A Meet-in-the-Middle Attack on 8-Round AES

Hüseyin Demirci[1] and Ali Aydın Selçuk[2]

[1] Tübitak UEKAE, 41470 Gebze, Kocaeli, Turkey
huseyind@uekae.tubitak.gov.tr
[2] Department of Computer Engineering
Bilkent University, 06800, Ankara, Turkey
selcuk@cs.bilkent.edu.tr

Abstract. We present a 5-round distinguisher for AES. We exploit this distinguisher to develop a meet-in-the-middle attack on 7 rounds of AES-192 and 8 rounds of AES-256. We also give a time-memory tradeoff generalization of the basic attack which gives a better balancing between different costs of the attack. As an additional note, we state a new square-like property of the AES algorithm.

Key words: AES, Rijndael, meet-in-the-middle cryptanalysis, square attack.

1 Introduction

In year 2000, the Rijndael block cipher was adopted by NIST as the Advanced Encryption Standard (AES), the new standard encryption algorithm of the US government to replace DES. The algorithm is a member of the family of square-type algorithms [7] designed by Vincent Rijmen and John Daemen. It is currently one of the most widely used and analyzed ciphers in the world.

AES is a 128-bit block cipher and accepts key sizes of 128, 192 and 256 bits. These versions of AES are called AES-128, AES-192 and AES-256 and the number of rounds for these versions are 10, 12 and 14 respectively. The algorithm is easy to understand, but the underlying mathematical ideas are strong. It has an SP-network structure. Interaction between the operations is chosen so that it satisfies full diffusion after two rounds. There is only one non-linear function in the algorithm, but it does not seem to have any considerable weakness so far.

AES has been remarkably secure against attacks. Some related key attacks can go up to 10 rounds on AES-192 and AES-256 with a complexity close to the complexity of exhaustive search. Attacks that are not of related-key type have been unable to go any further than 8 rounds. Most successful attacks in this class have been based on the square property observed by the designers of the Square algorithm [7].

In this paper we provide a distinguisher on 5 inner rounds of AES. This distinguisher relates a table entry of the fifth round to a table entry of the first round using 25 parameters that remain fixed throughout the attack. Using this distinguisher, we are able to attack up to 8 rounds of AES-256. For attacking

AES-192, we use a birthday-paradox-like approach to reduce the precomputation complexity, which enables a 7-round attack on AES-192. Our attack is also related to the meet-in-the-middle attack of Demirci et al. [9] on the IDEA block cipher, where a large sieving set is precomputed according to a certain distinguishing property of the cipher, and this set is later used to discover the round keys by a partial decryption.

This paper proceeds as follows: In Section 2 we briefly explain the AES block cipher and give a survey of the previous attacks. In Section 3, we review the 4-round AES distinguisher of Gilbert and Minier [12]. In Section 4, we introduce our 5-round distinguisher for AES. In Section 5, we describe our attacks on AES-192 and AES-256 based on this distinguisher. We conclude the paper with a summary of the results in Section 6. As an additional note, we present a novel square-like property of the AES algorithm in the appendix.

2 The AES Encryption Algorithm

The AES encryption algorithm organizes the plaintext as a 4×4 table of 1-byte entries, where the bytes are treated as elements of the finite field $GF(2^8)$. There are three main operations used in AES: the s-box substitution, shift row, and mix column operations. There is a single s-box substitution used for all entries of the table based on the inverse mapping in $GF(2^8)$ plus an affine mapping, which is known to have excellent differential and linear properties [19]. The shift row operation shifts the ith row i units left for $i = 0, 1, 2, 3$. Mix column operation is an MDS matrix multiplication which confuses the four entries of each column of the table. Key mixing is done at the end of each round where the bytes of the round key are XORed to the corresponding plaintext bytes of the table. Initially there is a key whitening before encryption begins, and in the final round there is no mix column operation. The key scheduling of AES is almost linear. Our analysis is independent of the key schedule algorithm. The interaction between the operations is designed in such a way that full diffusion is obtained after two rounds. Full details of the encryption, decryption, and key schedule algorithms can be found in [11].

