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

Hüseyin Demirci
Ali Aydın Selçuk
presented by
Orhun Kara

Outline

- The AES
- A 5-round distinguisher
- Attack on 7-round AES-192, AES-256 and 8-round AES-256
- Some optimizations – birthday paradox approach
- An improved attack
- Semi-square property
- Conclusion

AES Operations

- AES S-box: Uses x^{-1} plus an affine mapping.

- Shift Row Operation: Shift the ith row $(i-1)$ units left for $i = 1, 2, 3, 4$.

$$\begin{array}{|c|c|c|c|}\hline P_{11} & P_{12} & P_{13} & P_{14} \\ \hline P_{21} & P_{22} & P_{23} & P_{24} \\ \hline P_{31} & P_{32} & P_{33} & P_{34} \\ \hline P_{41} & P_{42} & P_{43} & P_{44} \\ \hline \end{array} \rightarrow \begin{array}{|c|c|c|c|}\hline P_{11} & P_{12} & P_{13} & P_{14} \\ \hline P_{22} & P_{23} & P_{24} & P_{21} \\ \hline P_{33} & P_{34} & P_{31} & P_{32} \\ \hline P_{44} & P_{41} & P_{42} & P_{43} \\ \hline \end{array}$$

- Mix Column Operation: Multiply each column with the following matrix:

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 02 & 02 \end{bmatrix}$$

- Add Round Key Operation

AES

- The initial whitening

- For $i = 1$ to $r - 1$, do:
  - S-Box substitution
  - Shift Row
  - Mix Column
  - Add Round Key

- For the final round, do:
  - S-Box substitution
  - Shift Row
  - Add Round Key

- Key Scheduling: Uses recursive operations. If 16 (24, 32) consecutive bytes of the subkey are known, one can get all the subkey values of AES-128 (AES-192, AES-256).

Attacks on AES

AES is designed considering classical differential and linear cryptanalysis. Structural mechanisms can be exploited. Square properties, impossible differentials, collision properties of the inner variables have been used for cryptanalysis.

Square-like Attacks

A chosen-plaintext attack, where a certain byte a_{ij} of the plaintext takes every value $0 \leq a_{ij} \leq 255$ over the plaintext set.

This a_{ij} is called the “active” byte. Other, fixed input bytes are “passive”.

Distinguishers can be discovered for such plaintext sets. E.g., the “Square Property”:

Proposition 1 (Daemen & Rijmen) *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.*

A 3-Round Distinguisher

Gilbert & Minier (2000):

Consider the inner rounds of AES (i.e., no whitening). Take a plaintext set, where a_{11} is active and the other bytes are passive. At the end of round 1, the state matrix is:

$2t_{11}+c_1$	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 $t_{11} = S(a_{11})$, and m_{ij} and c_i are fixed values that depend on the passive entries and subkey values. At the end of the second round, this gives

$$\begin{aligned}
C_{11}^{(2)} &= 2S(2t_{11}+c_1)+c_5,\\
C_{22}^{(2)} &= S(3t_{11}+c_4)+c_6,\\
C_{33}^{(2)} &= 2S(t_{11}+c_3)+c_7,\\
C_{44}^{(2)} &= S(t_{11}+c_2)+c_8.
\end{aligned}$$

At the end of the third round, we have

$$C_{11}^{(3)} = 2S(C_{11}^{(2)}) + 3S(C_{22}^{(2)}) + S(C_{33}^{(2)}) \\ + S(C_{44}^{(2)}) + K_{11}^{(3)}.$$

Hence, for such a plaintext set,

$$a_{11} \to C_{11}^{(3)}$$

is completely specified by 9 fixed parameters:

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

A New 4-Round Distinguisher

Proposition 2 *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 $f : a_{11} \rightarrow C_{11}^{(4)}$ is entirely determined by 25 fixed 1-byte parameters:*

Proof. Each of $C_{11}^{(3)}$, $C_{22}^{(3)}$, $C_{33}^{(3)}$, $C_{44}^{(3)}$ depends on 9 fixed parameters and t_{11}. The mapping,

$$a_{11} \rightarrow C_{11}^{(4)},$$

where

$$\begin{aligned} C_{11}^{(4)} &= 2S(C_{11}^{(3)}) + 3S(C_{22}^{(3)}) \\ &\quad + S(C_{33}^{(3)}) + S(C_{44}^{(3)}) + K_{11}^{(4)}, \end{aligned}$$

depends on t_{11} and, due to the overlaps, only on 25 fixed parameters, rather than 37:

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

Extension to 5 Rounds

Use 1-round decryption to express $C_{11}^{(4)}$:

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

is a function of a_{11} determined entirely by 25 fixed bytes, where $k^{(5)}$ denotes $0E \cdot K_{11}^{(5)} + 0B \cdot K_{21}^{(5)} + 0D \cdot K_{31}^{(5)} + 09 \cdot K_{41}^{(5)}$.

