Block ciphers

Bart Preneel

Katholieke Universiteit Leuven – Belgium

bart.preneel@esat.kuleuven.ac.be

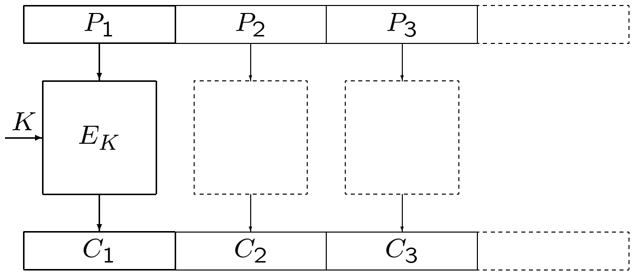
special thanks to Lars Knudsen and Vincent Rijmen

February 2003

Outline

- Definitions
- DES
- Modern cryptanalysis: linear, differential
- Modes of operation
- The wide trail design strategy
- AES & Rijndael
- Triple DES
- Advanced attacks
- Provable security against differential and linear cryptanalysis

Block Cipher Model



Transformation E_K takes one block at a time,
is identical for all blocks

Applications

- encryption
- pseudo-random string generation/stream ciphers
- hash functions
- one-way functions

Principles

$E : \{0,1\}^n \times \{0,1\}^k \to \{0,1\}^n$
$E_K : \{0,1\}^n \to \{0,1\}^n$

Block cipher: family of permutations in the block space

Every key selects one permutation

Block length $n \Rightarrow 2^n! \approx 2^{(n-1)2^n}$ permutations

Key length $k \Rightarrow 2^k$ selectable permutations

Shannon's Theory

Perfect secrecy in ciphertext-only attack obtained **only if** the key is used only once and $k > |P|$

Unicity distance: How many blocks of ciphertext are needed to be able to determine the secret key uniquely *in principle*?

$\min_x$: $H(K \mid C_1, C_2, \ldots, C_x) \approx 0$, $\quad x \approx k/(|P| \times R)$
with $R = 1 - H(P)/n$

If redundancy R_L is small, x is large. Use compression!

Known plaintext variant:
$\min'_x$: $H(K \mid P_1, \ldots P_{x'}, C_1, \ldots, C_{x'}) \approx 0$, $\quad x' \approx k/(|P|)$

x' is very small for popular block ciphers.

Block Cipher Design Goals

Provide less-than-unconditional security,
or 'practical' security

At a low cost (key length, performance, chip area, ...)

What is secure? [Shannon]

- secure against known attacks
- reducible to a problem known to be difficult

Assumptions on the Attacker

- ciphertext only
- known plaintext
- chosen plaintext
- chosen ciphertext
- adaptive chosen plaintext/ciphertext

Practical Security (1)

Given $(P_1, C_1), (P_2, C_2), (P_3, C_3), \ldots$, it should be 'difficult'

1. to recover the used key
2. to predict other (P_i, C_i) pairs
3. to predict information on P_j for known C_j
4. to characterize the family of 2^k permutations, or distinguish them from the other permutations

Practical Security (2)

Important design criteria [Shannon]

1. Nonlinearity (in plaintext and in key input) ('confusion')
2. Spreading of statistics over all bits ('diffusion')

product cipher: substitution $\times$ permutation

Importance of Nonlinearity

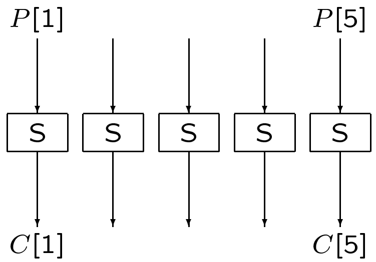
Known plaintext attacks:

$$\left.\begin{array}{l} P_1 \rightarrow C_1 \\ P_2 \rightarrow C_2 \end{array}\right\} \Rightarrow (P_1 + P_2) \rightarrow (C_1 + C_2)$$

Also dangerous: if $(P_1 + P_2) \rightarrow C_3 \approx (C_1 + C_2)$, or if $(P_1 + P_2) \rightarrow (C_1 + C_2)$ with large probability

Distance to all linear mappings is important

Importance of Diffusion



If $C[1]$ is only function of $P[1], K[1]$, then the cryptanalyst can solve separate 'sub-ciphers'

If $C[1]$ depends only very weakly on $K[i], P[j]$, then the cryptanalyst can solve separate problems

Practical Constraints

Block cipher should be realizable
in hardware and/or software

It should be possible to change the key easily

⇒ a designer has to use relatively small components, with a limited number of input bits, output bits

Smaller keys are more practical
Smaller blocks are more practical

'Security' imposes lower bounds on key length, block length

Practical Block Ciphers

Shannon:
alternate nonlinear substitutions and mixing functions

Feistel: repeat a 'simple' *round transformation*

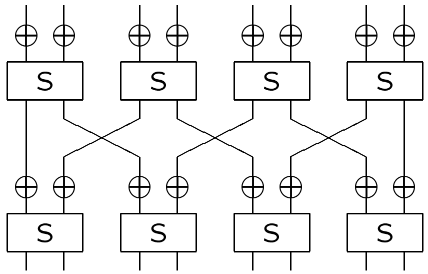
$$X \to f(X) = \mathcal{L}(\mathcal{S}(X + K))$$

$X + K$:	Addition of the (round) key
$\mathcal{S}$:	Nonlinear Substitution (small boxes)
$\mathcal{L}$:	Linear Mixing (interconnecting wiring)

Small code size, hardware area (important ??)

Iteration of a (weak) round transformation can make a strong cipher

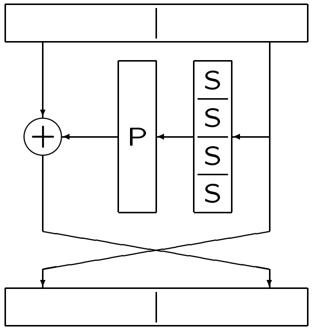
Iterated Block Ciphers



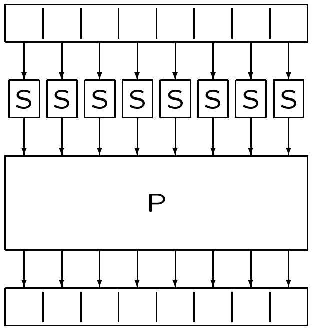
Iterated Block Ciphers

- Key schedule
  - Input: user selected key of k bits
  - Output: r round keys K_1, K_2, . . . , K_r
- C_0: plaintext
- $C_i = F(K_i, C_{i-1})$
- C_r: ciphertext
- F invertible for a fixed key, weak by itself
- most important block ciphers are iterated

Feistel Ciphers and S-P-Networks



Feistel Cipher



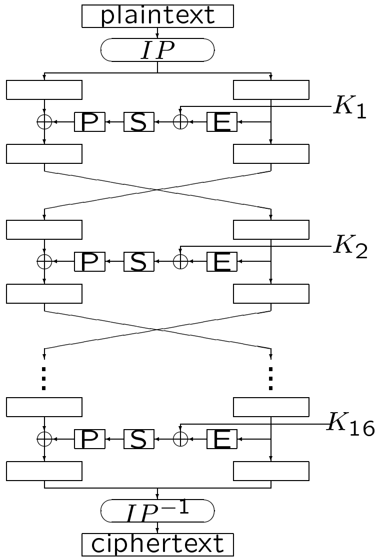
S/P-Network

DES: History

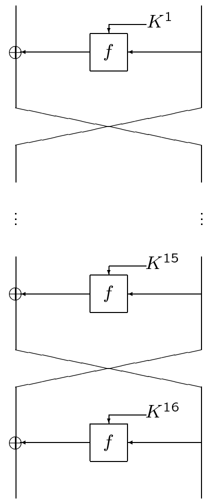
- 1973/74: NBS publishes a solicitation for encryption algorithms
- submission by IBM
- evaluation by NSA
- 1975: publication of proposed standard
- public discussion (trapdoors, key size)
- 1977: publication of FIPS 46 (DES)
- review every 5 years (1983-1988-1993)

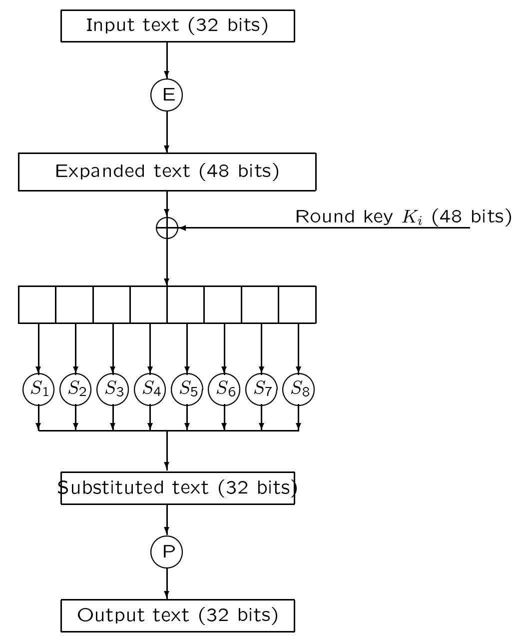
- IBM grants nonexclusive royalty-free license

DES: Description (1)



DES Description (2)





DES Description (5)

Input to round function expanded from 32 to 48 bits

48 bits split into blocks of each 6 bits

8 S-boxes, each taking 6 and returning 4 bits

S1:

e	4	d	1	2	f	b	8	3	a	6	c	5	9	0	7
0	f	7	4	e	2	d	1	a	6	c	b	9	5	3	8
4	1	e	8	d	6	2	b	f	c	9	7	3	a	5	0
f	c	8	2	4	9	1	7	5	b	3	e	a	0	6	d

S2:

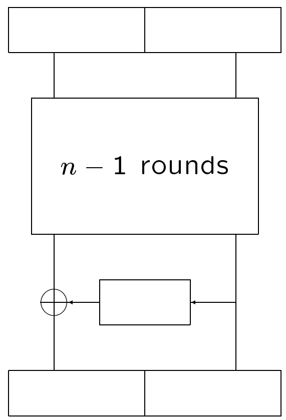
f	1	8	e	6	b	3	4	9	7	2	d	c	0	5	a
3	d	4	7	f	2	8	e	c	0	1	a	6	9	b	5
0	e	7	b	a	4	d	1	5	8	c	6	9	3	2	f
d	8	a	1	3	f	4	2	b	6	7	c	0	5	e	9

etc..