AES has been remarkably resistant against attacks. Although different attacks have been tried on reduced-round versions, there is no way to break the actual cipher faster than exhaustive search. The algorithm designers applied the square attack [7] to the cipher. The attack uses about 2^{32} chosen plaintexts and breaks 6 rounds of AES with about 2^{72} complexity. The square attack has been improved [10] and the workload has been reduced to 2^{46}. For the key lengths 192 and 256 bits, the attack can be increased one more round with the help of the key schedule [18]. In [12] a collision attack has been applied to the cipher using a distinguishing property of the four-round encryption. With 2^{32} chosen plaintexts, the attack breaks 7 rounds of AES-192 and AES-256 with a complexity of 2^{140}. For AES-128, the attack is marginally faster than exhaustive search. The impossible differential attack has been applied up to 7 rounds of AES [3, 5, 22, 20, 21]; but the complexities of these attacks are higher than the square attack. Biryukov applied the boomerang attack technique to 5 and 6 rounds of the cipher [4]. For the 128 bit key length, the boomerang attack breaks 5 rounds of AES using 2^{46} adaptive chosen plaintexts in 2^{46} steps of analysis. The 6-round boomerang attack requires 2^{78} chosen plaintexts, 2^{78} steps of analysis, and 2^{36} bytes of memory. There is also a class of algebraic attacks applied on AES [6]. The authors write the AES S-box as a system of implicit quadratic equations. As a result, the cryptanalysis of the system turns out to be solving a huge system of quadratic equations. In [6], XSL method is suggested if the system of equations is overdefined and sparse which is the case for AES. Recently, related key attacks have been applied to the cipher $[1, 2, 15, 14, 17, 23]$. These attacks work up to 10 rounds of AES-192 and AES-256.

Throughout the paper, we use $K^{(r)}$ and $C^{(r)}$ to denote the round key and the ciphertext of the rth round; $K_{ij}^{(r)}$ and $C_{ij}^{(r)}$ denote the byte values at row i, column j. The arithmetic among table entries are in $GF(2^8)$, where addition is the same as bit-wise XOR. By a one round AES encryption, we mean an inner round without whitening or exclusion of the mixcolumn operation unless otherwise stated.

2.1 The Square Property

The square attack [7] is the first attack on AES, which was applied by the designers of the algorithm [8]. Proposition 1 states the distinguishing property the square attack exploits.

Throughout this paper by an active entry, we mean an entry that takes all byte values between 0 and 255 exactly once over a given set of plaintexts. By a passive entry we mean an entry that is fixed to a constant byte value.

Proposition 1 ([8]). *Take a set of 256 plaintexts so that one entry in the plaintext table is active and all the other entries are passive. After applying three rounds of AES, the sum of each entry over the 256 ciphertexts is 0.*

This property leads to a straightforward attack on 4 rounds of AES where the last round key is searched and decrypted and the third round outputs are checked for this property. This attack can be extended one round from the top and one round from the bottom so that 6 rounds of AES can be attacked using this property $[7, 10]$.

The idea behind the square attack still forms the basis of most of the analysis on AES. Therefore, obtaining more square-like properties of the cipher is essential for evaluating its security. We state such a new square-like property of the AES algorithm in the appendix.

3 A 4-Round Distinguisher of AES

In [12], Gilbert and Minier showed an interesting distinguishing property for 4 rounds of AES: Consider the evolution of the plaintext over 4 inner rounds, with no whitening. Let a_{ij} denote the ith row, jth column of the plaintext. After the first s-box transformation, define $t_{ij} = S(a_{ij})$. At the end of round 1, our state matrix is of the form:

