

COMPUTER & NETWORK SECURITY

Lecture 6:

Attacks on DES

DES KEYS

Given one plaintext/cyphertext (\mathbf{m} , \mathbf{c}) pair, there is a high probability that only one key will satisfy

$$\mathbf{c} = \text{DES}(\mathbf{m}, \mathbf{k})$$

Consider DES as a collection of permutations: $\pi(1) \dots \pi(2^{56})$

If π_i are independent permutations then $\forall (\mathbf{m}, \mathbf{k}) \dots$

$$\Pr [\exists \mathbf{k}_1 \neq \mathbf{k} : \text{DES}(\mathbf{m}, \mathbf{k}_1) = \text{DES}(\mathbf{m}, \mathbf{k})]$$

$$= 256 \times 2^{-64}$$

$$= 2^{-56} = 1.39 \times 10^{-17} = 0.0000000000000000139\%$$

Thus given one (\mathbf{m} , \mathbf{c}) pair the key is (almost definitely) uniquely determined.
The problem is to find \mathbf{k} .

ATTACKS ON DES

Exhaustive key search

- For a strong n -bit block cypher with a j -bit key, the key can be recovered on average in 2^{j-1} operations given a small number ($< (j+4)/n$) of plaintext/cyphertext pairs
- For DES, $j = 56$ bits, $n = 64$ bits so exhaustive search is expected to yield the key in 2^{55} operations

Cyphertext-only DES key search

- Suppose DES is used to encrypt 8×8 ASCII characters (=64 bits) per block, with one bit being a parity bit
- Trial decryption yields all 8 parity bits valid with probability 2^{-8} and thus for t blocks 2^{-8t}
- Over 2^{56} keys, this leads to probability of correct key when parity bits are valid being $(1 - 2^{-8t})$
- thus only $t \sim 5$ -10 blocks are enough $> 99.9999999\%$ sure.

■ REDUCING EFFORT IN ATTACKS ON DES

DES is a Feistel network, thus:

$$\text{DES}(\neg \mathbf{m}, \neg \mathbf{k}) = \neg \text{DES}(\mathbf{m}, \mathbf{k})$$

This is called the “**Complementation Property**”

So for a CPA:

if $\mathbf{c}_1 = \text{DES}(\mathbf{m}, \mathbf{k})$ and $\mathbf{c}_2 = \text{DES}(\neg \mathbf{m}, \mathbf{k})$ then if
 $\text{DES}(\mathbf{m}, \mathbf{k}_1) \neq \mathbf{c}_1$ nor $\neg \mathbf{c}_2$ then $\mathbf{k} \neq \mathbf{k}_1$ nor $\neg \mathbf{k}_1$

Search space is reduced by half

2DES

Double encryption with DES (2DES) is bad

$$2DES_{k1, k2}(\mathbf{m}) = E_{k1}(E_{k2}(\mathbf{m}))$$

2DES is vulnerable to “**meet in the middle**” attack with known plaintext.

– i.e. for a fixed message, \mathbf{m} , create a table:

$$E_k(\mathbf{m}) \text{ for all } k \in \{0,1\}^{56}$$

Then for $\mathbf{c} = E_{k1, k2}(\mathbf{m})$ try:

$$D_k(\mathbf{c}) \text{ for all } k \in \{0,1\}^{56}$$

Until $D_k(\mathbf{c})$ appears in the table, since $D_{k1}(\mathbf{c}) = E_{k2}(\mathbf{m})$.

Thus 2DES can be broken on average in 2^{56} operations using 2^{56} memory slots (a time-space trade-off)

Not good for 112 bits of key (56 + 56)!

3DES

Two-key Triple DES (**3DES**) is DES three times, two keys (112 bits)

$$3DES_{k_1, k_2}(m) = E_{k_1}(D_{k_2}(E_{k_1}(m)))$$

The strength of DES (& 3DES) is that it does not form a group

$$DES_{k_1}(DES_{k_2}(m)) \neq DES_{k_3}(m)$$

Consider the time-space tradeoff on 2TDES:

– for time $2^{(56+64)}/s$ and space s , we can recover **k1** and **k2** in 2TDES

If $s > 28$ then we can do better than exhaustive search

If you have three distinct keys, it has 168 key bits
(effective key length of 112 bits due to “meet in the middle” attack)