Modern Cryptanalysis

- differential cryptanalysis
- linear cryptanalysis
- related key attack
- interpolation attack
- integral attack
- variants of the above...

The Last Round Trick



isolate the last round

Generic Attack on Iterated Ciphers

$$P \longrightarrow \boxed{F} \longrightarrow C_1 \longrightarrow \boxed{F} \longrightarrow C_2 \;\; \ldots\ldots \;\; \longrightarrow C_{r-1} \longrightarrow \boxed{F} \longrightarrow C$$

(keys K_1, K_2, K_r entering the respective F boxes)

1. Assume "correlation between P and C_{r-1}
2. Get a pair P, C
3. Repeat for all values $\tilde{K_r}$ of K_r:

   a) Let $C' = F^{-1}(C \oplus \tilde{K_r})$, compute $x = \text{cor}(P, C')$

   b) If key gives $\text{cor}(P, C_{r-1})$, increment counter
4. Go to 2.

Note: when $\tilde{K_r} = K_r$, $x = \text{cor}(P, C_{r-1})$

Generic Attack: Success

Signal to noise ratio:

$$S/N = \frac{\#\text{ times correct key is suggested}}{\#\text{ times a random key is suggested}}$$

If $S/N \neq 1$, repeat until correct key "sticks out"

Success and complexity depends on how easy it is to distinguis $\text{cor}(P, C_{r-1})$ from $\text{cor}(P, C')$

Differential Cryptanalysis (DC)

Chosen plaintext attack

- Assume X, text, is combined with key, K, via group operation $\otimes$
- Define difference of X_1 and X_2 as

$$\Delta(X_1, X_2) = X_1 \otimes X_2^{-1}$$

- Difference invariant after combination of key

$$\begin{aligned} &\Delta(X_1 \otimes K, X_2 \otimes K) \\ &= (X_1 \otimes K) \otimes (X_2 \otimes K)^{-1} \\ &= X_1 \otimes K \otimes K^{-1} \otimes X_2^{-1} = \Delta(X_1, X_2) \end{aligned}$$

Definition of *difference* relative to cipher; often exclusive-or

Differential Cryptanalysis (2)

$$P \longrightarrow \boxed{F} \longrightarrow C_1 \longrightarrow \boxed{F} \longrightarrow C_2 \;\; \ldots\ldots \;\; \longrightarrow C_{r-1} \longrightarrow \boxed{F} \longrightarrow C$$

(keys K_1, K_2, K_r entering the respective F boxes)

Main criterion
distribution of differences through nonlinear components of F is non-uniform

Definition:
An s-round *characteristic* is a series of differences defined as an $(s+1)$-tuple: $\Omega : \{\alpha_0, \alpha_1, \ldots, \alpha_s\}$,
where $\Delta P = \alpha_0$, $\Delta C_i = \alpha_i$ for $1 \leq i \leq s$

$\text{Pr}(\Omega) = \text{Pr}(\Delta C_i = \alpha_i, \ldots.., \Delta C_1 = \alpha_1 | \Delta P = \alpha_0)$

Probability is taken over all possible plaintexts and keys

Principle of DC

Attack on an r-round iterated block cipher.
Finds find round key K_r (k bits)

Find $(r-1)$-round characteristic determining ΔC_{r-1} with probability p

Repeat

1. choose pairs of plaintexts with difference ΔP
2. get the pairs of ciphertexts C and C^*
3. for $i = 0, \ldots, 2^k - 1$ do:
   - From ciphertexts and the guess $K_r = i$, decrypt ciphertexts one round
   - if expected difference ΔC_{r-1} is obtained, counter for i incremented

until one counter has a value much larger than other.

Joining Characteristics

If	$\Delta P = \alpha_0$	leads to $\Delta C_1 = \alpha_1$	with probability p_1
and	$\Delta C_1 = \alpha_1$	leads to $\Delta C_2 = \alpha_2$	with probability p_2
then	$\Delta P = \alpha_0$	leads to $\Delta C_2 = \alpha_2$	with probability $p_1 \cdot p_2$

Is this true?

If so, under which conditions? (independent round keys?)

Probability of Characteristics

Want to know (k is secret key)

$$\text{Pr}_P(\Delta C_i = \alpha_i, \ldots.., \Delta C_1 = \alpha_1 | \Delta P = \alpha_0, K = k)$$

But k is unknown? Average over all keys:

$$\text{Pr}_{P,K}(\Delta C_i = \alpha_i, \ldots.., \Delta C_1 = \alpha_1 | \Delta P = \alpha_0)$$

can be hard to calculate

Definition: An iterated cipher is *Markov cipher*, with respect to the defined difference, if

$$\text{Pr}_K(\Delta C_1 = \beta \mid \Delta C_0 = \alpha, C_0 = \gamma)$$

is independent of γ for all α, β

Probability of Characteristics (2)

For Markov ciphers with independent round keys

$$\text{Pr}_{P,K}(\Delta C_s = \alpha_s, \ldots.., \Delta C_1 = \alpha_1 | \Delta P = \alpha_0) =$$
$$\text{Pr}_K(\Delta C_s = \alpha_s, \ldots.., \Delta C_1 = \alpha_1 | \Delta P = \alpha_0) =$$
$$\prod_{i=1}^{s} \text{Pr}_K(\Delta C_1 = \alpha_i | \Delta P = \alpha_{i-1})$$

Fact: DES is Markov cipher with difference $\oplus$

Iterative Differential Characteristics

Problem: for t large, t-round characteristics hard to find

Definition
An s-round iterative characteristic has the form

$$\Omega : \{\alpha_i, \alpha_{i+1}, \ldots, \alpha_{i+s+1}\},$$

where $\alpha_i = \alpha_{i+s+1}$.

Construct ts-round characteristics by concatenating Ω with itself t times.

Principle of DC - Filtering

- Right pair: pair of plaintexts with intermediate ciphertexts following the characteristic
- Wrong pair: not a right pair
- Right pairs always suggest the correct value of the key
- Strategy: minimise the number of wrong pairs
- Often possible from ciphertexts alone to determine that a pair is wrong; in that case the pair is *filtered out* (not used) in the analysis

Signal to Noise Ratio

$$S/N = \frac{\#\text{ times correct key is counted}}{\#\text{ times a random key is counted}}$$

k	number of key bits to find
p	probability of characteristic
m	number of pairs required
β	ratio of used pairs to all pairs
α	# keys suggested by each used pair

$$S/N \;=\; \frac{m \cdot p}{\frac{m \cdot \beta \cdot \alpha}{2^k}} \;=\; \frac{p \cdot 2^k}{\alpha \cdot \beta}$$

If $S/N \neq 1$ repeat attack until correct key "sticks out"

Complexity

- p_Ω: probability of characteristic Ω used

  Chosen plaintexts needed roughly

  $$c \times 1/p_\Omega$$

  where $c > 1$ is a function of S/N (usually small)
- Increase S/N ratio: filter out wrong pairs
- Filtering essential to most differential attacks
- Success of differential attacks depends on
  - probability of characteristic
  - number of counters required
  - S/N ratio
  - time to run the attack

Example Cipher – B32

Plaintext and ciphertext 32 bits, r rounds.
Round keys independent, uniformly random

One round:

1. Split text into 8 pieces of four bits (nibbles)
2. Exclusive-or round key to text
3. Evaluate each nibble through the S-box

x	0	1	2	3	4	5	6	7
$S(x)$	7	3	6	1	13	9	10	11
x	8	9	10	11	12	13	14	15
$S(x)$	2	12	0	4	5	15	8	14

   and concatenate the results.