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.

A MitM Attack on 7-Round AES

1. (Precomputation)
For each different value of the 25-byte parameter set, compute

$$a_{11} \to C_{11}^{(4)},$$

for each $0 \leq a_{11} \leq 255$, according to Proposition 2.

2. Search $K_{init} = (K_{11}^{(0)}, K_{22}^{(0)}, K_{33}^{(0)}, K_{44}^{(0)})$; choose an appropriate set of 256 plaintexts to obtain the desired starting value at the end of round 1. Also search for $K_{11}^{(1)}$ to obtain $C_{11}^{(1)}$. Encrypt this set with 7-round AES.

3. Search $K_{final} = (K_{11}^{(7)}, K_{24}^{(7)}, K_{33}^{(7)}, K_{42}^{(7)}, k^{(6)})$ for each K_{init} tried, do a partial decryption of the ciphertext set, and obtain a set of 256 $C_{11}^{(5)}$.

4. If K_{init} and K_{final} are correct, the function $C_{11}^{(1)} \rightarrow C_{11}^{(5)}$ will match one of the functions obtained in the precomputation stage. If it doesn't, eliminate that key. At the end, the process will result in 10 discovered key bytes.

5. Repeat the attack with other target values and obtain other key bytes from K_{final} to find a dominant part of the subkey values.

6. Search the remaining key bytes exhaustively.

Complexity of the Attack

AES	192/7	256/7	256/8
Data	2^{32}		
Precomp.	2^{208}		
Memory	2^{206}		
Key Search	2^{80}	2^{80}	2^{208}

Complexity of the attack on 7-round AES is dominated by the precomputation phase and the memory requirement. This can be reduced by a time-memory tradeoff approach.

A Time-Memory Tradeoff

Instead of covering every possible function for $f : a_{11} \to C_{11}^{(4)}$, we can choose to cover a certain fraction of this set, and repeat the plaintext search several times to compensate for it.

If we reduce the precomputation by a factor of n_1 and repeat the plaintext search n_2 times, probability of catching the right key is about

$$1 - e^{-\frac{n_2}{n_1}}$$

which is 98% for $n_2 = 4n_1$.

Improved Attack

AES	192/7	256/7	256/8
Data	2^{34+n}		
Precomp.	2^{208-n}		
Memory	2^{206-n}		
Key Search	2^{82+n}	2^{82+n}	2^{210+n}

where we assume, for some n, $n_1 = 2^n$ and $n_2 = 4n_1$.

It is possible to choose n so that none of the complexities exceed 2^{192} for attacking 7 rounds of AES-192.

An Improved Attack – by Orhun Kara

Consider the partial decryption to obtain $C_{11}^{(4)}$:

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

We can get rid of $k^{(5)}$ if we take XOR of two partial ciphertexts.

Hence, in the precomputation phase, for $1 \leq i \leq 255$, store

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

rather than $f(i)$, and look for this XOR in the precomputed set in the key search phase.

Then the key search complexity is reduced to 2^{72} for the 7-round attack, and to 2^{200} for the 8-round.

Semi-Square Property of AES

Proposition 3 *Take a set of* $(2^7)^4$ *plaintexts so that all the non-diagonal entries are fixed. For the diagonal entries, choose certain bit position and fix that bit of all the four entries, and vary the other diagonal positions over every possible combination. Apply 3 rounds of AES to this set. Then, the sum of each entry over the ciphertext set will be 0.*

How to Exploit Semi-square Property

- An attack can trivally be based on this distinguisher
- But it is less efficient than normal Square Attack
- Square property uses one active entry (256 plaintexts)
- Semi-square property uses 4 semi-active entries ($(2^7)^4$ plaintexts)
- It is also difficult to increase the number of rounds since it uses diagonal entries.
- It is interesting to observe the leakage of information through the strong S-box when 1-bit position is fixed.

Conclusion

- Developed the first 5-round distinguisher of AES.
- Attacked 7 rounds of AES-192 and 7 & 8 rounds of AES-256.
- Also presented a new semi-square property of AES.
- The meet-in-the-middle attack presents a new way of exploiting square-like properties of AES.