If you use two keys ($k_1 = k_3, k_2$) then it has 112 key bits
(effective key length of 80 bits due to chosen-plaintext / known-plaintext attacks)

DESX

A modification of DES to avoid exhaustive key search is DESX

$k_1 = 56$ bits (DES key)

$k_2 = 64$ bits (whitening key)

$k_3 = h(k_2, k_3) = 64$ bits

$$\text{DESX}_{k_1, k_2, k_3}(m) = k_3 \oplus E_{k_1}(m \oplus k_2)$$

Whitening key gives greater resilience to brute force attacks

Given j plaintext / cyphertext pairs, the effective key size is greater than or equal to

$$\begin{aligned} |k| + n - 1 - \log j &= 56 + 64 - 1 - \log j \\ &= 119 - \log j \\ &\geq 100 \text{ bits} \end{aligned}$$

DIFFERENTIAL CRYPTANALYSIS

This method is a better-than-brute-force approach to attacking DES with (plaintext, cyphertext) pairs (CPA)

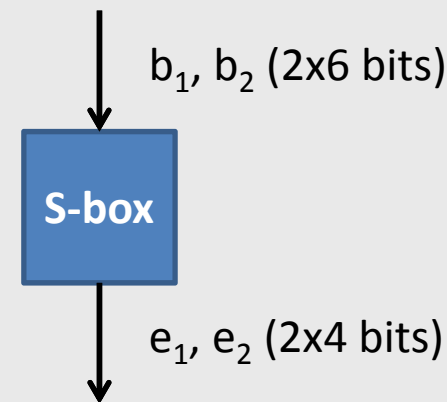
Involves looking at the XOR of two texts

We consider any s-box function $F(\mathbf{x}, \mathbf{k}_i)$:

Define the difference measure (on input) as

$$\begin{aligned}\Delta &= b_1 \oplus b_2 \\ &= (x_1 \oplus k_i) \oplus (x_2 \oplus k_i) \\ &= x_1 \oplus x_2\end{aligned}$$

The input XOR ($b_1 \oplus b_2$) does not depend on the key but the output XOR ($e_1 \oplus e_2$) does