4. Rotate the 32 bits two positions to the right

Exclusive-or round key to output of last round

Example B32 – Differential Attack (1)

Differential distribution table for S:

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0:	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1:	0	2	0	0	6	0	2	2	0	0	2	0	0	0	2	0
2:	0	4	6	0	0	0	0	2	2	0	0	0	0	2	0	0
3:	0	0	0	2	0	2	6	2	0	0	0	2	2	0	0	0
4:	0	0	0	2	0	0	0	2	2	0	8	0	2	0	0	0
5:	0	0	0	0	0	0	0	0	0	2	0	2	2	4	6	0
6:	0	0	2	0	0	2	0	0	4	0	2	4	0	2	0	0
7:	0	2	0	0	2	0	0	0	0	2	0	0	6	0	0	4
8:	0	0	2	0	0	6	4	0	2	0	0	0	0	0	0	2
9:	0	4	4	2	2	0	0	0	0	0	0	2	2	0	0	0
a:	0	0	0	0	4	2	0	6	0	0	0	0	0	2	0	2
b:	0	2	0	8	0	2	0	0	0	0	2	0	0	0	2	0
c:	0	0	2	0	0	2	0	0	0	0	2	0	2	0	2	6
d:	0	2	0	0	0	0	2	0	4	2	0	4	0	0	2	0
e:	0	0	0	2	0	0	0	2	2	0	0	0	0	6	2	2
f:	0	0	0	0	2	0	2	0	0	10	0	2	0	0	0	0

Example B32 – Differential Attack (2)

Consider combination $1 \xrightarrow{S} 4$ with probability $\frac{6}{16}$

Use it to build one-round iterative characteristic

$$00000001_x \xrightarrow{S} 00000004_x \xrightarrow{R} 00000001_x$$

Build s-round characteristic

$$00000001_x \overset{s \text{ rounds}}{\longrightarrow} 00000001_x$$

of probability $(\frac{6}{16})^s$

Example B32 – Differential Attack (3)

Consider B32 with 5 rounds

4-round iterative characteristic with prob. $(\frac{6}{16})^4 \simeq 1/51$

In analysis use only ciphertexts of difference

$$R(0000000h_x)$$

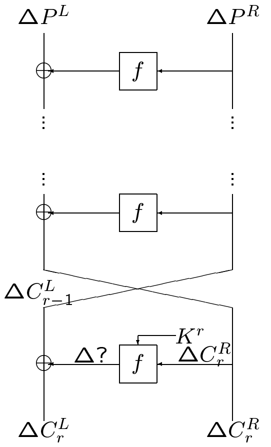
for some hex value h, where R is the rotation by two positions

Signal-to-noise ratio: $k = 4$, $p = 1/51$, $\beta = 1/51$, $\alpha = 1$

$$S/N \ = \frac{p \cdot 2^k}{\alpha \cdot \beta} = 16$$

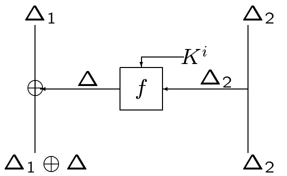
Finds 4 bits of final round key using 102 pairs of plaintexts

Feistel Cipher: Differential Cryptanalysis



Feistel Cipher: Joining Differential Char.

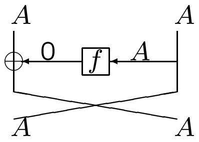
- suppose we wish to add a round to a characteristic with output difference (Δ_1, Δ_2)



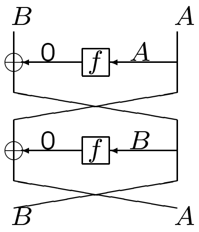
- after swap of left and right, the left output difference is always equal to the right input difference
- assuming that rounds are independent, probabilities can be multiplied (DES is a Markov cipher)

Feistel Cipher: Iterative Characteristics

A one-round iterative characteristic

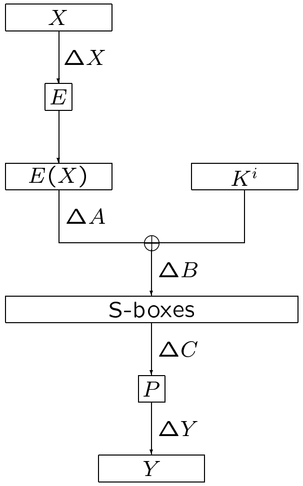


A two-round iterative characteristic

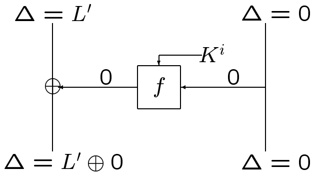


Requirement: f not permutation (for a fixed key)
Special case: $A = 0$ (higher probability)

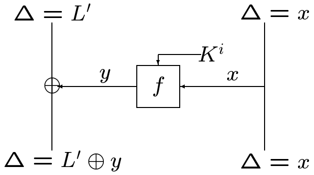
DES: Differential Cryptanalysis (Feistel)



DES: A One-Round Characteristic



One-round characteristic with $p = 1$



one-round characteristic with $p = \frac{14}{64}$ when

$x = 6000\,0000_x$ and $y = 0080\,8200_x$

DES: Difference Distribution for S1

	0	1	2	3	4	5	6	...	a	b	c	d	e	f
0:	64	0	0	0	0	0	0	...	0	0	0	0	0	0
1:	0	0	0	6	0	2	4	...	12	4	10	6	2	4
2:	0	0	0	8	0	4	4	...	8	6	12	6	4	2
3:	14	4	2	2	10	6	4	...	4	0	2	2	2	0
4:	0	0	0	6	0	10	10	...	6	4	2	8	6	2
5:	4	8	6	2	2	4	4	...	4	0	12	2	4	6
6:	0	4	2	4	8	2	6	...	4	2	4	2	0	12
7:	2	4	10	4	0	4	8	...	8	2	2	2	4	4
8:	0	0	0	12	0	8	8	...	2	8	8	2	2	4
9:	10	2	4	0	2	4	6	...	8	0	10	0	2	12
a:	0	8	6	2	2	8	6	...	6	0	4	0	2	10
b:	2	4	0	10	2	2	4	...	2	6	6	4	2	12
c:	0	0	0	8	0	6	6	...	6	4	6	6	14	2
d:	6	6	4	8	4	8	2	...	4	6	0	2	0	2
e:	0	4	8	8	6	6	4	...	4	0	0	4	0	8
f:	2	0	2	4	4	6	4	...	2	2	2	6	8	8
10:	0	0	0	0	0	0	2	...	6	12	4	6	8	6
11:	6	8	2	4	6	4	8	...	6	6	0	4	0	0
...														
...														
39:	6	2	2	4	12	6	4	...	2	4	2	4	4	0
3a:	6	4	6	4	6	8	0	...	6	2	2	6	4	0
3b:	2	6	4	0	0	2	4	...	8	6	4	4	6	2
3c:	0	10	4	0	12	0	4	...	4	12	4	4	2	0
3d:	0	8	6	2	2	6	0	...	0	4	0	12	4	4
3e:	4	8	2	2	2	4	4	...	0	2	0	8	4	4
3f:	4	8	4	2	4	0	2	...	4	8	8	6	2	2

DES: S1 (2)

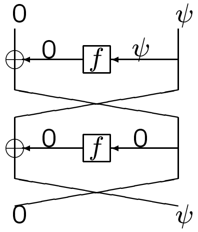
Consider S1 and the combination $3_x \rightarrow 0_x$
It follows by easy calculations that for

P_1^*	P_2^*	P_1^*	P_2^*
8	b	b	8
11	12	12	11
15	16	16	15
18	$1b$	19	$1a$
$1a$	19	$1b$	18
25	26	26	25
39	$3a$	$3a$	39

it holds that $S1(P_1^*) \oplus S1(P_2^*) = 0_x$

DES: 2-Round Iterative Differential Char.

A two-round iterative characteristic

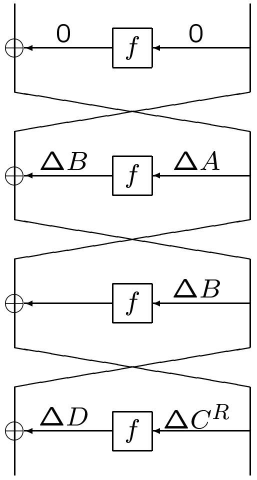


with $\psi = 19600000$: $p_\psi = 14/64 \cdot 8/64 \cdot 10/64 \approx 1/234$.

This is the best characteristic known for 8 rounds or more probability for $2r$ rounds is $1/234^r$.

Note that the probability is in fact key dependent (depending on overlapping key bits between S-boxes)

DES Reduced to 4 Rounds



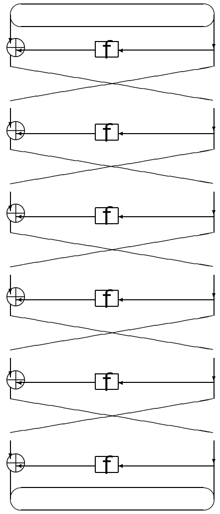
ΔA different inputs to only 1 S-box
28 bits known of ΔB and
28 bits known of ΔD

Count on 6 key bits:
$S/N = \frac{1 \cdot 2^6}{4} = 16$

Count on 12 key bits:
$S/N = \frac{1 \cdot 2^{12}}{16} = 256$

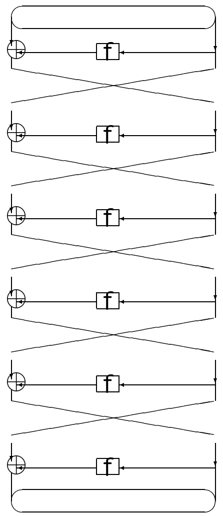
Note: independent of values
of S-boxes!