$2t_{11} + c_1$	m_{12}	m_{13}	m_{14}
$t_{11} + c_2$	m_{22}	m_{23}	m_{24}
$t_{11} + c_3$	m_{32}	m_{33}	m_{34}
$3t_{11} + c_4$	m_{42}	m_{43}	m_{44}

where m_{ij} and c_i, $1 \leq i \leq 4$, $2 \leq j \leq 4$, are fixed values that depend on the passive entries and subkey values. At the end of the second round, this gives

$$\begin{aligned} C_{11}^{(2)} &= 2S(2t_{11} + c_1) + 3S(m_{22}) + S(m_{33}) + S(m_{44}) + K_{11}^{(2)} \\ &= 2S(2t_{11} + c_1) + c_5, \end{aligned}$$

for some fixed value c_5. Similarly we can get the other diagonal entries as:

$$\begin{aligned} C_{22}^{(2)} &= S(3t_{11} + c_4) + c_6 \\ C_{33}^{(2)} &= 2S(t_{11} + c_3) + c_7 \\ C_{44}^{(2)} &= S(t_{11} + c_2) + c_8 \end{aligned}$$

Since $C_{11}^{(3)} = 2C_{11}^{(2)} + 3C_{22}^{(2)} + C_{33}^{(2)} + C_{44}^{(2)} + K_{11}^{(3)}$, we can summarize the above observations with the following proposition:

Proposition 2 ([12]). *Consider a set of 256 plaintexts where the entry a_{11} is active and all the other entries are passive. Encrypt this set with 3 rounds of AES. Then, the function which maps a_{11} to $C_{11}^{(3)}$ is entirely determined by 9 fixed 1-byte parameters.*

Proof. To write the equation for $C_{11}^{(3)}$, the constants c_i, $1 \leq i \leq 8$, and $K_{11}^{(3)}$ are required. Therefore, the nine fixed values

$$\left(c_1, c_2, \ldots, c_8, K_{11}^{(3)}\right)$$

completely specify the mapping $a_{11} \rightarrow C_{11}^{(3)}$. $\square$

Proposition 2 can be generalized: Note that the argument preceding the proposition applies to any other third round ciphertext entry and hence the statement is true for any $C_{ij}^{(3)}$. Similarly, any other a_{ij} can be taken as the active byte instead of a_{11}.

Gilbert and Minier [12] observed that the constants c_1, c_2, c_3, and c_4 depend on the values (a_{21}, a_{31}, a_{41}) on the first column, whereas the other constants c_5, c_6, c_7, and c_8 are independent of these variables. They used this information to find collisions over 3 rounds of the cipher: Assume that c_1, c_2, c_3, and c_4 behave as random functions of the variables (a_{21}, a_{31}, a_{41}). If we take about 2^{16} random (a_{21}, a_{31}, a_{41}) values and fix the other passive entries of the plaintext, by the birthday paradox, two identical functions $f, f' : a_{11} \rightarrow C_{11}^{(3)}$ will be obtained with a non-significant probability by two different values of (a_{21}, a_{31}, a_{41}). This distinguishing property was used to build attacks on AES up to 7 rounds.

Through a 1-round decryption, we get the following distinguisher for 4-round AES:

Proposition 3 ([12]). *Consider a set of 256 plaintexts where the entry a_{11} is active and all the other entries are passive. Apply 4 rounds of AES to this set. Let the function S^{-1} denote the inverse of the AES s-box and $k^{(4)}$ denote $0E \cdot K_{11}^{(4)} + 0B \cdot K_{21}^{(4)} + 0D \cdot K_{31}^{(4)} + 09 \cdot K_{41}^{(4)}$. Then,*

$$S^{-1}[0E \cdot C_{11}^{(4)} + 0B \cdot C_{21}^{(4)} + 0D \cdot C_{31}^{(4)} + 09 \cdot C_{41}^{(4)} + k^{(4)}]$$

is a function of a_{11} determined entirely by 1 key byte and 8 bytes that depend on the key and the passive entries. Thus,

$$0E \cdot C_{11}^{(4)} + 0B \cdot C_{21}^{(4)} + 0D \cdot C_{31}^{(4)} + 09 \cdot C_{41}^{(4)}$$

is a function of a_{11} determined entirely by 10 constant bytes.

4 A 5-Round Distinguisher of AES

In this section, we show how the observations of Gilbert and Minier [12] can be extended to 5 rounds. To the best of our knowledge, this is the first 5-round distinguishing property of AES. This property will help us to develop attacks on 7 rounds of AES-192 and AES-256, and on 8 rounds of AES-256.

