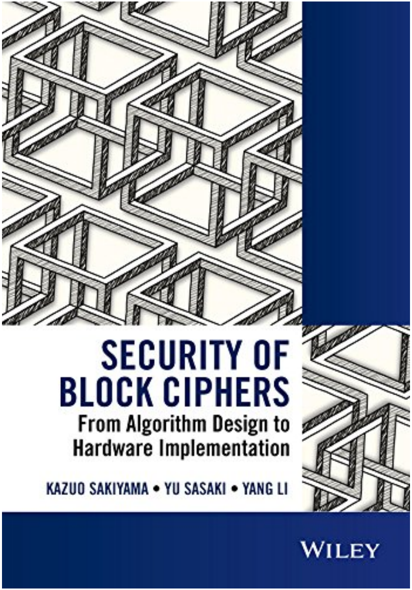


CS 553
CRYPTOGRAPHY

Lecture 14
More on Analyzing AES

Instructor
Dr. Dhiman Saha

The Square Attack

Integral Cryptanalysis of `AES`

Basic Set of Integral Cryptanalysis

$$
\begin{aligned}
P_0 &= (0, c_1, c_2, c_3,\ c_4, c_5, c_6, c_7,\ c_8, c_9, c_{10}, c_{11},\ c_{12}, c_{13}, c_{14}, c_{15}),\\
P_1 &= (1, c_1, c_2, c_3,\ c_4, c_5, c_6, c_7,\ c_8, c_9, c_{10}, c_{11},\ c_{12}, c_{13}, c_{14}, c_{15}),\\
P_2 &= (2, c_1, c_2, c_3,\ c_4, c_5, c_6, c_7,\ c_8, c_9, c_{10}, c_{11},\ c_{12}, c_{13}, c_{14}, c_{15}),\\
&\vdots\\
P_{255} &= (255, c_1, c_2, c_3,\ c_4, c_5, c_6, c_7,\ c_8, c_9, c_{10}, c_{11},\ c_{12}, c_{13}, c_{14}, c_{15}),
\end{aligned}
$$

$\mathcal{P} = \{P_0, P_1, P_2, \ldots, P_{255}\}$

P_i

$0 \le i \le 255$

i	c_4	c_8	c_{12}
c_1	c_5	c_9	c_{13}
c_2	c_6	c_{10}	c_{14}
c_3	c_7	c_{11}	c_{15}

- Unordered Set of 256 Plaintexts
- One byte takes all values in $\{0,1\}^8$, others are fixed
- c_i is constant
- $c_1, c_2, \cdots, c_{15} \in \{0,1\}^8$

Generally denoted by $\mathcal{A}$ — All

The byte in which all values appear exactly once among all the texts in the set is called the **all** property.

Generally denoted by $\mathcal{C}$ — Constant

The byte in which all texts in the set have an identical value is called the **constant** property.

$\mathcal{P} = \{P_0, P_1, P_2, \dots, P_{255}\}$

P_i

$0 \leq i \leq 255$

i	c_4	c_8	c_{12}
c_1	c_5	c_9	c_{13}
c_2	c_6	c_{10}	c_{14}
c_3	c_7	c_{11}	c_{15}

- The set $\mathcal{P}$ in terms of $\mathcal{A}$ and $\mathcal{C}$

$$\mathcal{P} = \{\mathcal{A}, \mathcal{C}, \mathcal{C}, \mathcal{C};\ \mathcal{C}, \mathcal{C}, \mathcal{C}, \mathcal{C};\ \mathcal{C}, \mathcal{C}, \mathcal{C}, \mathcal{C};\ \mathcal{C}, \mathcal{C}, \mathcal{C}, \mathcal{C}\}$$

- Basic idea: Study properties of $\mathcal{P}$ through `AES`

Processing $\mathcal{P}$ through Subkey XOR

$$\mathcal{P}^{\text{AK}} = \{P_0 \oplus sk_0, P_1 \oplus sk_0, P_2 \oplus sk_0, \ldots, P_{255} \oplus sk_0\}$$