DES Reduced to 10 Rounds



use 8-round iterative characteristic
with $p = (1/234)^4 \approx 2^{-31.5}$
count on 18 key bits
$S/N = \frac{2^{18} . 2^{-31.5}}{4^3 . 2^{-52}} \simeq 2^{33}$
number of text pairs: $4 \cdot 2^{31.5}$

The Full DES



use 14-round iterative characteristic
with $p = (1/234)^7 \approx 2^{-55.1}$
count on 18 key bits
$S/N = \frac{2^{18} . 2^{-55.1}}{4^3 . 2^{-52}} \simeq 2^9$
number of text pairs: $4 \cdot 2^{55.1}$

The Full DES: DC Doesn't Work

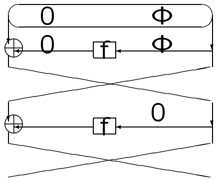
- not faster than exhaustive key search on 16-round DES: needs 2^{57} pairs
- DES designed to prevent differential attacks

Refined Attack [Biham-Shamir 1991]

- first round trick (one round "for free")
- success as soon as one right pair is encountered
- 56-bit key tested at once, no counters
- implemented at UiB in 1997 for subclass of keys

DES: The First Round Trick

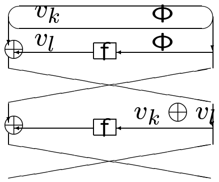
Two rounds in a characteristic.



- $0 \leftarrow \Phi$ with prob. p
- $2N$ plaintexts $\rightarrow$ N pairs
- After one round $N \times p$ pairs have expected difference

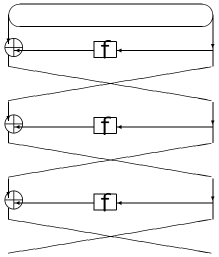
DES: The First Round Trick (2)

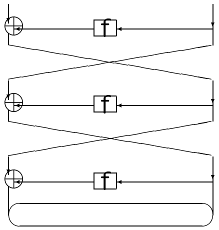
Two rounds in a characteristic.



- Assume $\{v \mid v \leftarrow \Phi\} = \{v_1, ..., v_N\}$ (closed)
- Choose N plaintexts $(P^L \oplus v_j \mid P^R \oplus \Phi)$
- Choose N plaintexts $(P^L \oplus v_i \mid P^R)$
- $2N$ plaintexts $\rightarrow$ N^2 pairs
- N possible values of v_k, v_l fixed
- $v_l = v_k$ in N pairs of the N^2 pairs
- After one round N pairs have expected difference

The Full DES – Revisited





Charact. prob. $(1/234)^6 \simeq 2^{-47.2}$

Linear Cryptanalysis, Matsui 1993

- Known plaintext attack
- Uses linear relations between bits of plaintext P and ciphertext $C = E_K(P)$ to determine (bits of) key K

- Suppose with probability $p \neq \frac{1}{2}$

$$(P \cdot \alpha) \oplus (C \cdot \beta) = 0 \quad (**)$$

- Collect N pairs of plaintext/ciphertext (using 1 key!)
- T : number of times left side of (**) is 0
  If $p > 1/2$, $E(T) > N/2$,
  if $p < 1/2$, $E(T) < N/2$,

- If P and C are independent, $T \simeq N/2$.

Linear Attack: Complexity

- $\mathbf{T}$ binomial random variable 0 with $p > 1/2$

$$\begin{aligned}
\Pr(T > N/2) &= 1 - \Pr(T \leq N/2) \\
&\simeq 1 - \Phi\left(\frac{N/2 + 1/2 - Np}{\sqrt{p(1-p)} \times \sqrt{N}}\right) \\
&\simeq 1 - \Phi(-2\sqrt{N}\,|p - 1/2|) \\
&= \Phi(2\sqrt{N}\,|p - 1/2|)
\end{aligned}$$

  where Φ is the normal distribution function
- if $N = |p - 1/2|^{-2}$ the success rate is about 97.72%
- we call $|p - 1/2|$ the *bias*

Joining Linear Characteristics

Random boolean variables X, Y, and Z

If	$\alpha \cdot X = \beta \cdot Y$	with bias $\lvert p_1 - 1/2\rvert$
and	$\beta \cdot Y = \gamma \cdot Z$	with bias $\lvert p_2 - 1/2\rvert$
then	$\alpha \cdot X = \gamma \cdot Z$	with bias $2 \times \lvert p_1 - 1/2\rvert \times \lvert p_2 - 1/2\rvert$

Piling Up-Lemma
Let Z_i, $1 \leq i \leq n$, be indep. random boolean variables, which are 0 with prob. p_i. Then

$$\Pr(Z_1 \oplus Z_2 \oplus .... \oplus Z_n = 0) = 1/2 + 2^{n-1} \prod_{i=1}^{n} (p_i - 1/2)$$

The Piling-Up Lemma is applicable to iterated ciphers with independent round keys

Linear Characteristics - Iterated Ciphers

$$C_i \longrightarrow \overset{\overset{K}{\downarrow}}{\oplus} \longrightarrow D \longrightarrow \boxed{f} \longrightarrow C_{i+1}$$

$(\alpha \cdot C_i) \oplus (\alpha \cdot D) = (\alpha \cdot K)$

$(\alpha \cdot D) = (\beta \cdot C_{i+1})$ with $p_i \neq 1/2$

$(\alpha \cdot C_i) \oplus (\alpha \cdot K) = (\beta \cdot C_{i+1})$ with $p_i \neq 1/2$

$(\alpha \cdot C_i) = (\beta \cdot C_{i+1})$ with bias $|p_i - 1/2|$

Characteristic (δ_i, δ_{i+1}) with bias $|p_i - 1/2|$ means that $\delta_i \cdot C_i = \delta_{i+1} \cdot C_{i+1}$ with probability p_i (or $1/2 - p_i$).

r-round characteristic $(\delta_0, \ldots, \delta_r)$ with bias $|p - 1/2|$ means that $\delta_0 \cdot C_0 = (\delta_r \cdot C_r)$ with probability p (or $1/2 - p$).

Linear Attack - r-Round Iterated Cipher

$$P = C_0, \quad C_i \longrightarrow \overset{\overset{K_i}{\downarrow}}{\boxed{g}} \longrightarrow C_{i+1}, \quad C = C_r$$

- Consider r-round characteristic $(\delta_0, \ldots, \delta_r)$:

$$(P \cdot \delta_0) \oplus (C \cdot \delta_r) = 0 \quad \text{with some bias } b$$

- Consider

$$(P \cdot \delta_0) \oplus g^{-1}(C, K_r) \cdot \delta_{r-1} = 0 \quad \text{with bias } b'. \quad (*)$$

- For the right value of K_r, (*) is a characteristic for $r-1$ rounds, and in this case $b' \geq b$
- Assumption: for wrong values of K_r, (*) is a random approximation with bias $\simeq 0$

Linear Attack (2)

- Perform attack for all values of K_r
- Take value for K_r, which produces bias closest to expected
- Complexity $N \simeq c \cdot |p - 1/2|^{-2}$, c small constant
- Extension:

  $(g(P, K_1) \cdot \delta_1) \oplus (g^{-1}(C, K_r) \cdot \delta_{r-1}) = 0 \quad$ with bias b''.

  Note that $b'' \geq b' \geq b$

  Repeat basic attack for all values of K_1, K_r

Probability of Linear Characteristics

Want to know (k is secret key)

$$\mathrm{Pr}_P(C_r \cdot \delta_i = P \cdot \delta_0 \mid K = k)$$

But k unknown? Average over all keys:

$$\mathrm{Pr}_{P,K}(C_r \cdot \delta_i = P \cdot \delta_0)$$

can be hard to calculate

If $\mathrm{Pr}_K(C_r \cdot \delta_i = P \cdot \delta_0) = \mathrm{Pr}_K(C_i \cdot \delta_i = C_{i-1} \cdot \delta_0 \mid C_{i-1} = \gamma)$ and independent round keys, then

$$\mathrm{Pr}_{P,K}(C_r \cdot \delta_i = P \cdot \delta_0)$$

can be calculated from one-round probabilities and Piling-up Lemma.

Iterative Linear Characteristics

Problem: for t large, t-round characteristics hard to find

Definition
An s-round iterative characteristic has the form

$$\Omega : \{\delta_i, \delta_{i+1}, \ldots, \delta_{i+s+1}\},$$

where $\delta_i = \delta_{i+s+1}$.

Construct ts-round characteristics by concatenating Ω with itself t times.