Proposition 4. *Consider a set of 256 plaintexts where the entry a_{11} is active and all the other entries are passive. Encrypt this set with 4 rounds of AES. Then, the function which maps a_{11} to $C_{11}^{(4)}$ is entirely determined by 25 fixed 1-byte parameters.*

Proof. By Proposition 2, in the third round we have

$$\begin{aligned} C_{11}^{(3)} = {} & 2S(2S(2t_{11} + c_1) + c_5) + 3S(2S(2t_{11} + c_4) + c_6) \\ & + S(S(t_{11} + c_3) + c_7) + S(S(t_{11} + c_2) + c_8) + K_{11}^{(3)}. \end{aligned} \tag{1}$$

Similarly it can be shown that

$$\begin{aligned} C_{22}^{(3)} = {} & S(S(3t_{11} + c_4) + c_9) + 2S(3S(2t_{11} + c_3) + c_{10}) \\ & + 3S(S(t_{11} + c_2) + c_{11}) + S(3S(2t_{11} + c_1) + c_{12}) + K_{22}^{(3)}, \end{aligned} \tag{2}$$

$$\begin{aligned} C_{33}^{(3)} = {} & S(S(t_{11} + c_3) + c_{13}) + S(2S(t_{11} + c_2) + c_{14}) \\ & + 2S(S(2t_{11} + c_1) + c_{15}) + 3S(2S(3t_{11} + c_4) + c_{16}) + K_{33}^{(3)} \end{aligned} \tag{3}$$

$$\begin{aligned} C_{44}^{(3)} = {} & 3S(S(t_{11} + c_2) + c_{17}) + S(S(2t_{11} + c_1) + c_{18}) \\ & + S(3S(3t_{11} + c_4) + c_{19}) + 2S(S(t_{11} + c_3) + c_{20}) + K_{44}^{(3)}. \end{aligned} \tag{4}$$

Since

$$C_{11}^{(4)} = 2S(C_{11}^{(3)}) + 3S(C_{22}^{(3)}) + S(C_{33}^{(3)}) + S(C_{44}^{(3)}) + K_{11}^{(4)}, \tag{5}$$

the fixed values

$$\left(c_1, c_2, \ldots, c_{20}, K_{11}^{(3)}, K_{22}^{(3)}, K_{33}^{(3)}, K_{44}^{(3)}, K_{11}^{(4)}\right) \tag{6}$$

are sufficient to express the function $a_{11} \to C_{11}^{(4)}$. □

Although each of the diagonal entries depend on 9 fixed parameters, it is interesting to observe that the fourth round entry $C_{11}^{(4)}$ is entirely determined by 25 variables, rather than 37. This is a result of the fact that the constants c_1, c_2, c_3 and c_4 are common in formulas (1–4) of all the diagonal entries. Note that, like Proposition 2, Proposition 4 can also be generalized to any entry.

Since this 4-round property is related to a single entry, we can develop a 5-round distinguisher by considering the fifth round decryption:

Proposition 5. *Consider a set of 256 plaintexts where the entry a_{11} is active and all the other entries are passive. Apply 5 rounds of AES to this set. Let the function S^{-1} denote the inverse of the AES S-box and $k^{(5)}$ denote $0E \cdot K_{11}^{(5)} + 0B \cdot K_{21}^{(5)} + 0D \cdot K_{31}^{(5)} + 09 \cdot K_{41}^{(5)}$. Then,*

$$S^{-1}[0E \cdot C_{11}^{(5)} + 0B \cdot C_{21}^{(5)} + 0D \cdot C_{31}^{(5)} + 09 \cdot C_{41}^{(5)} + k^{(5)}]$$

is a function of a_{11} determined entirely by 5 key bytes and 20 bytes that depend on the key and the passive entries. Thus,

$$0E \cdot C_{11}^{(5)} + 0B \cdot C_{21}^{(5)} + 0D \cdot C_{31}^{(5)} + 09 \cdot C_{41}^{(5)}$$

is a function of a_{11} determined entirely by 26 constant bytes.

