Yoyo Game with AES

Navid Ghaedi Bardeh

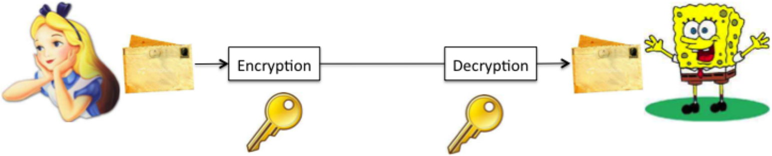
University of Bergen

May 8, 2018

Outline

Classical Model of Symmetric Cryptography

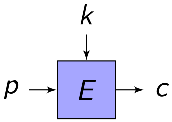
Alice and Bob exchange the secret key through a secure channel.



Block Cipher

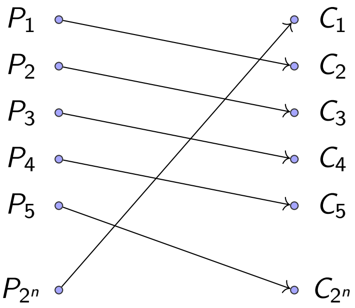
A block of plaintext p encrypt to a block of ciphertext c under the action of the key k:

$$\begin{aligned} E : \{0,1\}^n \times \{0,1\}^\kappa &\rightarrow \{0,1\}^n \\ (p,k) &\rightarrow E(p,k) = c \end{aligned}$$

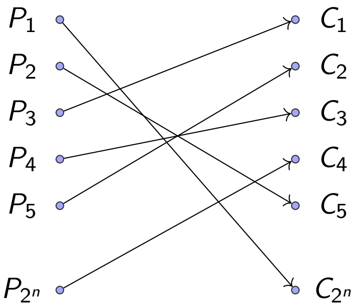


Block Cipher(cont.)

Each key induces a permutation between the plaintexts and the ciphertexts



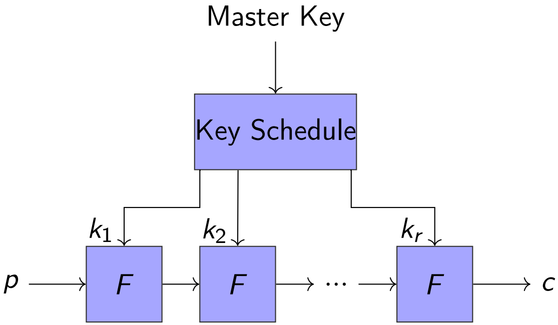
Under key K_1



Under key K_2

Iterated Block Cipher

Iterate a round function f several times:

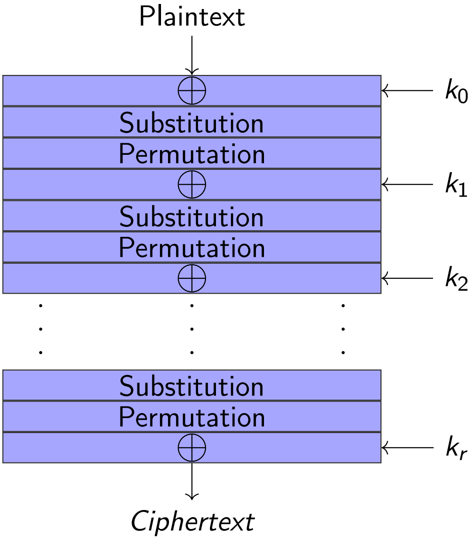


How to build the round function?

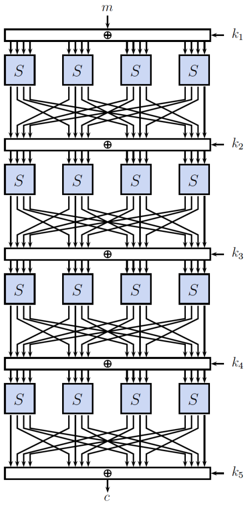
Two typical approaches:

- Feistel Network
- Substitution Permutation Network (SPN)

Substitution Permutation Network (SPN)



Substitution Permutation Network (SPN)



Cryptanalysis of block ciphers

In symmetric key cryptography, security proofs are partial and insufficient

- An algorithm is secure as long there is no attack against it
- Make it secure against all known attacks.
- The more an algorithm is analysed without being broken, the more reliable it is.

What is a broken cipher?

- If a block cipher encrypts messages with a k-bit key, no attack with time complexity less than 2^k should be known
- Otherwise, the cipher is considered as broken (even if the complexity of the attack is not practical).