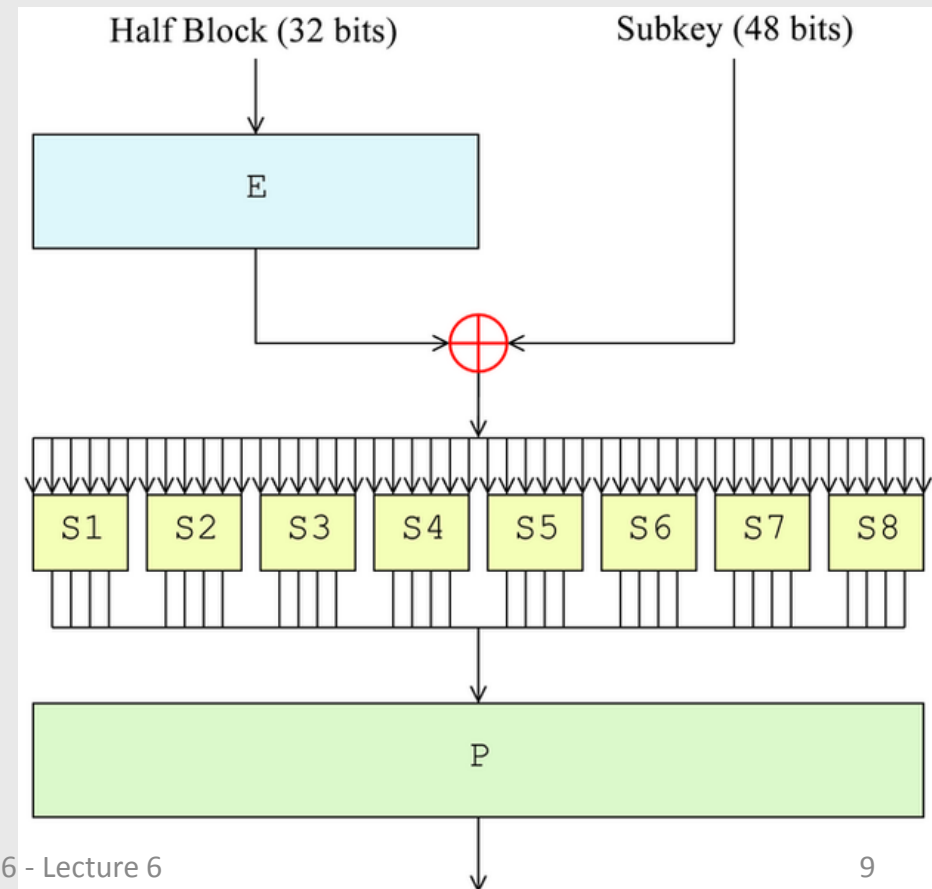


DES ROUND FUNCTION

The function $F(x, k_i): \{0,1\}^{32} \times \{0,1\}^{48} \rightarrow \{0,1\}^{32}$

Half block is reversibly expanded to 48 bits in the Expander (E) function

S-Box collapses groups of 6 bits into groups of 4 bits (i.e. convert 48 bits back to 32 bits)



DIFFERENTIAL CRYPTANALYSIS

Now define the set $\Delta(b)$ consisting of ordered pairs (b_1, b_2) having input XOR b :

$$\Delta(b) = \{(b_1, b_2) \in \{0,1\}^6 \mid b_1 \oplus b_2 = b\}$$

where

$$|\Delta(b)| = 2^6 = 64$$

Take the example where $b = 110100$, then if we consider the first S-box the pairs might be:

$$\Delta(b) = \{(000000, 110100), \quad (000001, 110101), \quad \dots \quad (111111, 001011)\}$$

\oplus
110100

\oplus
110100

\oplus
110100

DIFFERENTIAL CRYPTANALYSIS

If this is done for all 64 pairs in $\Delta(b)$ then the following distribution of output XORs ($e_1 \oplus e_2$) is obtained:

$(e_1 \oplus e_2)$	0000	0001	0010	0011 ...	1111
	0	8	16	6	6

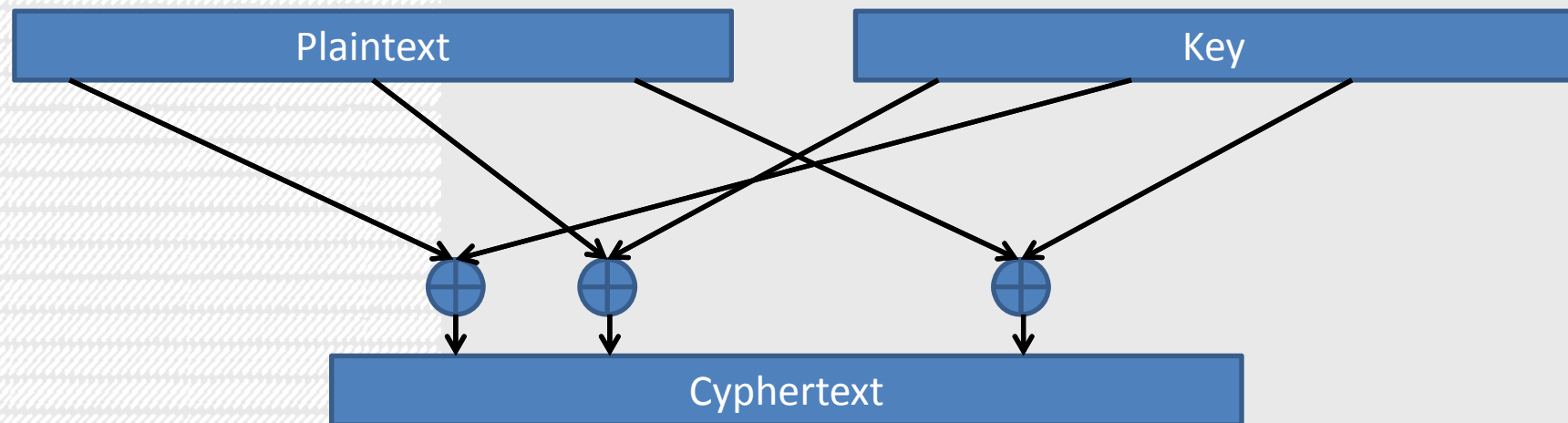
Now suppose that $(b_1 \oplus b_2) = 110100$ and $(e_1 \oplus e_2) = 0001$, then (b_1, b_2) must be one of eight possible pairs, hence b_1 is one of 16 possible values.

