A MEET IN THE MIDDLE ATTACK ON 8-ROUND AES

Almog Apirion

Huseyn Demirci & Ali Aydin Selcuk

outline

- Introduction
- We will present 5-round distinguisher for AES
  - Relates a table entry of the fifth round to a table entry of the first round using 25 parameters that remain fixed
- Using this distinguisher to develop a meet-in-the-middle attack
  - 7 rounds of AES-192 and AES-256
  - 8 rounds of AES-256
- time-memory tradeoff
  - generalization of the basic attack which gives a better balancing between different costs of the attack

Introduction

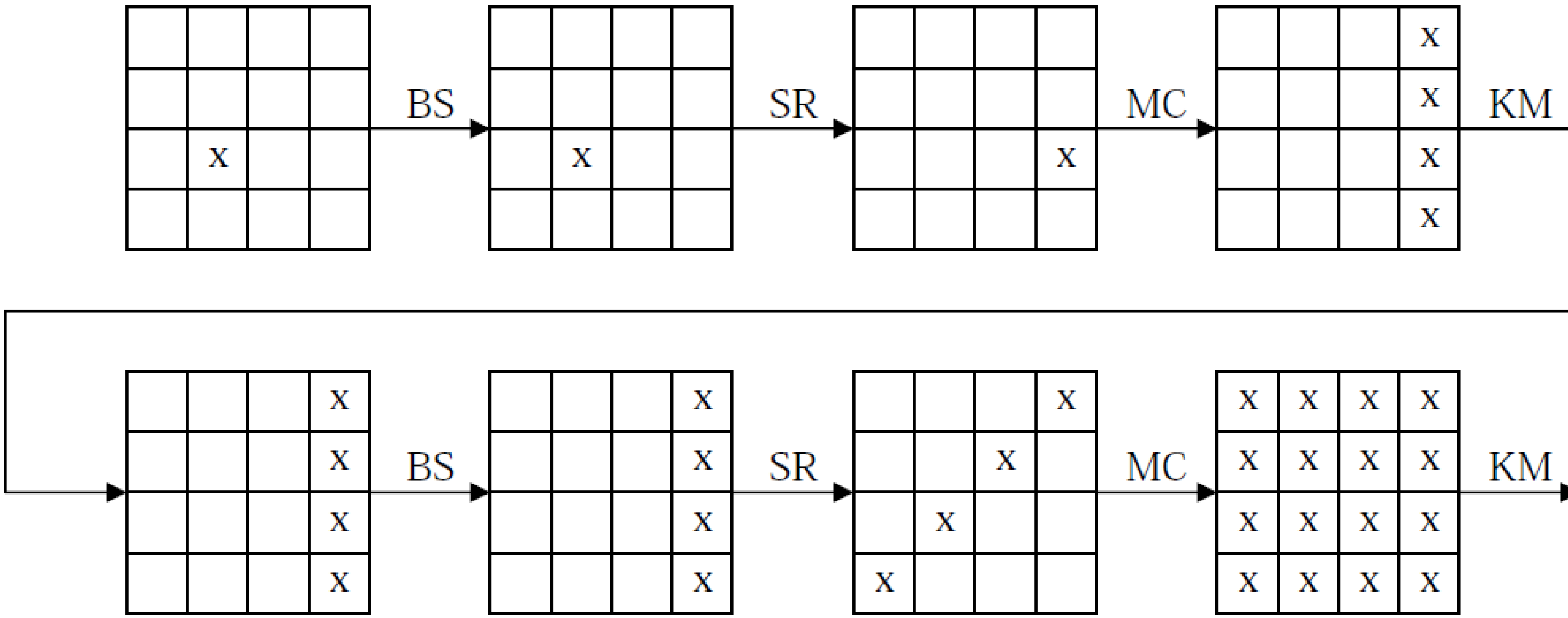
- In year 2000, the Rijndael was adopted by NIST as the **A**dvanced **E**ncryption **S**tandard
  - Currently one of the most widely used and analyzed ciphers in the world
  - only one non-linear function
  - Does not seem to have any considerable weaknesses so far
- AES-128, AES-192 and AES-256
  - number of rounds 10, 12 and 14 respectively
  - full diffusion after two rounds