Distinguisher Attack

- of the weakest cryptographic attack.
- one simulates the block cipher for which the cryptography key has been chosen at random;
- the other simulates a truly random permutation.

Goal: distinguish the two oracles, i.e. decide which oracle is the cipher.

Introduction

- The Yoyo game was introduced by Biham et al. against Skipjack (Feistel block cipher)
- Yoyo Game: Suppose a plaintext pair has (or has not) a specific property. It is possible to generate other plaintext pairs that has (or has not) the same property by exchanging a specific word of their ciphertexts and decrypt new ciphertext pair.
- Open problem: How to do this for SPN ciphers and in particular for AES

Generic SPN block cipher

- Let $\alpha = (\alpha_0, \alpha_1, \ldots, \alpha_{n-1}) \in \mathbb{F}_q^n$ denote the state of a block cipher.
- Let $q = 2^k$ and let $s(x)$ be a *kxk* permutation s-box.
- The S-box working on a state is defined by

$$S(\alpha) = (s(\alpha_0), s(\alpha_1), \ldots, s(\alpha_{n-1}))$$

- Let L be a linear layer in the block cipher
- We consider SPNs of the form:
  - two rounds: $S \circ L \circ S$

The yoyo operation

Definition

For a vector $c \in \mathbb{F}_2^n$ and a pair of states $\alpha, \beta \in \mathbb{F}_q^n$ define a new state $\rho^c(\alpha, \beta)$ by

$$\rho^c(\alpha, \beta)_i = \begin{cases} \alpha_i & \text{if } c_i = 1, \\ \beta_i & \text{if } c_i = 0. \end{cases}$$

Example

Let $c = (0110)$ and $\alpha = (\alpha_0, \alpha_1, \alpha_2, \alpha_3)$ and $\beta = (\beta_0, \beta_1, \beta_2, \beta_3)$. Then

$$\alpha' = \rho^{(0110)}(\alpha, \beta) = (\beta_0, \alpha_1, \alpha_2, \beta_3)$$

and

$$\beta' = \rho^{(0110)}(\beta, \alpha) = (\alpha_0, \beta_1, \beta_2, \alpha_3)$$

Call $(\alpha', \beta') = (\rho^c(\alpha, \beta), \rho^c(\beta, \alpha))$ a yoyo pair.

Properties of the yoyo operation

Lemma

Let $\alpha' = \rho^c(\alpha, \beta)$ *and* $\beta' = \rho^c(\beta, \alpha)$.

a) $\alpha' \oplus \beta' = \alpha \oplus \beta$
b) $S(\alpha') \oplus S(\beta') = S(\alpha) \oplus S(\beta)$
c) $L(S(\alpha')) \oplus L(S(\beta')) = L(S(\alpha)) \oplus L(S(\beta))$

Proof.

a)

$$\rho^c(\alpha, \beta)_i \oplus \rho^c(\beta, \alpha)_i = \begin{cases} \alpha_i \oplus \beta_i & \text{if } c_i = 1, \\ \beta_i \oplus \alpha_i & \text{if } c_i = 0 \end{cases}$$

b)

$$s(\rho^c(\alpha, \beta)_i) \oplus s(\rho^c(\beta, \alpha)_i) = \begin{cases} s(\alpha_i) \oplus s(\beta_i) & \text{if } c_i = 1, \\ s(\beta_i) \oplus s(\alpha_i) & \text{if } c_i = 0 \end{cases}$$

c) the result follows from the linearity of L.

□

The zero difference pattern

Definition (Zero difference pattern)

Let $\alpha = (\alpha_0, \alpha_1, \ldots, \alpha_{n-1}) \in \mathbb{F}_q^n$. Define

$$\nu(\alpha) = (z_0, z_1, \ldots, z_{n-1}) \in \mathbb{F}_2^n$$

where

$$z_i = \begin{cases} 1 & \text{if } \alpha_i \text{ is zero,} \\ 0 & \text{otherwise.} \end{cases}$$

Example

Let $\alpha = (\alpha_0, \alpha_1, 0, \alpha_3)$. Then

$$\nu(\alpha) = (0, 0, 1, 0)$$

Lemma

Let $\alpha' = \rho^c(\alpha, \beta)$ *and* $\beta' = \rho^c(\beta, \alpha)$.