Since x_1 is known (this is a KPA), the 6 bits of the key XORed with x_1 to give b_1 are one of 16 possible values.

This procedure is repeated for different Δ s to make deductions about the key bits and eventually recover the key

LINEAR CRYPTANALYSIS

Consider the cyphertext derived by combining certain bits from the plaintext and key:



The cypher can easily be broken, for example, if

$$c[1] = p[4] \oplus p[17] \oplus k[5] \oplus k[3]$$

i.e. $k[3] \oplus k[5] = c[1] \oplus p[4] \oplus p[17]$

This is because the cypher is linear

■ LINEAR CRYPTANALYSIS

If we use the following notation:

$$p[i_1, \dots, i_u] = p[i_1] \oplus p[i_2] \oplus \dots \oplus p[i_u]$$

(the xor bits of the plaintext)

Also define

$$\rho = \Pr [p[i_1, \dots, i_u] \oplus c[j_1, \dots, j_v] = k [s_1, \dots, s_w]]$$

Now if $|\rho - 0.5|$ is large, then we can guess $k [s_1, \dots, s_w]$

Optimally for a break $|\rho - 0.5| = 0.5$ ($\rho = 0$ or 1)
(perfect cipher would have $\rho = 0.5$)

ALGORITHM TO RECOVER KEY BITS

Given R plaintext, cyphertext pairs (R is large):

if $\rho > 0.5$ then

$k[s_1, \dots, s_w] = \text{majority}\{p[i_1, \dots, i_u]\} \oplus c[j_1, \dots, j_v]$
over all plaintext cyphertext pairs

if $\rho < 0.5$ then

$k[s_1, \dots, s_w] = \text{minority} = 1 \oplus \text{majority}$

(*) Fact: if given $R \geq (\rho - 0.5)^{-2}$ then the correct value of $k[s_1, \dots, s_w]$ is obtained with probability $> 97.7\%$

LINEAR CRYPTANALYSIS OF DES

In 1993, Matsui made the following observation about DES which approximates the 5th S-box as a linear function:

$$\rho_5 = \Pr[x[4] = S5(x)[0,1,2,3]] = 12/64 = 0.19$$

(Note $x \in \{0,1\}^6$)

For the i th DES round

$$\Pr[r_i[15] \oplus F(r_i, k_i)[7,18,24,29] = k_i[22]] = \rho_5 = 0.19$$

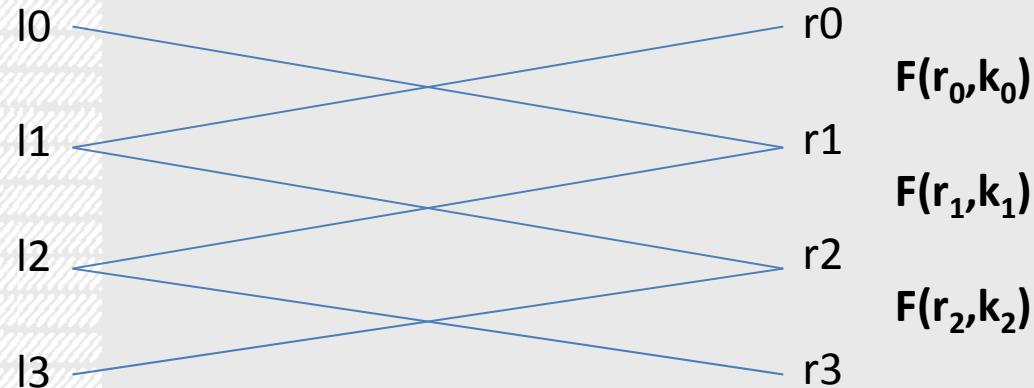
(r_i is the i th round right hand part)

where the bits have been chosen to undo the permutation

ATTACK ON 3DES

Left half of plaintext

Right half of plaintext



Left half of plaintext

Right half of plaintext

From the first round we can write

$$\Pr[r_1[7,18,24,29] \text{ xor } l_0[7,18,24,29] \text{ xor } r_0[15] = k_0[22]] = \rho_5$$

From the last round we can write

$$\Pr[r_1[7,18,24,29] \text{ xor } c_r[7,18,24,29] \text{ xor } c_l[15] = k_2[22]] = \rho_5$$