$0 \leq i \leq 255$

$i \oplus sk_0[0]$	$c_4 \oplus sk_0[4]$	$c_8 \oplus sk_0[8]$	$c_{12} \oplus sk_0[12]$
$c_1 \oplus sk_0[1]$	$c_5 \oplus sk_0[5]$	$c_9 \oplus sk_0[9]$	$c_{13} \oplus sk_0[13]$
$c_2 \oplus sk_0[2]$	$c_6 \oplus sk_0[6]$	$c_{10} \oplus sk_0[10]$	$c_{14} \oplus sk_0[14]$
$c_3 \oplus sk_0[3]$	$c_7 \oplus sk_0[7]$	$c_{11} \oplus sk_0[11]$	$c_{15} \oplus sk_0[15]$

$\Rightarrow$

$\mathcal{A}$	$\mathcal{C}$	$\mathcal{C}$	$\mathcal{C}$
$\mathcal{C}$	$\mathcal{C}$	$\mathcal{C}$	$\mathcal{C}$
$\mathcal{C}$	$\mathcal{C}$	$\mathcal{C}$	$\mathcal{C}$
$\mathcal{C}$	$\mathcal{C}$	$\mathcal{C}$	$\mathcal{C}$

Lemma

By XORing an (un)known constant to each of the texts in the set,

- *the byte with all property* **still satisfies** *the all property, and*
- *the byte with constant property* **still satisfies** *the constant property.*

Processing $\mathcal{P}$ through SubBytes Operation

$$\mathcal{P}^{\text{SB}} = \{\text{SB}(P_0), \text{SB}(P_1), \text{SB}(P_2), \dots, \text{SB}(P_{255})\}$$

$0 \le i \le 255$

$S(i)$	$S(c_4)$	$S(c_8)$	$S(c_{12})$
$S(c_1)$	$S(c_5)$	$S(c_9)$	$S(c_{13})$
$S(c_2)$	$S(c_6)$	$S(c_{10})$	$S(c_{14})$
$S(c_3)$	$S(c_7)$	$S(c_{11})$	$S(c_{15})$

$\Rightarrow$

$\mathcal{A}$	$\mathcal{C}$	$\mathcal{C}$	$\mathcal{C}$
$\mathcal{C}$	$\mathcal{C}$	$\mathcal{C}$	$\mathcal{C}$
$\mathcal{C}$	$\mathcal{C}$	$\mathcal{C}$	$\mathcal{C}$
$\mathcal{C}$	$\mathcal{C}$	$\mathcal{C}$	$\mathcal{C}$

Lemma (Recall, S-box $\rightarrow$ bijective/fixed)

By applying the S-box for each of the texts in the set,

- *the byte with all property* **still satisfies** *the all property,*
- *the byte with constant property* **still satisfies** *the constant property.*

Processing $\mathcal{P}$ through ShiftRows Operation

Recall

ShiftRows only affects the byte positions.

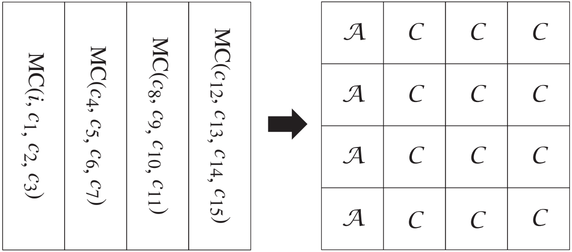
- No effect on value of a byte
- Note: Integral analysis only exploits the property inside a byte

Verdict

ShiftRows operation does not violate the properties used in the integral cryptanalysis

$\mathcal{P}^{\mathrm{MC}} = \{\mathrm{MC}(P_0), \mathrm{MC}(P_1), \mathrm{MC}(P_2), \ldots, \mathrm{MC}(P_{255})\}$

$0 \leq i \leq 255$



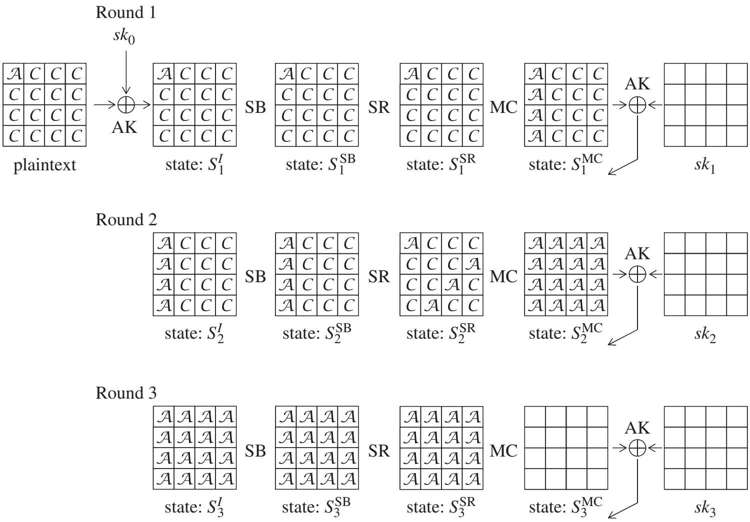
Processing $\mathcal{P}$ through MixColumns Operation