Example Cipher – B32

Plaintext and ciphertext 32 bits, r rounds.
Round keys independent, uniformly random

One round:

1. Split text into 8 pieces of four bits (nibbles)
2. Exclusive-or round key to text
3. Evaluate each nibble through the S-box

x	0	1	2	3	4	5	6	7
$S(x)$	7	3	6	1	13	9	10	11
x	8	9	10	11	12	13	14	15
$S(x)$	2	12	0	4	5	15	8	14

   and concatenate the results.
4. Rotate the 32 bits two positions to the right

Exclusive-or round key to output of last round Round keys independent, uniformly random

Example B32 – Linear Attack (1)

Linear approximation table for S (entries are $(p-1/2)*16$)

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0:	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1:	0	2	0	-2	0	6	0	2	2	0	-2	0	-2	0	2	0
2:	0	-4	0	0	-2	2	-2	-2	0	0	-4	0	2	2	-2	2
3:	0	-2	0	-2	-2	0	-2	0	2	0	2	0	0	2	0	-6
4:	0	2	0	2	0	-2	0	-2	6	0	-2	0	2	0	2	0
5:	0	0	-4	0	0	0	4	0	0	4	0	0	0	4	0	0
6:	0	2	-4	2	2	0	-2	0	-2	-4	-2	0	0	2	0	-2
7:	0	0	0	0	2	2	2	-6	0	0	0	0	-2	-2	-2	-2
8:	0	-4	-2	-2	2	-2	0	0	0	0	-2	2	-2	-2	4	0
9:	0	2	-2	0	-6	0	0	-2	-2	0	0	2	0	-2	2	0
10:	0	0	2	2	0	0	2	2	0	0	-2	6	0	0	-2	-2
11:	0	-2	2	4	0	2	2	0	-2	0	0	-2	2	0	4	-2
12:	0	-2	-2	0	-2	0	4	2	2	-4	0	-2	0	-2	-2	0
13:	0	0	2	2	-2	-2	0	0	0	0	-2	-2	-6	2	0	0
14:	0	-2	-2	4	0	2	-2	0	2	0	4	2	-2	0	0	2
15:	0	0	2	-2	0	0	2	-2	0	-4	2	2	0	4	2	2

Example B32 – Linear Attack (2)

Consider the combinations
$4 \xrightarrow{S} 8$ with bias $\frac{6}{16}$ and $2 \xrightarrow{S} 1$ with bias $\frac{4}{16}$ and

Use it to build one-round characteristics

$$40000000_x \xrightarrow{S} 80000000_x \xrightarrow{R} 20000000_x$$

$$20000000_x \xrightarrow{S} 10000000_x \xrightarrow{R} 04000000_x$$

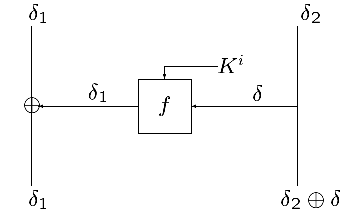
Build 2-round characteristic

$$40000000_x \xrightarrow{\text{2 rounds}} 04000000_x$$

with bias $2\frac{6}{16}\frac{4}{16} = \frac{3}{16}$

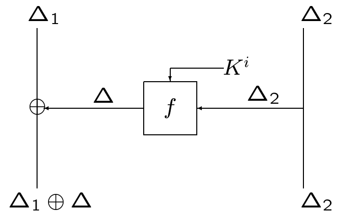
Build $2s$-round characteristic with bias $2^{s-1}(\frac{3}{16})^s$

Feistel Cipher: Joining Linear Char.

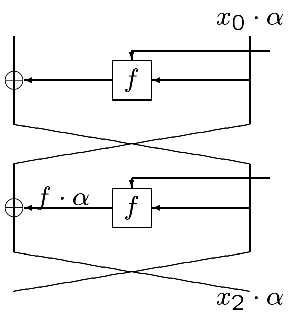


δ denotes a bit mask

duality with joining characteristics for DC

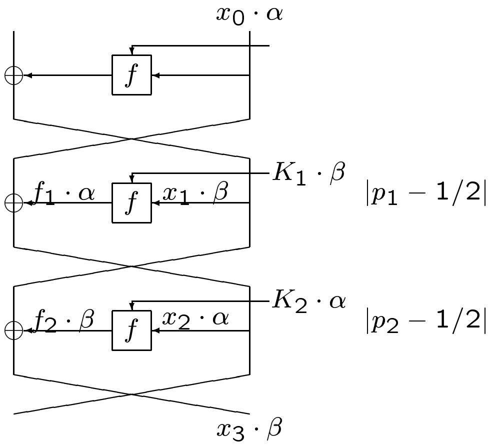


Feistel Cipher: 2-round Iterative Char.



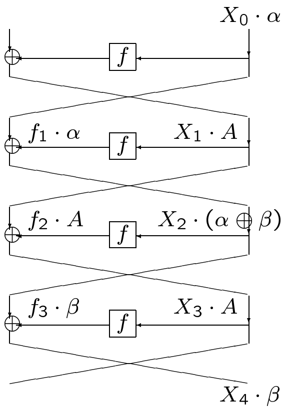
- Only non-trivial when output bits of f are not balanced

Feistel Cipher: 3-round Iterative Char.



$$
\begin{aligned}
(x_0 \cdot \alpha) \oplus (x_3 \cdot \beta) &= (f_1 \cdot \alpha \oplus x_2 \cdot \alpha) \oplus (f_2 \cdot \beta \oplus x_1 \cdot \beta) \\
&= (f_1 \cdot \alpha \oplus x_1 \cdot \beta) \oplus (f_2 \cdot \beta \oplus x_2 \cdot \alpha) \\
&= 0 \text{ with bias } 2|p_1 - 1/2||p_2 - 1/2|
\end{aligned}
$$

Feistel Cipher: 4-round Iterative Char.



$$
|\Pr_X(X_4 \cdot \beta = X_0 \cdot \alpha) - 1/2| = 2^2 \times |p_1 - 1/2| \times |p_2 - 1/2| \times |p_3 - 1/2|
$$

LC Applied to DES (Feistel)

- All components linear, except S-boxes
- "Probabilistic linearisation" of S-boxes possible
- DES S-box: 6 input bits $X = x_1, ..., x_6$ and 4 output bits $Y = y_1, ..., y_4$
- For each S-box:
  $\forall\ X, \alpha$ and β, find probability $p_{\alpha,\beta}$:

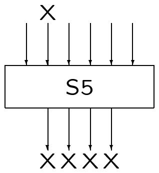
  $$X \cdot \alpha = Y \cdot \beta,$$

  where $Y =$ S-box(X)
- The higher bias, the better the attack

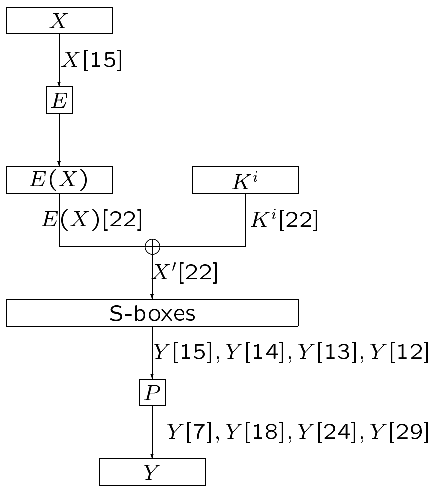
DES: S-box 5 (entries are $(p_{\alpha,\beta} - 1/2) * 64$)

$\alpha \backslash \beta$	1	2	3	...	9	10	...	13	14	15
1	0	0	0	...	0	0	...	0	0	0
2	4	−2	2	...	0	2	...	−2	0	−4
3	0	−2	6	...	0	−2	...	−2	4	−4
⋮										
16	2	−2	0	...	−2	−2	...	2	10	−20
17	2	−2	0	...	2	2	...	−6	2	4
18	−2	0	−2	...	6	4	...	4	−6	0
⋮										
63	−4	−4	−4	...	−2	6	...	0	−4	0

most extreme bias: for 12 out of 64 possible inputs

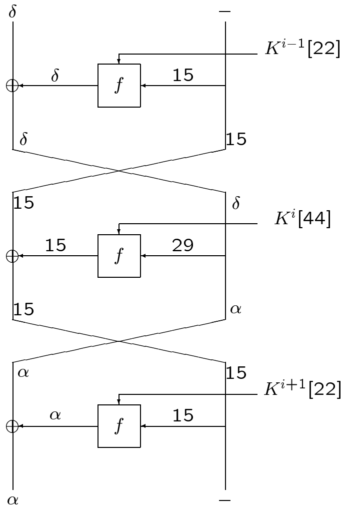


DES: One-Round Linear Characteristic



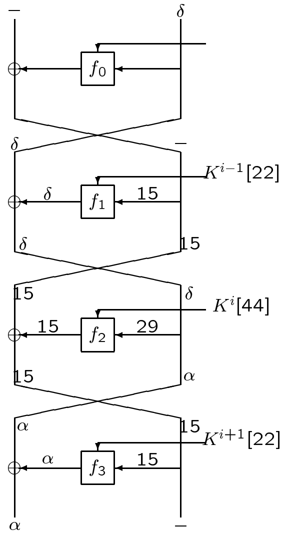
$X[15] \oplus f(X, K^i)[7, 18, 24, 29] = 0$ with $|p - 1/2| = 20/64$