XORing together gives

$$\begin{aligned} \Pr[l_0[7,18,24,29] \text{ xor } c_r[7,18,24,29] \text{ xor } c_l[15] \text{ xor } r_0[15] &= k_0[22] \text{ xor } k_2[22]] \\ &= \rho_5 \cdot \rho_5 + (1 - \rho_5)^2 = 0.7 \end{aligned}$$

Using the fact (*) we can find $k_0[22] \text{ xor } k_2[22]$ using $R = (0.7 - 0.5)^{-2}$

= 25 plaintext/cyphertext pairs

DES STRENGTH AGAINST ATTACKS

Attack	Complexity			
	# Messages		Requirements	
	Known	Chosen	Storage	Processing
Exhaustive Precomputation	-	1	2^{56}	1 (lookup)
Exhaustive Search	1	-	Neg.	2^{55}
Linear Cryptanalysis	2^{43} (85%)	-	For texts	2^{43}
	2^{38} (10%)	-	For texts	2^{50}
Differential Cryptanalysis	-	2^{47}	For texts	2^{47}
	2^{55}	-	For texts	2^{55}

REPLACING DES

US government wanted DES used for all security: “standards”

RSA Security wanted to demonstrate DES was weak due to short key length

(1997)

First DES Challenge solved in 96 days using distributed computing (idle CPU)

Second DES Challenge solved in 41 days by *distributed.net* (idle CPU)

(1998)

EFF created Deep Crack for \$250k – decrypted message after only 56 hours

(1999)

Deep Crack + Distributed.net decrypted DES cyphertext in 22hrs 15mins

EFF'S DES CRACKER

Whitfield Diffie and Martin Hellman (Stanford University) estimated that a machine fast enough to test that many keys in a day would have cost about \$20 million in 1976 (pretty minimal cost for the NSA or other govts)

Composed of 1856 custom ASIC DES chips

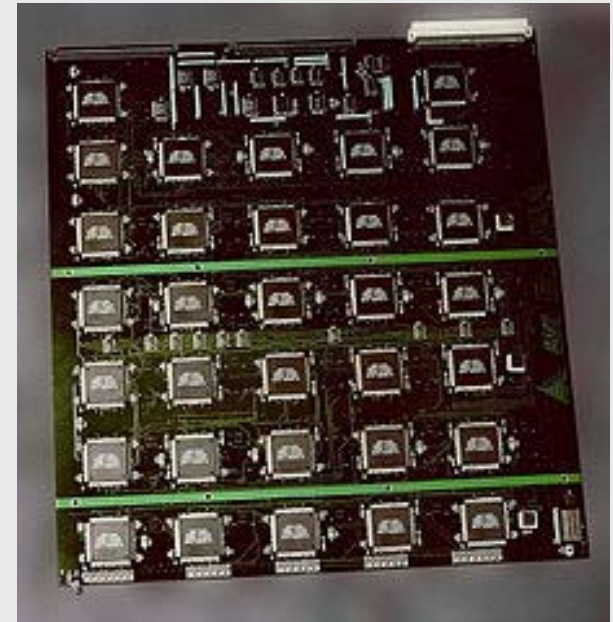
Machine could test 90 billion ($\sim 2^{36}$) keys per second

Entire key space in 9 days

(on average the key would be found in half that)

2006: COPACOBANA costs \$10k and will recover a DES key in ~ 6.4 days

2008: Reduced to less than one day using 128 off the shelf FPGAs



■ ADVANCED ENCRYPTION STANDARD (AES)

NIST Requirements

- Block cypher
- 128 bit blocks
- 128/192/256 bit keys
- Strength equal to or better than 3DES at greatly improved efficiency
 - Smartcard, hardware, software
 - Flexibility
 - Simplicity and elegance
- Royalty-free worldwide
- Security for over 30 years
- May protect sensitive data for 100 years
- Public confidence in the cypher

AES CANDIDATES

15 submissions from international field

Numerous strong candidates

Name	Type	Rounds	Rel. Speed (cycles)	Gates
Twofish	Feistel	16	1254	23k
Serpent	SPnetwork	32	1800	70k
Mars	Ext. Feistel	32	1600	70k
Rijndael	Square	10, 12, 14	1276	-
RC6	Feistel	20	1436	-