Introduction

- AES Flash DEMO

Introduction

- Full diffusion after two rounds



Attack on AES

- AES has been remarkably resistant against attacks
- Related key attacks on AES 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**

Square Attacks

- Also called Saturation attacks
- Most powerful cryptanalysis of AES to date was by this method
- Exploits the **byte-oriented structure** of the cipher
- Could break a reduced AES version using only 7 rounds of encryption
- But is faster than exhaustive key search

Attack on AES

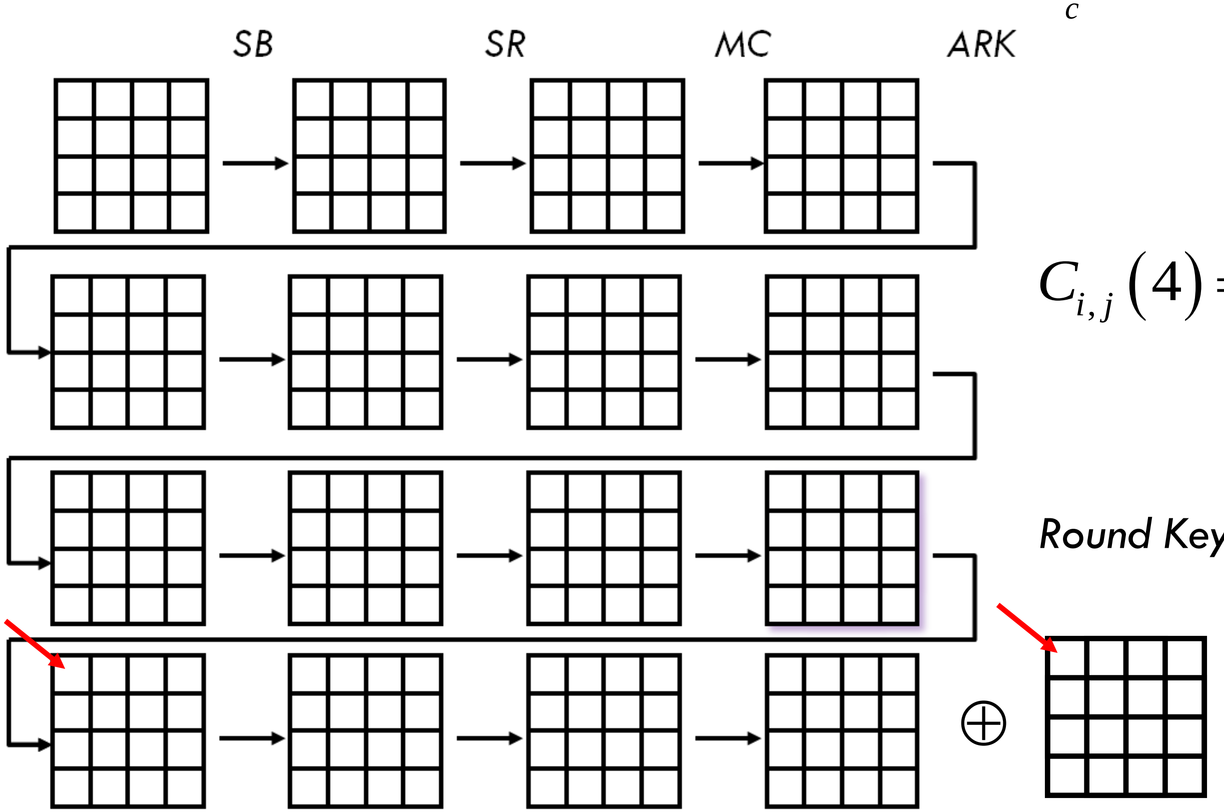
- The square property on AES, was observed by the designers
  - If one byte is modified in the plaintext, then exactly 4 are modified after one round, and all the 16 are modified after two rounds
    - The same property holds in decryption
- **Conclusion**: one-byte difference cannot lead to a one-byte difference after three rounds