25 bytes may be too much to search exhaustively in an attack on AES-128; but for AES-256, we can precalculate and store all the possible values of this function, and using this distinguisher we can attack on 7 and 8 rounds. For AES-192, we can apply a time-memory tradeoff trick to reduce the complexity of the precomputation of the function over these 25 parameters and to make the attack feasible for 192-bit key size.

5 The Attack on AES

In this section, we describe a meet-in-the-middle attack on 7-round AES based on the distinguishing property observed in Section 4. In the attack, we first precompute all possible $a_{11} \to C_{11}^{(4)}$ mappings according to Proposition 4. Then we choose and encrypt a suitable plaintext set. We search certain key bytes, do a partial decryption on the ciphertext set, and compare the values obtained by this decryption to the values in the precomputed set. When a match is found, the key value tried is most likely the right key value. The details of the attack are as follows:

1. For each of the $2^{25\times 8}$ possible values of the parameters in (6), calculate the function $a_{11} \rightarrow C_{11}^{(4)}$, for each $0 \leq a_{11} \leq 255$, according to equations (1–4) and (5).
2. Let K_{init} denote the initial whitening subkey blocks $(K_{11}^{(0)}, K_{22}^{(0)}, K_{33}^{(0)}, K_{44}^{(0)})$. Try each possible value of K_{init}, and choose a set of 256 plaintexts accordingly to satisfy that the first entry takes every value from 0 to 255 and all other entries are fixed at the end of round 1. Also search $K_{11}^{(1)}$ to guess the value of $C_{11}^{(1)}$. Encrypt this set of plaintexts with 7 rounds of AES.
3. Let K_{final} denote the subkey blocks $(K_{11}^{(7)}, K_{24}^{(7)}, K_{33}^{(7)}, K_{42}^{(7)}, k^{(6)})$, where $k^{(6)}$ denotes $0E \cdot K_{11}^{(6)} + 0B \cdot K_{21}^{(6)} + 0D \cdot K_{31}^{(6)} + 09 \cdot K_{41}^{(6)}$. Search over all possible values of K_{final}. Using K_{final}, do a partial decryption of the ciphertext bytes $C_{11}^{(7)}$, $C_{24}^{(7)}$, $C_{33}^{(7)}$ and $C_{42}^{(7)}$ to obtain the entry $C_{11}^{(5)}$ over the set of 256 ciphertexts obtained in Step 2.
4. Now if the K_{init} and K_{final} subkeys are guessed correctly, the function $C_{11}^{(1)} \rightarrow C_{11}^{(5)}$ must match one of the functions obtained in the precomputation stage. Compare the sequence of the 256 $C_{11}^{(5)}$ values obtained in Step 3 to the sequences obtained in precomputation. If a match is found, the current key is the correct key by an overwhelming probability, since the probability of having a match for a wrong key is approximately $2^{8\times 25}\, 2^{-2048} = 2^{-1848}$.
5. Repeat the attack three times with different target values, $C_{21}^{(5)}$, $C_{31}^{(5)}$, and $C_{41}^{(5)}$, instead of $C_{11}^{(5)}$, using the same plaintext set. Having already discovered K_{init}, this attack gives us another 15 key bytes from the final two rounds.
6. Now having recovered most of the key bytes, we can search the remaining key bytes exhaustively.

This attack requires 2^{32} chosen plaintexts where the first column of the plaintext takes every possible value and the rest remain constant. There is a precomputation step which calculates 2^{200} possible values for 256 plaintexts. Therefore the complexity of this step, which will be done only once, is 2^{208} evaluations of the function. In the key search phase, for every combination of K_{init}, $K_{11}^{(1)}$, and K_{final}, we do partial decryption over 256 ciphertexts which makes 2^{88} partial decryptions. As in [7] and [10], we assume that 2^8 partial decryptions take approximately the time of a single encryption. Therefore the processing complexity of the attack is comparable to 2^{80} encryptions.