AES

Rijndael (pronounced “rain-dahl”) announced Oct 2000

Operates on 128 bit blocks

Key length is variable: 128, 192 or 256 bits

An SP-network

Uses a single S-box which acts on a byte input to give a byte output (think of as a 256 byte lookup table):

$$S(\mathbf{x}) = \mathbf{M}(1/\mathbf{x}) + \mathbf{b} \text{ over the field } GF(2^8)$$

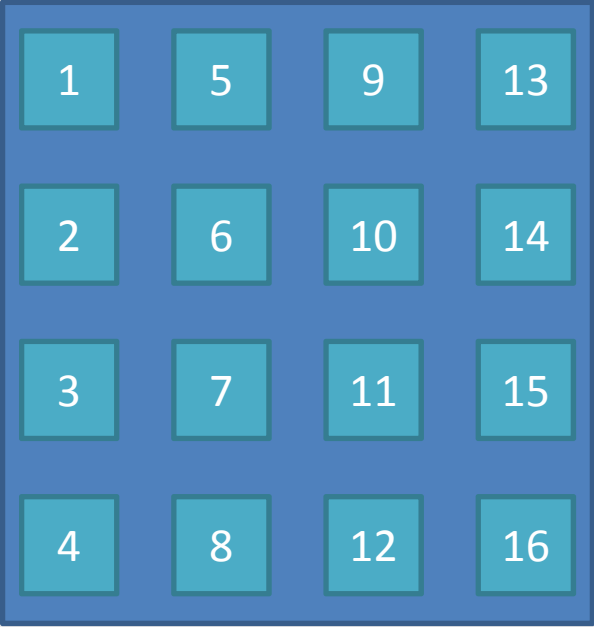
(\mathbf{M} is a matrix, \mathbf{b} is a constant)

Construction gives tight differential and linear bounds

AES OPERATIONS

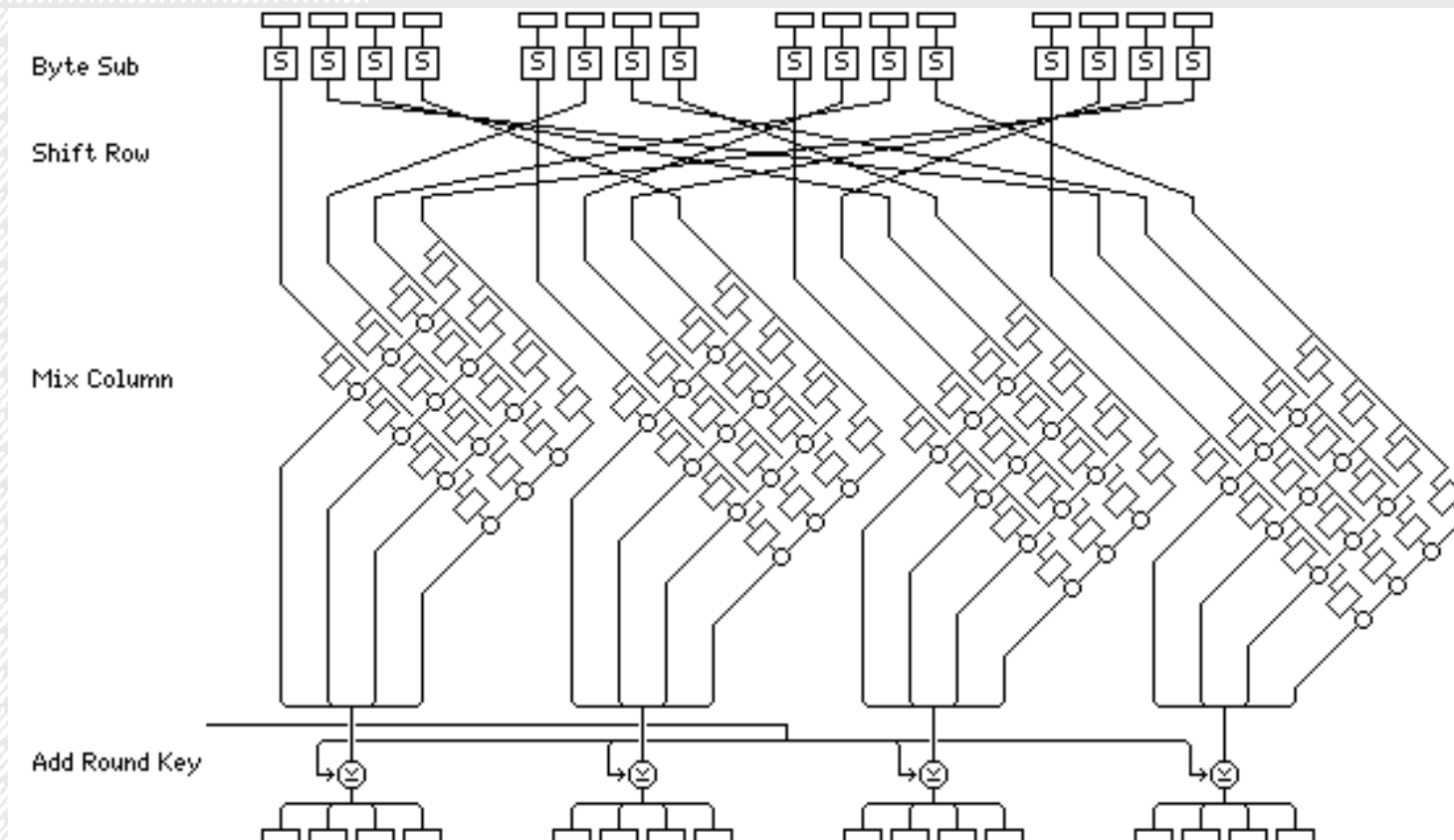
Linear transformation arranging 16 bytes of the value being encoded in a square and doing bitwise shuffling and mixing.