The Square Property

- Attacks using the square property exploits the following property **(Proposition1)**:
- Take a set of 256 plain texts 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 cipher texts is 0

The Square Property

- leads to a straightforward attack on 4 rounds of AES
- the last round key is searched and decrypted and the third round outputs are checked for this property



$$C_{i,j}(4) = S[C_{j,i}(3)] \oplus K_{i,j}(4)$$

Definitions

- $K(r)$-the round key of the r-th round
- $C(r)$-the cipher text of the r-th round
- $C_{i,j}(r)$,$K_{i,j}(r)$- byte values at row i , column j
- **Addition** is the same as bit-wise XOR
- one (inner) **round** AES encryption, round without whitening or exclusion of the mix column operation
- **Active entry**- an entry that takes all byte values between 0 and 255 exactly once
- **Passive entry**- an entry that is fixed to a constant byte value

A 4-Round Distinguisher of AES

- Denote a_{ij} the i-th row, j-th column of the plaintext
- Let $t_{ij} = S(a_{ij})$ be that byte after the first s-box transformation
- At the end of round 1:

$2t_{11} + c_1$	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}

- m_{ij} and c_i are fixed values that depend on the passive entries and sub-key values
- At the end of the second round: $$\begin{aligned} C_{11}^{(2)} &= 2S(2t_{11}+c_1) + 3S(m_{22}) + S(m_{33}) + S(m_{44}) + K_{11}^{(2)} \\ &= 2S(2t_{11}+c_1) + c_5, \end{aligned}$$
- we can get the other diagonal entries as: $$\begin{aligned} C_{22}^{(2)} &= S(3t_{11}+c_4) + c_6 \\ C_{33}^{(2)} &= 2S(t_{11}+c_3) + c_7 \\ C_{44}^{(2)} &= S(t_{11}+c_2) + c_8 \end{aligned}$$
- $C_{11}^{(3)} = 2C_{11}^{(2)} + 3C_{22}^{(2)} + C_{33}^{(2)} + C_{44}^{(2)} + K_{11}^{(3)}$

A 4-Round Distinguisher of AES

- **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 3 rounds ,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 $a_{11} \rightarrow c_{11}^{(3)}$.
  - that the argument applies to any other third round cipher-text entry:

