 87\%$	$\frac{280\text{tl}+336\text{op}}{280\text{tl}+385\text{op}} \approx \approx 93\%$
Implementation	96%	83%	97%
	Encryption mode		
Theoretical, Implementation	100%	100%	100%

processors in one clock cycle can perform several xor operations but only one table lookup.

4 Conclusion

We have presented a tweak in the key schedule of AES that leads to a cipher that is resistant against the latest related-key differential attacks found in AES and it has the same security level against the other published attacks on AES. The resistance of xAES against the related-key attacks comes from the additional rotations of the columns in the key schedule which stop potential local collisions found in AES.

The low number of operations introduced by the tweak, keeps the speed of the new cipher on a level that is close to the speed of AES. More precisely, xAES is efficient as AES in the encryption mode, and slightly less efficient than AES in the hash mode.

Acknowledgements. The author would like to thank the anonymous reviewers of SAC 2010 for their valuable comments and Aleksandar and Antonio Nikolić for providing additional computational power. Ivica Nikolić is supported by the Fonds National de la Recherche Luxembourg grant TR-PHD-BFR07-031.

References

1. Biham, E., Dunkelman, O., Keller, N.: Related-key boomerang and rectangle attacks. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 507–525. Springer, Heidelberg (2005)
2. Biryukov, A.: The Boomerang Attack on 5 and 6-Round Reduced AES. In: Dobbertin, H., Rijmen, V., Sowa, A. (eds.) AES 2005. LNCS, vol. 3373, pp. 11–15. Springer, Heidelberg (2005)

3. Biryukov, A., Khovratovich, D.: Related-key cryptanalysis of the full AES-192 and AES-256. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 1–18. Springer, Heidelberg (2009)
4. Biryukov, A., Khovratovich, D., Nikolić, I.: Distinguisher and related-key attack on the full AES-256. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 231–249. Springer, Heidelberg (2009)
5. Biryukov, A., Nikolić, I.: Automatic search for related-key differential characteristics in byte-oriented block ciphers: Application to AES, Camellia, Khazad and others. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 322–344. Springer, Heidelberg (2010)
6. Daemen, J., Rijmen, V.: The Design of Rijndael: AES - The Advanced Encryption Standard. Springer, Heidelberg (2002)
7. Demirci, H., Selçuk, A.A.: A meet-in-the-middle attack on 8-round AES. In: Nyberg, K. (ed.) FSE 2008. LNCS, vol. 5086, pp. 116–126. Springer, Heidelberg (2008)
8. Ferguson, N., Kelsey, J., Lucks, S., Schneier, B., Stay, M., Wagner, D., Whiting, D.: Improved cryptanalysis of rijndael. In: Schneier, B. (ed.) FSE 2000. LNCS, vol. 1978, pp. 213–230. Springer, Heidelberg (2001)
9. Fleischmann, E., Gorski, M., Lucks, S.: Attacking 9 and 10 rounds of AES-256. In: Boyd, C., González Nieto, J. (eds.) ACISP 2009. LNCS, vol. 5594, pp. 60–72. Springer, Heidelberg (2009)
10. Gilbert, H., Minier, M.: A collision attack on 7 rounds of Rijndael. In: AES Candidate Conference, pp. 230–241 (2000)
11. Gueron, S.: Intel's new AES instructions for enhanced performance and security. In: Dunkelman, O. (ed.) FSE 2009. LNCS, vol. 5665, pp. 51–66. Springer, Heidelberg (2009)
12. Kim, J., Hong, S., Preneel, B.: Related-key rectangle attacks on reduced AES-192 and AES-256. In: Biryukov, A. (ed.) FSE 2007. LNCS, vol. 4593, pp. 225–241. Springer, Heidelberg (2007)
13. Lu, J., Dunkelman, O., Keller, N., Kim, J.: New impossible differential attacks on AES. In: Chowdhury, D.R., Rijmen, V., Das, A. (eds.) INDOCRYPT 2008. LNCS, vol. 5365, pp. 279–293. Springer, Heidelberg (2008)
14. May, L., Henricksen, M., Millan, W., Carter, G., Dawson, E.: Strengthening the key schedule of the AES. In: Batten, L.M., Seberry, J. (eds.) ACISP 2002. LNCS, vol. 2384, pp. 226–240. Springer, Heidelberg (2002)
15. National Institute of Standards and Technology. Advanced encryption standard (AES). FIPS 197 (November 2001)