- **Step 1: Add Round Key**
 - simply XORs in the subkey for the current round
- **Step 2: Byte-sub**
 - S-box substitution
- **Step 3: Shuffle (ShiftRows)**
 - top row of four bytes is unchanged
 - second row is shifted one place left
 - third row is shifted two places left
 - fourth row is shifted three places left
- **Step 4: Mix Column**
 - four bytes in a column are mixed using a matrix multiplication
- **Result: Change in the input effects all of output in 2 rounds**



1	5	9	13
2	6	10	14
3	7	11	15
4	8	12	16

AES STRUCTURE



AES OVERVIEW

Number of rounds variable:

- 10 for 128-bit keys
- 12 for 192-bit keys
- 14 for 256-bit keys

Gives 50% margin of safety based on current known attacks

- Attack for 6 round 128 bit keys
- Attack for 7 round 192 bit keys
- Attack for 9 round 256 bit keys
- However require enormous amount of texts (certificational)

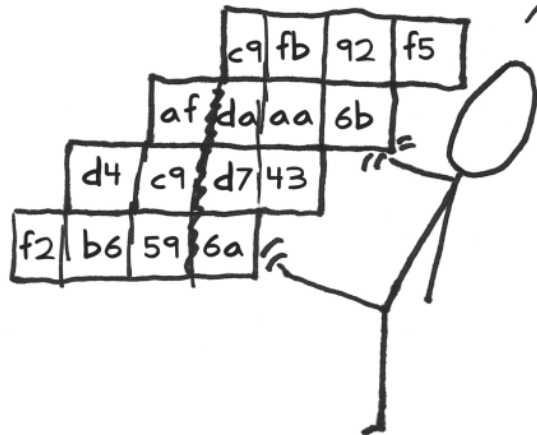
Safety against feasible attacks believed to currently be ~100%

HIGHLY RECOMMENDED: STICK FIGURE GUIDE TO AES

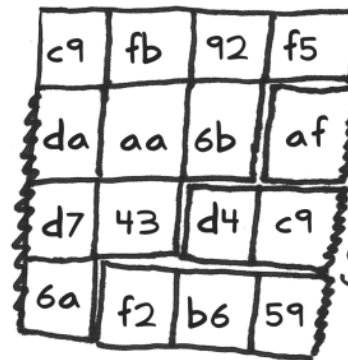
Applying Diffusion, Part 1: Shift Rows

Next I shift the rows to the left

Hiiii yaah!



...and then wrap them around the other side



Denotes
'permutation'



■ REFERENCES

Towards the 128-bit Era: AES Candidates (interest only)

<http://home.ecn.ab.ca/~jsavard/crypto/co0408.htm>

Stallings (3rd Ed)

§4

Handbook of Applied Cryptography

§7.1 - §7.4