$$a_{ij} \rightarrow c_{ij}^{(3)}$$

A 4-Round Distinguisher of AES

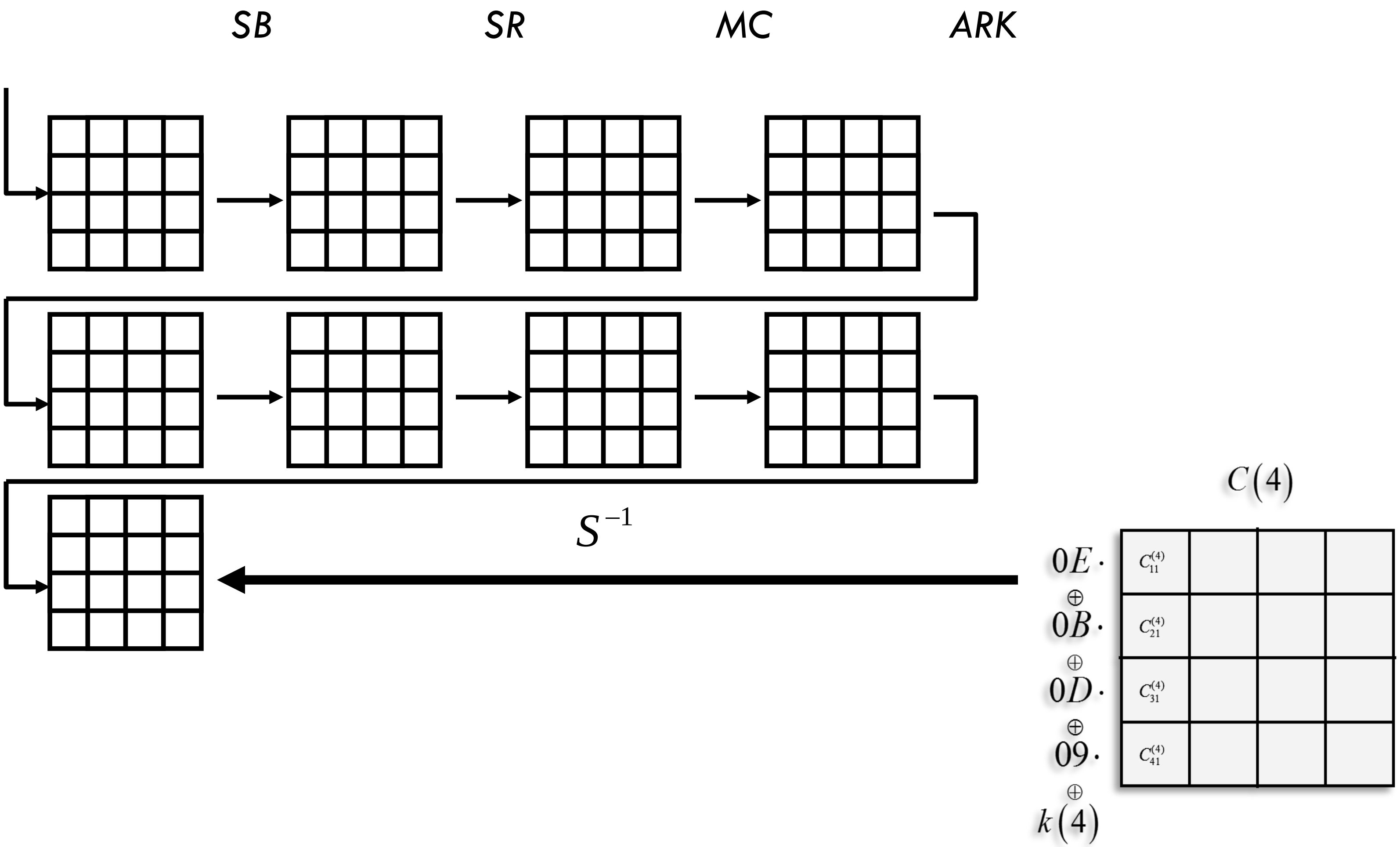
- **Proposition 3** :Consider a set of 256 plaintexts where the entry a_{11} is active and all the other entries are passive
- Apply 4 rounds of AES to this set
- function S^{-1} denote the inverse of the AES s-box and $k^{(4)}$ denote: $0E \cdot K_{11}^{(4)} + 0B \cdot K_{21}^{(4)} + 0D \cdot K_{31}^{(4)}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