$$\begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix} \times \begin{bmatrix} i \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} 2i & \oplus & 3c_1 & \oplus & c_2 & \oplus & c_3 \\ i & \oplus & 2c_1 & \oplus & 3c_2 & \oplus & c_3 \\ i & \oplus & c_1 & \oplus & 2c_2 & \oplus & 3c_3 \\ 3i & \oplus & c_1 & \oplus & c_2 & \oplus & 2c_3 \end{bmatrix}$$

$$= \begin{bmatrix} 2i \\ i \\ i \\ 3i \end{bmatrix} \oplus \begin{bmatrix} 3c_1 & \oplus & c_2 & \oplus & c_3 \\ 2c_1 & \oplus & 3c_2 & \oplus & c_3 \\ c_1 & \oplus & 2c_2 & \oplus & 3c_3 \\ c_1 & \oplus & c_2 & \oplus & 2c_3 \end{bmatrix}$$

- XORing the constant does not change the **all** property and **constant** property.
- Dependence only on i which has all property.
- So, $i, 2i$, and $3i$ vary to take all the 256 values,
- Note: the order of the values changes.

Integral property for 2.5-round AES



Does any property remain after MixColumns of Round 3?

The Balanced Property $\quad$ All $\Longrightarrow$ Balanced $\not\Longrightarrow$ All

Idea

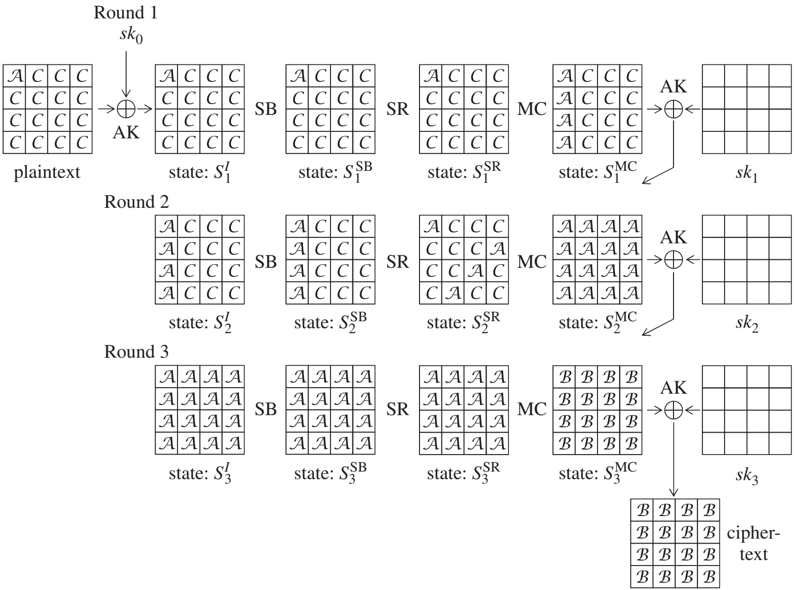
Compute XOR sum of all the 256 texts i.e., $\bigoplus_{i=0}^{255} S_{3,i}^{MC}[0]$

$$\begin{aligned}
\bigoplus_{i=0}^{255} S_{3,i}^{\mathrm{MC}}[0] &= \bigoplus_{i=0}^{255} (2 \cdot S_{3,i}^{\mathrm{SR}}[0] \oplus 3 \cdot S_{3,i}^{\mathrm{SR}}[1] \oplus S_{3,i}^{\mathrm{SR}}[2] \oplus S_{3,i}^{\mathrm{SR}}[3]) \\
&= \bigoplus_{i=0}^{255} (2 \cdot S_{3,i}^{\mathrm{SR}}[0]) \oplus \bigoplus_{i=0}^{255} (3 \cdot S_{3,i}^{\mathrm{SR}}[1]) \oplus \bigoplus_{i=0}^{255} S_{3,i}^{\mathrm{SR}}[2] \oplus \bigoplus_{i=0}^{255} S_{3,i}^{\mathrm{SR}}[3] \\
&= 0 \oplus 0 \oplus 0 \oplus 0 = 0.
\end{aligned}$$

True for all bytes in S_3^{MC} $\quad$ **XOR Sum is Zero**

Denoted by $\mathcal{B}$: $\quad \forall j \quad \bigoplus_{i=0}^{255} S_{3,i}^{MC}[j] = 0, \quad 0 \leq j \leq 15$

Integral property for three-round AES



Integral Distinguisher

- Verify XOR sum of 256 states = Zero
- Hold with probability 1 for `AES` 3 rounds

What about random permutation?