DES: Three-Round Linear Characteristic



$\alpha = [7, 18, 24, 29]$, $\delta = [7, 18, 24]$
holds with probability $\frac{1}{2} + 6.1 \times 10^{-3}$

DES: 4-round Iterative Char., Bias $2^{-7.4}$



$\alpha = [7, 18, 24, 29]$, $\delta = [7, 18, 24]$

Linear Attack on DES (1)

- Concatenate 4-round characteristic with itself
- Yields 14-round characteristic with bias 1.2×2^{-21}
- Guess 6 round key bits in both first and last rounds

$$(P \cdot \delta) \oplus (f(P^R, k_1) \cdot \delta_1) \oplus$$
$$(C \cdot \delta_r) \oplus (f(C^R, k_r) \cdot \delta_{r-1}) = 0$$

- Gives 12 key bits
- Swap role of plaintext and ciphertext, repeat attack
- In total, one gets 24 bits of key information
- Optimization: key ranking
  Take n most likely values for the 24 bits,
  try exhaustively the remaining 32 bits

Linear Attack on DES (2)

- Estimate - with 2^{45} known plaintexts a DES key can be recovered with 98.8% success rate
- Matsui-test:
  - January, 1994
  - key found in 50 days on 12 HP9735 workstations – (120 MIPS each)
  - 2^{43} known plaintexts
- Ciphertext only attack possible, assuming English plaintexts encoded in ASCII

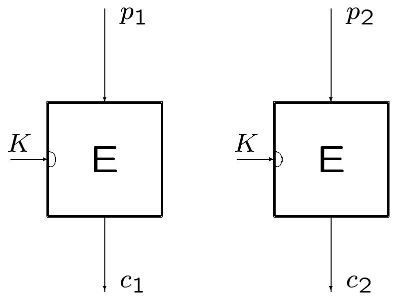
Chosen Plaintext Linear Attack on DES

(Knudsen-Mathiassen, 2000)

- Better bias by careful choice of plaintexts
- Basic attack: Matsui finds 13 key bits, K-M finds 12 key bits

	Matsui's attack			K-M's attack		
Plaintexts	2^{43}	2^{44}	2^{45}	2^{40}	2^{41}	2^{42}
Success rate	32%	78%	99%	6%	32%	86%

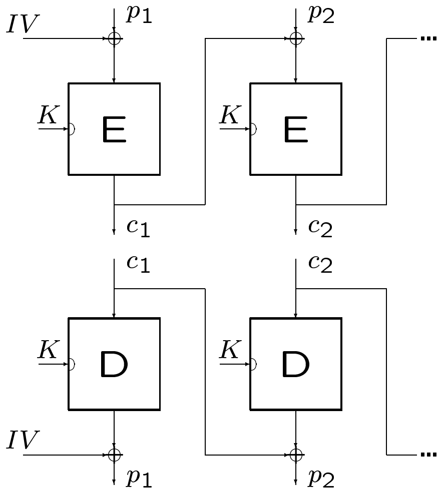
Block cipher: modes



ECB: properties

- blocks independent
- does not hide patterns and repetitions (attacker can use dictionary with known plaintext/ciphertext pairs)
- errors: expansion within 1 block
- limited number of applications: e.g. encryption of keys

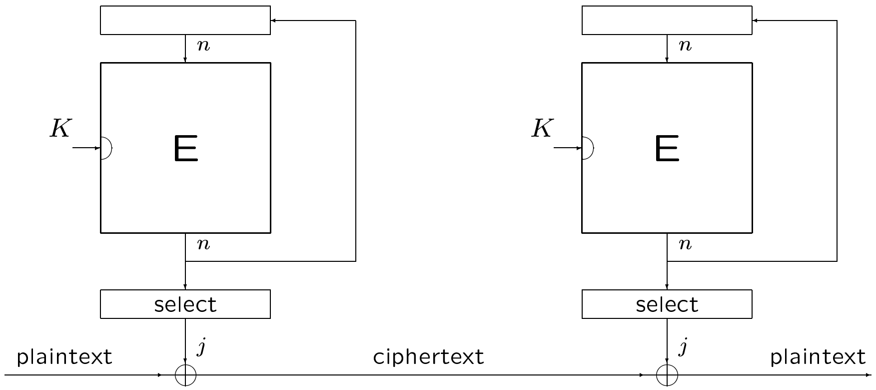
Block cipher: CBC mode



CBC: properties

- ciphertext depends on all previous plaintext blocks (internal memory)
- different IV hides patterns and repetitions
- error propagation: expansion in 1 block, copied into next block
- decryption of 1 block requires only ciphertext of previous block
- 'default' mode

Block cipher: OFB mode

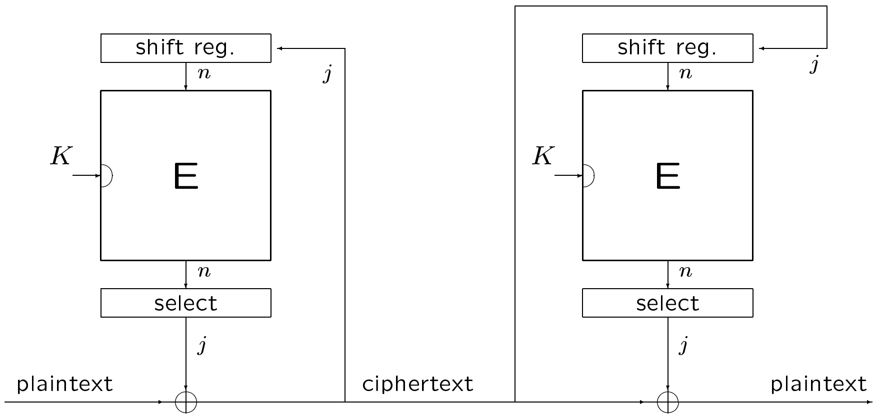


OFB: properties

additive stream cipher

- no linking between subsequent blocks
- different IV necessary; otherwise insecure
- uses only encryption
- if $j < n$: more effort per bit
- key stream independent of plaintext: can be precomputed
- no error propagation: errors are only copied

Block cipher: CFB mode



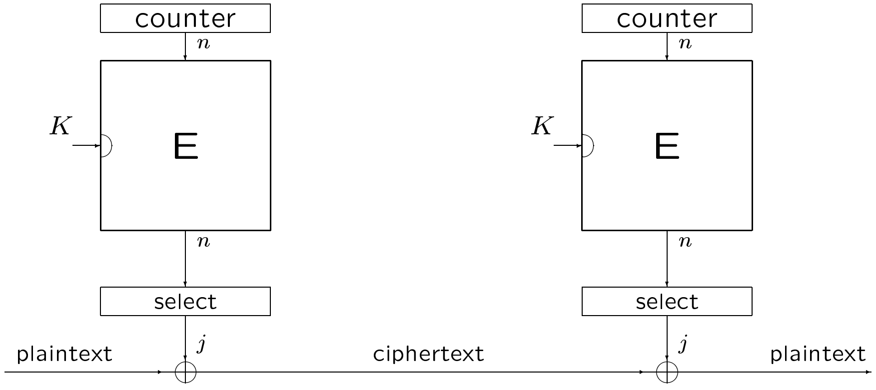
CFB: properties

self-synchronizing stream cipher

- ciphertext depends on all previous plaintext blocks (internal memory)
- different IV hides patterns and repetitions
- uses only encryption
- if $j < n$: more effort per bit
- error propagation: error copied and propagated over 1 block
- $j = 1$ no synchronization needed between sender and receiver; synchronization if n bits have been received correctly (if $j > 1$: only character level synchronization)

often used with $j = 1, 7, 8$ for synchronization

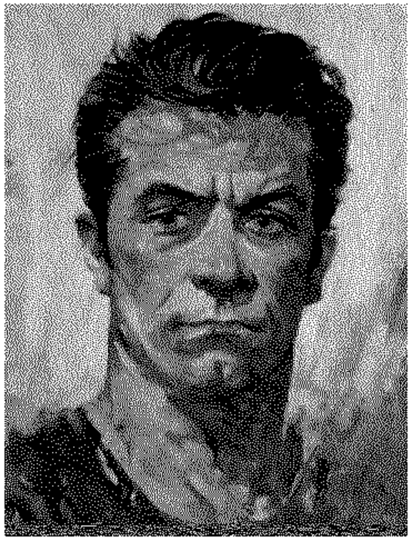
Block cipher: counter mode



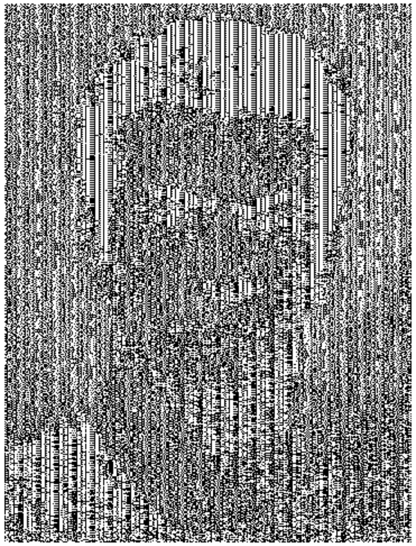
Overview modes

	ECB	CBC	OFB/CTR	CFB
patterns	−	+	+	+
repetition	−	IV	IV	IV
block size	n	n	j	j
error prop.	1 block	2 blocks	NONE	1 block + 1 char
synchr.	per block	per block	exact	per j $j = 1$: self-synchr.
speed	1	1	j/n	j/n
application	keys	general	no error prop..	synchron.