A 4-Round Distinguisher of AES

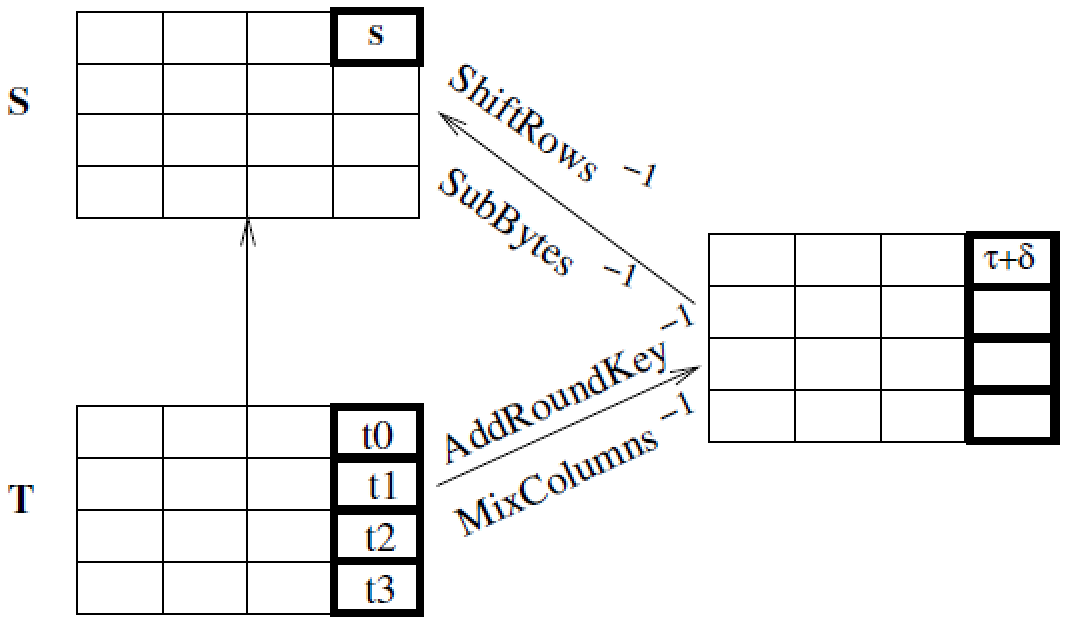
$$C(4) \qquad\qquad\qquad\qquad\qquad\qquad K(4)$$

$$\begin{array}{c} 0E\cdot \\ \oplus \\ 0B\cdot \\ \oplus \\ 0D\cdot \\ \oplus \\ 09\cdot \\ \oplus \\ k(4) \end{array} \begin{bmatrix} C_{11}^{(4)} & & & \\ C_{21}^{(4)} & & & \\ C_{31}^{(4)} & & & \\ C_{41}^{(4)} & & & \end{bmatrix} \quad k(4) = \begin{array}{c} 0E\cdot \\ \oplus \\ 0B\cdot \\ \oplus \\ 0D\cdot \\ \oplus \\ 09\cdot \end{array} \begin{bmatrix} K_{11}^{(4)} & & & \\ K_{21}^{(4)} & & & \\ K_{31}^{(4)} & & & \\ K_{41}^{(4)} & & & \end{bmatrix}$$

A 4-Round Distinguisher of AES



A 4-Round Distinguisher of AES



A 5-Round Distinguisher of AES

- The pervious observations can be extended to 5 rounds
- 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 $a_{11} \rightarrow c_{11}^{(4)}$ is entirely determined by 25 fixed1-byte parameters

A 5-Round Distinguisher of AES

- **Proposition 4 - proof:**
  - By Proposition 2 , in the third round:

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

  - Similarly it can be shown that

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

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

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

  - Since

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

  - to express the function $a_{11} \rightarrow c_{11}^{(4)}$ the following fixed values are sufficient:

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

A 5-Round Distinguisher of AES

- Although each of the diagonal entries depend on 9 fixed parameters, the fourth round entry $C_{11}^{(4)}$ is entirely determined by25 variables(rather than 9*4+1=37),This is a result of the fact that the constants c1, c2, c3 and c4 are common in formulas (1-4) of all the diagonal entries
  - that argument applies to any other cipher-text 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**

A 5-Round Distinguisher of AES

- **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
- 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

A 5-Round Distinguisher of AES

- **AES-128**,25 bytes may be too much to search
- **AES-256**, we can pre-calculate and store all the possible values of this function and using this distinguisher we can attack on 7 and 8 rounds
- **AES-192**, we can apply a time-memory tradeoff trick to reduce the complexity of the pre-computation of the function over these 25 parameters and to make the attack feasible for 192-bit key size