a) $\nu(\alpha \oplus \beta) = \nu(S(\alpha) \oplus S(\beta))$

Typical use of yoyo operation

$$
\begin{array}{ccccccc}
p^0 & \oplus & p^1 & \overset{\nu}{=} & p^{0\prime} & \oplus & p^{1\prime} \\
\Downarrow & S & \Downarrow & & \Uparrow & S^{-1} & \Uparrow \\
S(p^0) & \oplus & S(p^1) & = & L^{-1}(S^{-1}(c^{0\prime})) & \oplus & L^{-1}(S^{-1}(c^{1\prime})) \\
\Downarrow & L & \Downarrow & & \Uparrow & L^{-1} & \Uparrow \\
L(S(p^0)) & \oplus & L(S(p^1)) & = & S^{-1}(c^{0\prime}) & \oplus & S^{-1}(c^{1\prime}) \\
\Downarrow & S & \Downarrow & & \Uparrow & S^{-1} & \Uparrow \\
c^0 & \oplus & c^1 & \overset{\rho^c}{\Rightarrow} & c^{0\prime} & \oplus & c^{1\prime}
\end{array}
$$

Adaptive

a) Pick two plaintexts p^0 and p^1 with a zero difference $\nu(p^0 \oplus p^1)$.

b) Encrypt p^0 and p^1 to c^0 and c^1.

c) Make two new ciphertexts $c^{0\prime} = \rho^c(c^0, c^1)$ and $c^{1\prime} = \rho^c(c^1, c^2)$.

d) Decrypt $c^{0\prime}$ and $c^{1\prime}$.

e) $\nu(p^0 \oplus p^1) = \nu(p^{0\prime} \oplus p^{1\prime})$

Advanced Encryption Standard (AES)

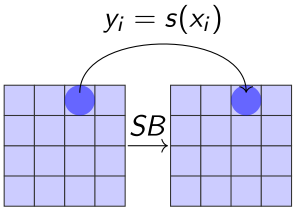
- Byte-oriented Substitution-Permutation Network.
- Block size of 128 bits, key size of 128, 192, 256 bits.
- Number of rounds depend on key size 10, 12, 14 rounds resp.
- 128 bits of block size, seen as a 4×4 matrix of bytes.

An round of AES

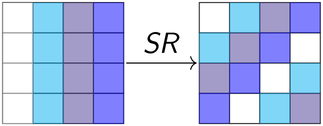
Each round is a composition of four byte-oriented transformations:

- SubBytes
- ShiftRows
- MixColumns
- AddRoundKey

SubBytes

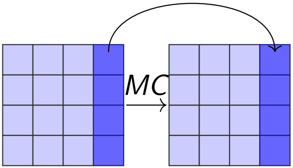


ShiftRows



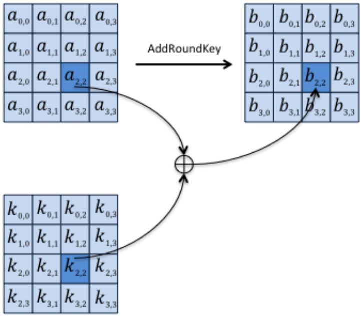
MixColumns

$$C \leftarrow M \times C$$



$$M = \begin{pmatrix} x & x+1 & 1 & 1 \\ 1 & x & x+1 & 1 \\ 1 & 1 & x & x+1 \\ x+1 & 1 & 1 & x \end{pmatrix}$$

AddRoundKey



Super-box representation of 2 rounds of AES

- $R^2 = AK \circ MC \circ SR \circ SB \circ AK \circ MC \circ SR \circ SB.$
- Rewrite the operations :
- $R^2 = AK \circ MC \circ SR \circ (SB \circ AK \circ MC \circ SB) \circ SR.$
- Then:
- $\textit{Super-box} = SB \circ AK \circ MC \circ SB$

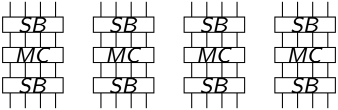


Figure: Super-box of AES

Four Rounds of AES

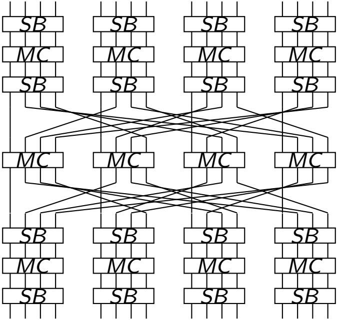


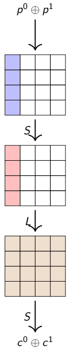
Figure: $S \circ L \circ S$ in AES

Four Round AES Yoyo Distinguisher

Theorem

Four rounds of AES can be distinguished from a random cipher using one pair of chosen plaintexts and one (adaptively) chosen ciphertext pair.