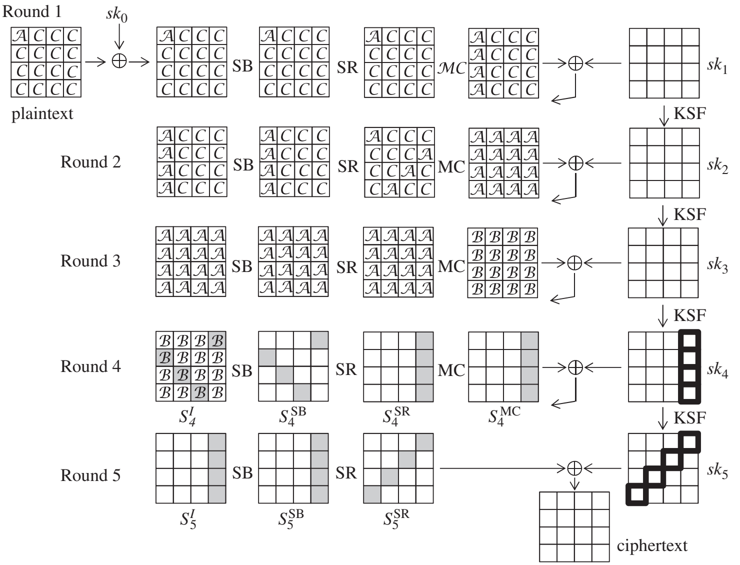
- XOR sum of 256 randomly generated bytes is 0 with probability 2^{-8}
- For all 16 bytes this holds with $2^{-8\cdot 16} = 2^{-128}$ i.e., negligible

- Distinguishing Complexity

$$(Data, Time, Memory) = (256, 256, negl)$$

Key Recovery Attack with Integral Cryptanalysis for Five Rounds

Guess 8-bytes ($4 \rightarrow sk_5, sk_4$) 5-Round Key Recovery



Verify: $\bigoplus S_4^I[1] = \bigoplus S_4^I[6] = \bigoplus S_4^I[11] = \bigoplus S_4^I[12] = 0$

$$\bigoplus S_4^I[1] = \bigoplus S_4^I[6] = \bigoplus S_4^I[11] = \bigoplus S_4^I[12] = 0 \qquad (1)$$

- Correct guess satisfies (1) deterministically
- Wrong guesses satisfy probabilistically
- The probability that randomly chosen 4 byte values become 0:

$$2^{(-8)4} = 2^{-32}$$

- With 2^{64} guesses, expected number of subkeys passing (1):

$$2^{64} \cdot 2^{-32} = 2^{32}$$

- With one set subkey space reduces by 32 bits ($2^{64} \rightarrow 2^{32}$)
- For next set, reduces list is used, reduction by another 32 bits.
- Expected number of subkeys passing is ≈ 1

Subkey Space Reduction

$$\bigoplus S_4^I[1] = \bigoplus S_4^I[6] = \bigoplus S_4^I[11] = \bigoplus S_4^I[12] = 0 \qquad (1)$$

- Correct guess satisfies (1) deterministically
- Wrong guesses satisfy probabilistically
- The probability that randomly chosen 4 byte values become 0:

$$2^{(-8)4} = 2^{-32}$$

- With 2^{64} guesses, expected number of subkeys passing (1):

$$2^{64} \cdot 2^{-32} = 2^{32}$$

- With one set subkey space reduces by 32 bits ($2^{64} \rightarrow 2^{32}$)
- For next set, reduces list is used, reduction by another 32 bits.
- Expected number of subkeys passing is ≈ 1

Subkey Space Reduction

$$\bigoplus S_4^I[1] = \bigoplus S_4^I[6] = \bigoplus S_4^I[11] = \bigoplus S_4^I[12] = 0 \qquad (1)$$