The Attack on AES

- **MITM attack on 7-round AES outline:**
  - First we pre-compute all possible $a_{11} \rightarrow c_{11}^{(4)}$ mappings
  - Then we choose and encrypt a suitable plaintext set
  - We search certain key bytes
    - Do a partial decryption on the cipher-text set
  - Compare the values obtained by this decryption to the values in the pre-computed set
  - When a match is found the key value tried is most likely the right key value

The Attack on AES – step 1

- For each of the 2^{25x8} values of $\left(c_1, c_2, \ldots, c_{20}, K_{11}^{(3)}, K_{22}^{(3)}, K_{33}^{(3)}, K_{44}^{(3)}, K_{11}^{(4)}\right)$

calculate the function $a_{11} \rightarrow c_{11}^{(4)}$ for each $0 \le a_{11} \le 255$

$C_{11}^{(4)} = 2S(C_{11}^{(3)}) + 3S(C_{22}^{(3)}) + S(C_{33}^{(3)}) + S(C_{44}^{(3)}) + K_{11}^{(4)}$ according to equations (1-4)

a(7)	a(6)	a(5)	a(4)	a(3)	a(2)	a(0)	a(1)
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1
…	…	…					…
1	1	1	1	1	1	1	1

X

c1	c2	…	c20	k11(3)	k22(3)	k33(3)	k44(3)	k11(4)	C11(4)
0	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	1	
…	…	…	…	…	…	…	…	…	
255	255	255	255	255	255	255	255	255	

The Attack on AES – step 2

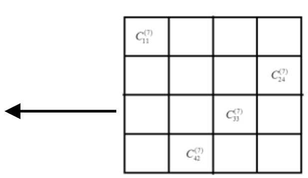
- Let K_{init} be the initial whitening subkey blocks $(K_{11}^{(0)}, K_{22}^{(0)}, K_{33}^{(0)}, K_{44}^{(0)})$,Try each possible value of K_{init}
- choose a set of 256 plaintexts accordingly to satisfy that the first entry takes every value from 0 to 255 and all other entries are fixed at the end of round 1
- Search $K_{11}^{(1)}$ to guess the value of $C_{11}^{(1)}$
- Encrypt this set of plaintexts with 7 rounds of AES.

The Attack on AES – step 3

- Let K_{final} be 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 cipher-text $(C_{11}^{(7)}, C_{24}^{(7)}, C_{33}^{(7)}, C_{42}^{(7)})$ to obtain the entry $C_{11}^{(5)}$ over the set of 256 cipher-texts obtained in Step 2

K_{final}

K11(7)	K24(7)	K33(7)	K42(7)	k(6)
0	0	0	0	0
0	0	0	0	1
…	…	…	…	…
255	255	255	255	255



The Attack on AES – step 4

- If K_{final} and K_{init} subkeys are guessed correctly the function $a_{11} \rightarrow c_{11}^{(5)}$ must match one of the functions obtained in the pre-computation stage
  - Compare the sequence of the 256 $C_{11}^{(5)}$ from step 3 to the sequences obtained in pre-computation
  - If a match is found, the current key is the correct key
    - the probability of having a match for a wrong key: $2^{25x8} x 2^{-2048} = 2^{-1848}$

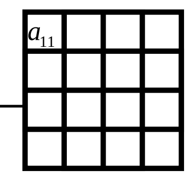
The Attack on AES – steps 5&6

- **Step 5**: Repeat the attack three times with different target values $C_{21}^{(5)}, C_{31}^{(5)}$ and $C_{31}^{(5)}$
  - using the same plaintext set
  - already discovered K_{init}
- this attack gives us another 15 key bytes from the final two rounds
- **Step 5**: Now having recovered most of the key bytes, we can search the remaining key bytes exhaustively