1. Select $p^0 \oplus p^1$ that differ in only one word
2. ask for encryption c^0 and c^1 of p^0 and p^1

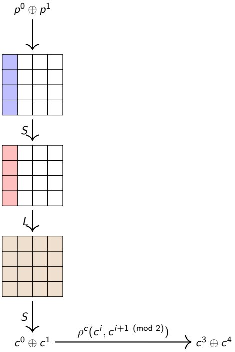


Four Round AES Yoyo Distinguisher

Theorem

Four rounds of AES can be distinguished from a random cipher using one pair of chosen plaintexts and one (adaptively) chosen ciphertext pair.

1. Select $p^0 \oplus p^1$ that differ in only one word
2. ask for encryption c^0 and c^1 of p^0 and p^1
3. construct $c^3 = \rho^c(c^0, c^1), c^4 = \rho^c(c^1, c^0)$

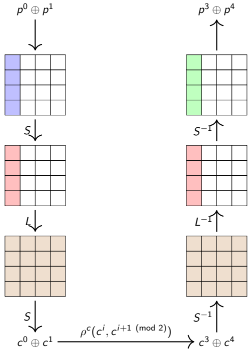


Four Round AES Yoyo Distinguisher

Theorem

Four rounds of AES can be distinguished from a random cipher using one pair of chosen plaintexts and one (adaptively) chosen ciphertext pair.

1. Select $p^0 \oplus p^1$ that differ in only one word
2. ask for encryption c^0 and c^1 of p^0 and p^1
3. construct $c^3 = \rho^c(c^0, c^1), c^4 = \rho^c(c^1, c^0)$
4. get plaintexts p^3, p^4.

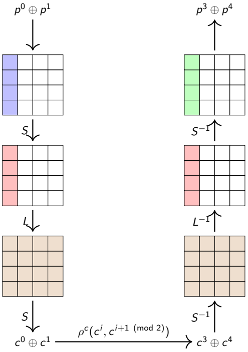


Four Round AES Yoyo Distinguisher

Theorem

Four rounds of AES can be distinguished from a random cipher using one pair of chosen plaintexts and one (adaptively) chosen ciphertext pair.

1. Select $p^0 \oplus p^1$ that differ in only one word
2. ask for encryption c^0 and c^1 of p^0 and p^1
3. construct $c^3 = \rho^c(c^0, c^1), c^4 = \rho^c(c^1, c^0)$
4. get plaintexts p^3, p^4.
5. if AES, then same zero difference pattern (prob for random $= 2^{-96}$)



Results

Table: *Secret-Key Distinguishers for AES*

Property	Rounds	Data	Cost
Trun. Diff.	3	$2^{4.3}$ CP	$2^{11.5}$ XOR
Integral	3	2^{8} CP	2^{8} XOR
Yoyo	3	3 ACC	1 XOR
Imp. Diff.	4	$2^{16.25}$ CP	$2^{22.3}$ M
Integral	4	2^{32} CP	2^{32} XOR
Yoyo	4	4 ACC	1 XOR
Struct. Diff.	5	2^{33}	$2^{36.6}$ M
Imp. Diff.	5	$2^{98.2}$ CP	2^{107} M
Integral	5	2^{128} CC	2^{128} XOR
Yoyo	5	$2^{25.8}$ ACC	$2^{24.8}$ XOR
Yoyo	6	$2^{122.83}$ ACC	$2^{121.83}$ XOR

Results

Table: *Comparison of key-recovery on 5 rounds of AES*

Attack	Rounds	Data	Computation	Memory
MitM	5	8 CP	2^{64}	2^{56}
Imp. Polyt.	5	15 CP	2^{70}	2^{41}
Integral	5	2^{11} CP	$2^{45.7}$	small
Imp. Diff.	5	$2^{31.5}$ CP	2^{33}	2^{38}
Boomerang	5	2^{39} ACC	2^{39}	2^{33}
Yoyo	5	$2^{11.3}$ ACC	2^{29}	small

Conclusion

- new records 3-6 round distinguishers AES
- new record 5 round key recovery
- can be applied directly to similar designs as well
- can be improved (more rounds) for lightweight designs
- results published at Asiacrypt 2017

Thanks for your attention!