Note that since we take the target entries used in Step 5 to be on the same column as $C_{11}^{(5)}$, such as $C_{21}^{(5)}$, equations (1–4) will remain identical in these computations, and the only change will be on a few coefficients in equation (5). Hence, there won't be a need for a separate precomputation; the necessary values for $C_{11}^{(1)} \rightarrow C_{21}^{(5)}$ can be obtained with a slight overhead. However, we will need separate memory to store the obtained values. Hence, the memory requirement of the attack is $4 \times 2^{208} = 2^{210}$ bytes, which is equivalent to 2^{206} AES blocks.

5.1 A Time-Memory Tradeoff

The cost of the attack above is dominated by generation of the function set in the precomputation phase. A time-memory tradeoff approach can be useful here to balance the costs: Instead of evaluating all the possible functions in the precomputation phase, we can evaluate and store only a part of the possible function space. On the other hand, we must repeat the key search procedure a number of times with different plaintext sets to compensate the effect of this reduction. In general, if we reduce the size of the function set by a factor of n_1 and repeat the key search procedure for each candidate key n_2 times, for some $n_1, n_2 > 1$, the probability of having a match for the right key becomes, for relatively large n_1,

$$1 - \left(1 - \frac{1}{n_1}\right)^{n_2} \approx 1 - e^{-\frac{n_2}{n_1}}, \tag{7}$$

which means a success probability of 63% for $n_2 = n_1$ and 98% for $n_2 = 4n_1$.

By this tradeoff approach, one can balance different costs of the attack. The attack's complexity is currently dominated by the complexity of the precomputation phase and the required storage. As seen in Table 1, the basic attack is not feasible on AES-192. By the tradeoff approach, the precomputation cost can be reduced as desired, and the attack becomes feasible on AES-192 for $n_1 > 2^{16}$ (i.e., $n > 16$).

5.2 Extension to 8 Rounds

To attack 8 rounds of AES, we follow exactly the same steps of the 7-round attack, but we also search the last round key exhaustively. Therefore the data, precomputation, and storage complexities do not change, whereas the complexity of the key search phase increases by a factor of 2^{128}. Hence the time complexity of the attack on 8-round AES becomes 2^{208} while the memory complexity is 2^{206}. Although this attack appears to be dominated by Hellman's [13] time-memory tradeoff on both counts, it is a non-trivial attack faster than exhaustive search on 8-round AES-256.

The performance of our attacks and the previous attacks on AES are summarized in Table 1. Related key attacks, which are a different phenomenon, are not included in the comparison.

As seen in Table 1, the complexity of our attacks includes a precomputation cost in addition to the regular time complexity. The precomputation cost is considered separately from the rest of the time complexity due to the fact that it is executed only once at the time of initialization. The precomputation costs are given in terms of one evaluation of the $a_{11} \rightarrow C_{11}^{(4)}$ function according to equations (1–5).

5.3 An Improved Attack

Orhun Kara [16] observed the following improvement on the attack we described above: In the partial decryption phase of the attack in Step 3 where the attacker

Block Cipher	Paper	Rounds	Type	Complexity Data	Memory	Time	Pre.
AES-192	[12]	7	Collision	2^{32}	2^{84}	2^{140}	2^{84}
	[21]	7	Imp. Differential	2^{92}	2^{153}	2^{186}	–
	[18]	7	Square	2^{32}	2^{32}	2^{184}	–
	[10]	7	Square	$19 \cdot 2^{32}$	2^{32}	2^{155}	–
	[10]	7	Square	$2^{128} - 2^{119}$	2^{64}	2^{120}	–
	This paper	7	MitM	2^{32}	2^{206}	2^{72}	2^{208}
	This paper	7	MitM-TM	2^{34+n}	2^{206-n}	2^{74+n}	2^{208-n}
	[10]	8	Square	$2^{128} - 2^{119}$	2^{64}	2^{188}	–
AES-256	[18]	7	Square	2^{32}	2^{32}	2^{200}	–
	[12]	7	Collision	2^{32}	2^{84}	2^{140}	2^{84}
	[10]	7	Square	$21 \cdot 2^{32}$	2^{32}	2^{172}	–
	[10]	7	Square	$2^{128} - 2^{119}$	2^{64}	2^{120}	–
	[21]	7	Imp. Differential	$2^{92.5}$	2^{153}	$2^{250.5}$	–
	This paper	7	MitM	2^{32}	2^{206}	2^{72}	2^{208}
	This paper	7	MitM-TM	2^{34+n}	2^{206-n}	2^{74+n}	2^{208-n}
	[10]	8	Square	$2^{128} - 2^{119}$	2^{104}	2^{204}	–
	This paper	8	MitM	2^{32}	2^{206}	2^{200}	2^{208}
	This paper	8	MitM-TM	2^{34+n}	2^{206-n}	2^{202+n}	2^{208-n}