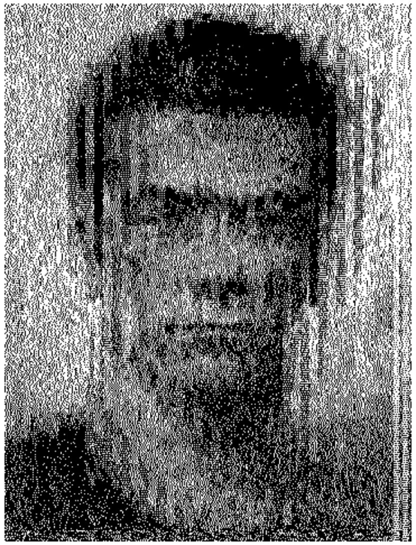
Plaintext image



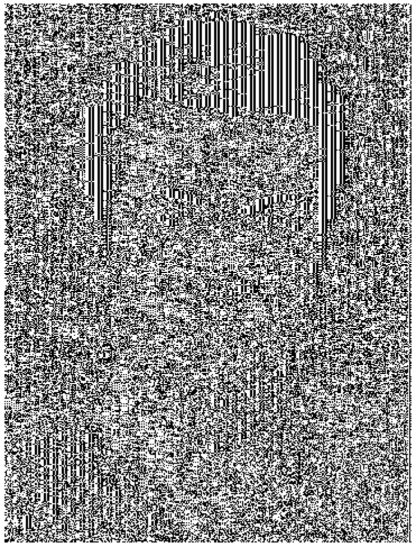
Substitution cipher



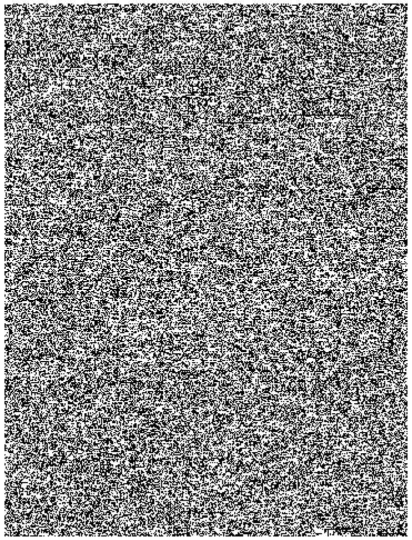
Transposition cipher



Block cipher in ECB mode



Blok cipher in CBC mode



The birthday paradox

Given a set with S elements.
Chooses r elements at random (with replacements), with $r \ll S$.
Find probability p that there at least two identical elements.

$$
\begin{aligned}
q &= 1 - p = 1 \cdot \frac{S-1}{S} \cdot \frac{S-2}{S} \cdots \frac{S-(r-1)}{S} = \prod_{k=1}^{r-1} \left(1 - \frac{k}{S}\right) \\
\ln q &= \sum_{k=1}^{r-1} \ln\left(1 - \frac{k}{S}\right) \approx -\sum_{k=1}^{r-1} \frac{k}{S} = -\frac{r(r-1)}{2S}
\end{aligned}
$$

$$
p = 1 - \exp\left(-\frac{r(r-1)}{2S}\right) \qquad \text{if } r = \sqrt{S} : p = 39\%
$$

birthday paradox: for $S = 365$ and $r = 23$, $p = 1/2$

Collisions: CBC and CFB (1)

In CBC mode:
after $2^{n/2}$ plaintext blocks, there will be two identical ciphertext blocks with probability 39% (birthday paradox)

$C_i = C_k$ implies $P_i \oplus C_{i-1} = P_k \oplus C_{k-1}$ or $P_i \oplus P_k = C_{i-1} \oplus C_{k-1}$
ciphertext reveals information on plaintext!

In CFB mode:
afer $j \cdot 2^{n/2}$ plaintext bits, there is a probability of 39% for two identical ciphertext blocks (and identical shift registers).

$\leadsto$ linear relation between 2 plaintext characters

$$P_i[j] \oplus P_k[j] = C_i[j] \oplus C_k[j]$$

Collisions: CBC and CFB (2)

if $n = 64$: after 2^{32} plaintexts
encryption speed of 64 Mbit/s: collisions after 1 hour

after 2^{40} plaintexts (10 days) one expects

$$(2^{40})^2/2^{65} = 2^{15} \quad \text{collisions}$$

solution: change key frequently or increase block length

Collisions: OFB

- If one feeds back in OFB $< n$ bits (not allowed by the standards!), then the next state function is a random function (rather than a permutation).

  birthday paradox $\rightsquigarrow$ key stream period of $\approx 2^{n/2}$

- Standard feeds back the full n bits; then the next state function is a random permutation

  One shows that the average period is then equal to $= 2^{n-1}$
  (the probability that a state lies in a cycle of length $< c$ equals $c/2^n$).

The Wide Trail Strategy

Differential cryptanalysis: Propagation of differences

$$\begin{array}{lcl} P_1 & \to & C_1 \\ P_1 + \delta & \to & C_1 + \epsilon \end{array}$$

Look for special values of ϵ
Probabilistic attack

Linear cryptanalysis: exploit input-output correlations

Estimation of data complexity

Linear Cryptanalysis

Matsui '93 (on DES)

Observation: some of the S-boxes have a relatively small distance to linear mappings, e.g.

$$S_5 : X_5 + Y_1 + Y_2 + Y_3 + Y_4 = 0 \text{ with prob. } 3/16$$

Other formalism:

$$c(X_5, Y_1 + Y_2 + Y_3 + Y_4) = -5/8$$

'linear approximation' of the S-box $\to$ linear approximation of the round function

$$c(X_i + X_j + Y_u + Y_v, K_a + K_b) = c_1$$

Linear Cryptanalysis

Study correlation between linear combinations of input bits and output bits

- linear operation: $c = 1$ or $c = 0$
- affine operation: $c = \pm 1$ or $c = 0$
- non-linear operation: $-1 \leq c \leq 1$

Linear Cryptanalysis

Approximation of multiple rounds ...

Typically, the more rounds, the smaller $|p - 0.5|$

Idea: guess parts of the round keys in first and last rounds
$\to P'_i, P'_j, C'_u, C'_v$.

$$P'_i + P'_j + C'_u + C'_v = \text{constant, with prob. } p'$$

Measure p' experimentally
Measured p' is likely to be maximal for the correct guesses of round keys

Guess more key bits ...

Differential Cryptanalysis

Biham and Shamir '91 (on DES)

Look at pairs of inputs: $X_1, X_2 = X_1 + \delta$,
outputs: $Y_1, Y_2 = Y_1 + \epsilon$

General observations:
- linear operators: $L(X + \delta) = L(X) + L(\delta)$

$$\begin{aligned}\epsilon &= Y_2 - Y_1 = L(X_2) - L(X_1) = L(X_2 - X_1)\\ &= L(\delta)\end{aligned}$$

- simple key addition:

$$\epsilon = Y_2 - Y_1 = (X_2 + K) - (X_1 + K) = X_2 - X_1 = \delta$$

- nonlinear operators: $\epsilon = S(X_2) - S(X_1)$ depends on the values of X_1 and X_2 (except if $\delta = 0$, then $\epsilon = 0$)

Differential Cryptanalysis

Observations on the round function of the DES:
- hwt(δ) large $\Rightarrow$ hwt(ϵ) large with probability ≈ 1
- hwt(δ) small $\Rightarrow$ hwt(ϵ) can be small in the worst case; this happens with some probability p_1

Differential attack: wait until the 'worst case' happens: when a pair of texts is encrypted that, under the used key, has only a small amplification of the difference (= bad diffusion).

Differential Cryptanalysis

Attack strategy:
- Collect a pair of plaintexts with small difference
- If the ciphertexts show a large difference, go to the next pair
- If the ciphertexts have a small difference, assume that the difference of the intermediate values was also small
- Using the known ciphertexts and plaintexts, derive information about the key
- Repeat until the error probability is small enough

Optimisation: consider chosen plaintexts (only differences have to be chosen)

Linear and Differential trails

Linear/affine components: δ determines ϵ
Example:

$$\begin{cases} y_1 = x_1 + x_3 \\ y_2 = x_2 + x_3 \end{cases}$$

$$\delta = (0,0,1) \Rightarrow \epsilon = (1,1); \delta = (1,1,1) \Rightarrow \epsilon = (0,0)$$

The pattern, or *trail* is fixed.