- Correct guess satisfies (1) deterministically
- Wrong guesses satisfy probabilistically
- The probability that randomly chosen 4 byte values become 0:

$$2^{(-8)4} = 2^{-32}$$

- With 2^{64} guesses, expected number of subkeys passing (1):

$$2^{64} \cdot 2^{-32} = 2^{32}$$

- With one set subkey space reduces by 32 bits ($2^{64} \rightarrow 2^{32}$)
- For next set, reduces list is used, reduction by another 32 bits.
- Expected number of subkeys passing is ≈ 1

Data Complexity

- The attacker prepares sets of 256 plaintexts $\mathcal{P}$.
- Guesses 64 bits of subkeys
- Each set of 256 plaintexts $\mathcal{P}$ can reduce the subkey space by a factor of 2^{32}
- In order to reduce the subkey space to 1, two sets of 256 plaintexts $\mathcal{P}$ are required.
- $2 \cdot 256 = 512$ plaintexts are passed to the encryption oracle
- The attacker obtains the corresponding **two** sets of 256 ciphertexts

Data Complexity $= 2^9$ Chosen Plaintexts

Time Complexity

- For first set, the **two-round** decryption is performed for each of the 2^{64} subkey guesses and 2^8 ciphertexts in the set
- Computational cost for first set is

$$2 \cdot 2^{64+8} = 2^{73} \text{ round function computations}$$

- Equivalent to

$$2^{73}/5 = 2^{70.7} \text{ five-round AES computations}$$

- Effort for second set cheaper by a factor of 2^{32} (ignored)
- This is repeated twice for remaining two columns
- Followed by exhaustive search for last column
- Effort for exhaustive search is again cheaper (ignored)
- Time complexity is

$$3 \cdot 2^{70.7} \approx 2^{72.3} \text{ 5-round AES computations}$$

- For first set, the **two-round** decryption is performed for each of the 2^{64} subkey guesses and 2^8 ciphertexts in the set
- Computational cost for first set is

$$2 \cdot 2^{64+8} = 2^{73} \text{ round function computations}$$

- Equivalent to

$$2^{73}/5 = 2^{70.7} \text{ five-round AES computations}$$

- Effort for second set cheaper by a factor of 2^{32} (ignored)
- This is repeated twice for remaining two columns
- Followed by exhaustive search for last column
- Effort for exhaustive search is again cheaper (ignored)
- Time complexity is

$$3 \cdot 2^{70.7} \approx 2^{72.3} \text{ 5-round AES computations}$$

Time Complexity

- For first set, the **two-round** decryption is performed for each of the 2^{64} subkey guesses and 2^8 ciphertexts in the set
- Computational cost for first set is

$$2 \cdot 2^{64+8} = 2^{73} \text{ round function computations}$$

- Equivalent to

$$2^{73}/5 = 2^{70.7} \text{ five-round AES computations}$$

- Effort for second set cheaper by a factor of 2^{32} (ignored)
- This is repeated twice for remaining two columns
- Followed by exhaustive search for last column
- Effort for exhaustive search is again cheaper (ignored)
- Time complexity is

$$3 \cdot 2^{70.7} \approx 2^{72.3} \text{ 5-round AES computations}$$

Time Complexity

- For first set, the **two-round** decryption is performed for each of the 2^{64} subkey guesses and 2^8 ciphertexts in the set
- Computational cost for first set is

$$2 \cdot 2^{64+8} = 2^{73} \text{ round function computations}$$

- Equivalent to

$$2^{73}/5 = 2^{70.7} \text{ five-round AES computations}$$

- Effort for second set cheaper by a factor of 2^{32} (ignored)
- This is repeated twice for remaining two columns
- Followed by exhaustive search for last column
- Effort for exhaustive search is again cheaper (ignored)
- Time complexity is

$$3 \cdot 2^{70.7} \approx 2^{72.3} \text{ 5-round AES computations}$$

Memory Complexity

- Need to store reduced subkey list from first set
- To use as base list for second setv
- Memory required reduced subkey space

$$2^{32} \text{ 8-byte information}$$

- Equivalent to

$$2^{31} \text{ AES states}$$

- Memory requirement for other part is negligible

Memory Complexity 2^{31} AES states

The complexity of this attack is

$$(Data, Time, Memory) = (2^9, 2^{72.3}, 2^{31})$$