Table 1. Plaintext, memory, time, and precomputation time complexities of the chosen plaintext attacks on AES-192 and AES-256. "MitM" stands for a meet-in-the-middle attack; "MitM-TM" denotes the time-memory tradeoff version of the attack as described in Section 5.1. Here we assume that if the precomputed set is reduced by a factor of 2^n, the key search procedure is repeated 2^{n+2} times to compensate for this reduction.

checks the partial ciphertext values of round 5, if the attacker looks at the XOR of two partial ciphertexts rather than looking at them individually, the $k^{(5)}$ term in the equation cancels and he does not need to include this term in the key search.

This improved variant of the attack works as follows: In the precomputation phase, for f denoting the mapping $a_{11} \rightarrow C_{11}^{(4)}$, the attacker computes and stores,

$$S(f(i)) + S(f(0)),$$

rather than storing $f(i)$, for $1 \leq i \leq 255$. And, accordingly, in the key search phase, he looks for this XORed value in the precomputed set.

In this new variant, the key search complexity is reduced by a factor of 2^8, to 2^{72} for the 7-round attack and to 2^{200} for the 8-round attack. The complexity figures in Table 1 are given to reflect this improvement on the basic attack.

6 Conclusion

We have shown that if only one entry of a set of plaintexts is active while the other 15 entries are passive, each entry of the ciphertext after 4 rounds of AES encryption can be entirely defined using 25 fixed bytes. Using this property, we have developed the first 5-round distinguisher of AES. This enabled us to develop attacks on 7 and 8 rounds of AES-256 and 7 rounds of AES-192. The attack has a huge precomputation and memory complexity, but the data and time complexities are comparable with the best existing attacks. We have used a birthday paradox approach to reduce the precomputation and memory complexities. The proposed attacks present a new way of utilizing square-like properties for attacking AES.

Acknowledgments

We would like to thank Çağdaş Çalık for a helpful discussion on time-memory tradeoff attacks and Orhun Kara for his careful review and many valuable comments on the paper.

References

1. E. Biham, O. Dunkelman, and N. Keller. Related-key and boomerang attacks. In *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 507–525. Springer-Verlag, 2005.
2. E. Biham, O. Dunkelman, and N. Keller. Related-key impossible differential attacks on AES-192. In *CT-RSA 2006*, volume 3860 of *LNCS*, pages 21–31. Springer-Verlag, 2006.
3. E. Biham and N. Keller. Cryptanalysis of reduced variants of Rijndael. In *The Third AES Candidate Conference*, 2000.
4. A. Biryukov. Boomerang attack on 5 and 6-round AES. In *The Fourth Conference on Advanced Encryption Standard*, 2004.
5. J. H. Cheon, M. J. Kim, K. Kim, J. Lee, and S. Kang. Improved impossible differential cryptanalysis of Rijndael. In *ICISC '2001*, volume 2288 of *LNCS*, pages 39–49. Springer-Verlag, 2001.
6. N. T. Courtois and J. Pieprzyk. Cryptanalysis of block ciphers with overdefined systems of equations. In *ASIACRYPT 2002*, volume 2501 of *LNCS*, pages 267–287. Springer-Verlag, 2002.
7. J. Daemen, L. Knudsen, and V. Rijmen. The block cipher SQUARE. In *FSE 1997*, volume 1267 of *LNCS*, pages 149–165. Springer-Verlag, 1997.
8. J. Daemen and V. Rijmen. AES proposal: Rijndael. In *The First AES Candidate Conference*, 1998.
9. H. Demirci, A. A. Selçuk, and E. Türe. A new meet in the middle attack on IDEA. In *SAC 2003*, volume 3006 of *LNCS*, pages 117–129. Springer-Verlag, 2004.
10. N. Ferguson, J. Kelsey, S. Lucks, B. Schneier, M. Stay, D. Wagner, and D. Whiting. Improved cryptanalysis of Rijndael. In *FSE 2000*, volume 1978 of *LNCS*, pages 213–230. Springer-Verlag, 2001.
11. FIPS PUB 197. NIST.