The Attack on AES

K_{init}

K11(0)	K22(0)	K33(0)	K44(0)	K11(1)
0	0	0	0	1
0	0	0	0	0
…	…	…	…	…
255	255	255	255	255



c1	c2	…	c20	k11(3)	k22(3)	k33(3)	k44(3)	k11(4)	C11(4)
0	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	1	
…	…	…	…	…	…	…	…	…	
255	255	255	255	255	255	255	255	255	

X

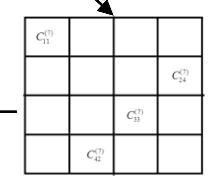
a(7)	a(6)	a(5)	a(4)	a(3)	a(2)	a(0)	a(1)
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1
…	…	…					…
1	1	1	1	1	1	1	1

Compare the sequence of the 256 sequences obtained in pre-computation
If a match is found, the current key is the correct key

K_{final}

K11(7)	K24(7)	K33(7)	K42(7)	k(6)
0	0	0	0	0
0	0	0	0	1
…	…	…	…	…
255	255	255	255	255

$C^{(7)}$



The Attack on AES- analysis

- 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 pre-computation step which calculates 2^{200} possible values for 256 plaintexts , Therefore the complexity of this step, which will be done only once, is 2^{200} 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 cipher-texts which makes 2^{88} partial decryptions
  - we assume that 2^{8} partial decryptions take approximately the time of a single encryption
- Therefore the processing complexity is comparable to 2^{80} encryptions

The Attack on AES- analysis

- The target entries used in Step 5 to be on the same column as $C_{11}^{(5)}$
  - $C_{21}^{(5)}, C_{31}^{(5)}, C_{41}^{(5)}$
  - equations (1- 4) will remain identical in these computations, and the only change will be on a few coefficients in equation (5).
- There won't be a need for a separate pre-computation
  - The necessary values for $a_{11} \rightarrow c_{21}^{(5)}$ 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 $4X\,2^{208} = 2^{210}$ bytes, which is equivalent to 2^{206} AES blocks

A Time-Memory Tradeoff

- The cost of the attack is dominated by generation of the function set in the pre-computation phase
- A time-memory tradeoff can balance the costs
  - Instead of evaluating all the possible functions in the pre-computation 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
- 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 $(n_1, n_2 > 1)$

A Time-Memory Tradeoff

- the probability of having a match for the right key becomes: $$1-\left(1-\frac{1}{n_1}\right)^{n_2} \approx 1-e^{-\frac{n_2}{n_1}}$$
  - which means a success probability of 63% for $n_1 = n_2$ and 98% for $n_2 = 4n_1$

A Time-Memory Tradeoff

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}

- the basic attack on AES-192 is not feasible
  - By tradeoff the attack becomes feasible for $n_1 > 2^{16} (n > 16)$
- The pre-computation cost is considered separately

Extension to 8 Rounds

- To attack 8-round AES we follow exactly the same steps of the 7-round attack, but we also search the last round key exhaustively
  - The data, pre-computation, 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 becomes 2^{200} (instead of 2^{72})
  - Faster than exhaustive search

An Improved Attack

- In the partial decryption phase of the attack in Step 3 where the attacker checks the partial cipher-text values of round 5
- if the attacker looks at the XOR of two partial cipher-texts rather than looking at them individually, the $k^{(5)}$ term in the equation cancels
- f denoting the mapping $a_{11} \rightarrow c_{11}^{(4)}$ the attacker computes and stores $S(\mathrm{f(i)}) + S(\mathrm{f(0)})$

- The key search complexity reduced by a factor of 2^8

Conclusion

- The attacks present a new way of utilizing square-like properties for attacking AES
  - if only one entry of a set of plaintexts is active each entry of the cipher-text after 4 rounds can be entirely defined using 25 fixed bytes
    - the first 5-round distinguisher of AES enabled to develop attacks on 7 and 8 rounds of AES-256 and 7 rounds of AES-192
- The attack has a huge pre-computation and memory complexity