12. H. Gilbert and M. Minier. A collision attack on 7 rounds of Rijndael. In *The Third AES Candidate Conference*, 2000.
13. M. E. Hellman. A cryptanalytic time-memory trade-off. *IEEE Trans. Information Theory*, 26(4):401–406, 1980.
14. S. Hong, J. Kim, S. Lee, and B. Preneel. Related-key rectangle attacks on reduced versions of SHACAL-1 and AES-192. In *FSE 2005*, volume 3557 of *LNCS*, pages 368–383. Springer-Verlag, 2005.
15. G. Jakimoski and Y. Desmedt. Related-key differential cryptanalysis of 192-bit key AES variants. In *SAC 2003*, volume 3006 of *LNCS*, pages 208–221. Springer-Verlag, 2004.
16. O. Kara. Personal communication.
17. J. Kim, S. Hong, and B. Preneel. Related-key rectangle attacks on reduced AES-192 and AES 256. In *FSE 2007*, volume 4593 of *LNCS*, pages 225–241. Springer-Verlag, 2007.
18. S. Lucks. Attacking seven rounds of Rijndael under 192-bit and 256-bit keys. In *The Third AES Candidate Conference*, 2000.
19. K. Nyberg and L.R Knudsen. Provable security against a differential attack. *Journal of Cryptology*, 8(1):27–38, 1995.
20. R. C. W. Phan. Classes of impossible differentials of Advanced Encryption Standard. *IEE Electronics Letters*, 38(11):508–510, 2002.
21. R. C. W. Phan. Impossible differential cryptanalysis of 7-round Advanced Encryption Standard AES. *Information Processing Letters*, 91:33–38, 2004.
22. R. C. W. Phan and M.U Siddiqi. Generalized impossible differentials of Advanced Encryption Standard. *IEE Electronics Letters*, 37(14):896–898, 2001.
23. W. Zhang, W. Wun, L. Zhang, and D. Feng. Improved related-key impossible differential attacks on reduced round AES-192. In *SAC 2006*, volume 4356 of *LNCS*, pages 15–27. Springer-Verlag, 2007.

A A Semi-Square Property of AES

In this section we present a semi-square property of the AES encryption algorithm. This property observes the effect of fixing a certain bit position over the diagonal entries.

Proposition 6. *Take a set of AES plaintexts where all the non-diagonal entries are fixed. For the diagonal entries, choose a certain bit position and fix that bit of all the four diagonal entries; vary the remaining bits and obtain the set of all possible* $(2^7)^4$ *values of these plaintexts. Apply three rounds of AES to this set. Then the sum of each table entry over the ciphertext set obtained will be 0.*

One can use this semi-square property as a distinguisher to develop attacks on AES. Instead of one active entry used in the square attack, the semi-square property uses 4 semi-active entries. Therefore, the semi-square property is less efficient in terms of the required data amount. Also it is difficult to increase the number of rounds in an attack since it uses the diagonal entries. On the other hand, it is interesting to observe the effect of fixing a one-bit position. Although the s-box of AES is perfect in terms of linear and differential properties, some structural properties can still be tracked if we fix a one-bit position. This property is not preserved if we fix two or more bit positions. Understanding the mechanism behind this observation can help us to deduce more square-like properties of the cipher. This example illustrates that square properties are not restricted to just the cases where all possible values of one cell are enumerated.