Probability, correlation = 1 or 0

Linear and Differential trails (2)

Nonlinear components: several (δ, ϵ) tuples
Example:

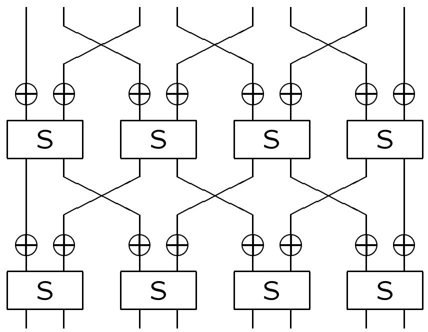
$$\begin{cases} y_1 = x_1 x_3 \\ y_2 = x_2 x_3 \end{cases}$$

$$\delta = (0,0,1) \Rightarrow \epsilon = (1,1),(0,1),(1,0),(0,0)$$

Probability, correlation < 1

Freedom in the *trail*

An Example



Nonlinear components: *active* or *passive*

Design Strategies

Is a design strategy necessary ?

L. O'Connor:
"Most ciphers are secure after sufficiently many rounds."
J.L. Massey:
"Most ciphers are too slow after sufficiently many rounds."

'Random' ciphers:

- less likely to contain unexpected weaknesses (?)
- may hide trapdoors
- more difficult to assess the security
- usually slower than 'designed' ciphers

Design Goals

Make the cipher resistant against linear and differential cryptanalysis, *and variants*.

Provable security ?

Get a high encryption/decryption performance

On many platforms ?

Bounds for Linear and Differential Cryptanalysis

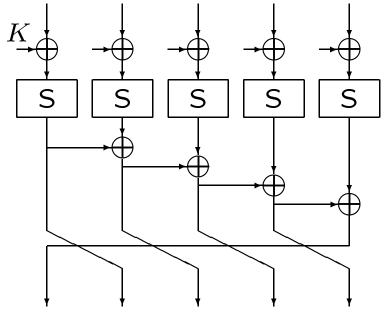
1. Assumption of independent rounds

2. Assumption of key averaging

$$
\begin{aligned}
P_{\text{cipher}} &= \prod_{r=1}^{R} P_{\text{round } r} \\
P_{\text{round}} &= \prod_{c} P_{\text{component } c} \\
&= \prod_{c} P_{\text{active non-linear component } c}
\end{aligned}
$$

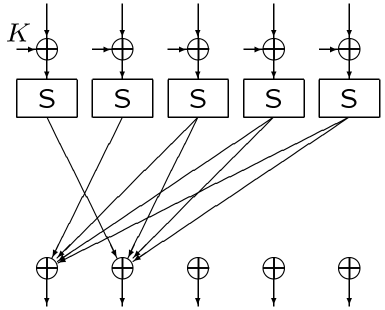
→ Design goal: low upper bounds
→ Lower bound # of active components

A Naive Solution (SEAL)



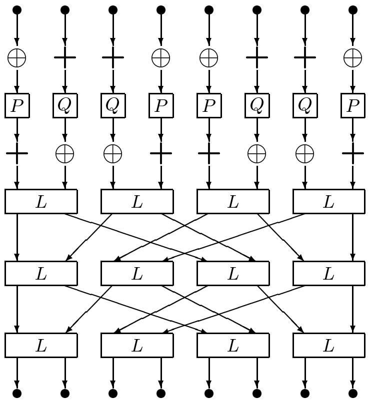
some outputs correlate with a small set of inputs

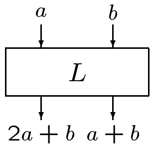
Another Naive Solution (Stafford-Tavares)



the smallest input changes have wide propagation, but there are still *narrow trails*

SAFER: PHT Transform





IDEA (and Others)

Idea: introduce the key in a more complex way:

$$X \to Y = K \cdot X$$

Properties:

1. K unknown $\Rightarrow \epsilon$ not predictable from δ
2. Very good diffusion, when *averaged* over the values for K

BUT:

1. In practice: 1 fixed key is attacked: relations between successive pairs (higher order differential cryptanalysis)
2. 'weak keys': 0, 1, ...

The Wide Trail Design Strategy

The round function of a block cipher consists of different components, with different functionality:

1. Non-linear substitution: low probability when active
2. Linear mixing transformations: make many non-linear components active
3. Key addition

Components can be selected and tested separately

Keep it as simple as possible

Linear transformation: Branch Number

Observation: *Cryptanalysts always exploit the worst case*

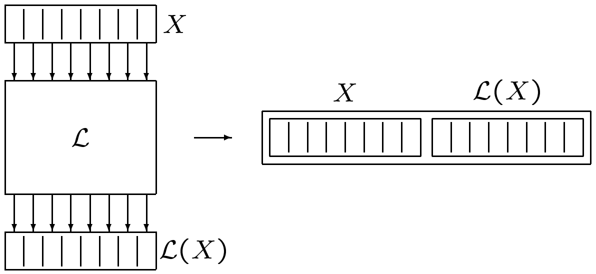
Worst case difference propagation
Linear layer $\mathcal{L}$

$$\begin{aligned} \mathcal{B}(\mathcal{L}) &= \min\{wt(X_2 - X_1) + wt(\mathcal{L}(X_2) - \mathcal{L}(X_1))\} \\ &= \min\{wt(X) + wt(\mathcal{L}(X))\} \end{aligned}$$

Upper bound: $\mathcal{B} \leq 1 + n$
$n = \#$blocks

Optimal Diffusion

Idea: linear codes: Every mapping defines a linear code



Linear code with codewords $(X, \mathcal{L}(X))$
dimension n
length $2n$

Separable codes with these dimensions, define a mapping

Optimal Diffusion (2)

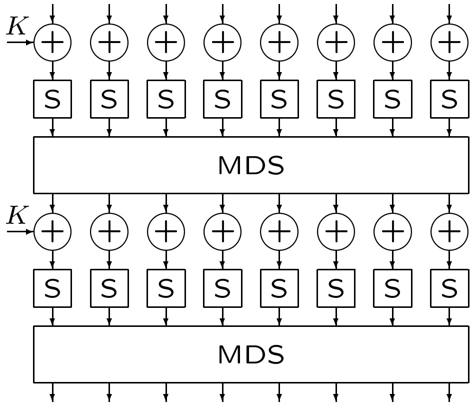
Linear code with codewords $(X, \mathcal{L}(X))$
dimension n
length $2n$

The singleton bound:

$$d <= 2n - n + 1 = n + 1$$

'Optimal codes':
Maximal Distance Separable codes

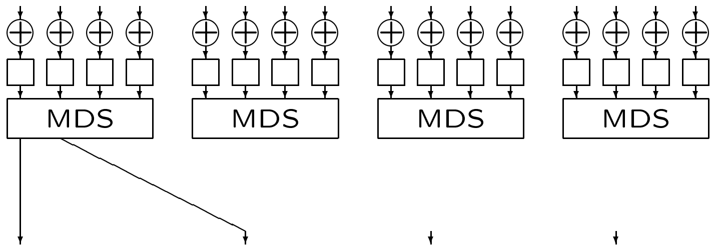
Shark



Number of 'activated S-boxes' per 2 rounds $\geq \mathcal{B} = 9$

Problem: efficient implementation

Square



Number of activated S-boxes per 4 rounds $\geq \mathcal{B}^2 = 25$

Efficient on 8-bit processors and 32-bit processors
Parallellism

AES: What is It ?

- Block cipher: plaintext *blocks* → ciphertext *blocks*
  Symmetric encryption algorithm
  Can be used in applications (e.g.: MAC)
- Dimensions:
  - Block length: 128 bits
  - Key length: 128, 192, 256 bits
- DES: 64-bit blocks, 56-bit keys
  3-DES: 112-bit keys

Who is Behind AES ?

- AES: Advanced Encryption Standard
- Initiative of the US National Institute of Standards and Technology (NIST)
  known from DES, DES modes of use, SHA-1,
  DSA, . . .
- Goal: to find a successor for DES
- `http://www.nist.gov/aes`

AES: Scope and Significance

- FIPS 197 standard published in November 2001
- Official scope is limited:
  - US Federal Administration
  - Documents that are 'sensitive but not classified'
- Significance is huge: AES is the successor of DES
- Major factors for quick acceptance:
  - No royalties
  - Low resource consumption
  - High quality

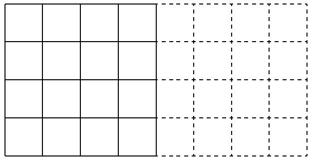
Some AES Commitments

- Standardisation: ISO, IPSec, ETSI
- Companies: RSA labs, Entrust, Pijnenburg
- Banking
- Consumer products

Rijndael

Based on Square; added variable block length
Rijndael round function `Round`:

1. `SubBytes`
2. `ShiftRows`
3. `MixColumns`
4. `AddRoundKey`



Rijndael (2)

Rijndael consists of 10/12/14 rounds:

1. `AddRoundKey`
2. 9/11/13 iterations of `Round`
3. `FinalRound`

`FinalRound` is a `Round` operation without `Mixcolumns`

Data Representation

Input: byte string

$$p_0p_1p_2p_3p_4p_5p_6p_7p_8p_9p_{10}p_{11}p_{12}p_{13}p_{14}p_{15}$$

Key: byte string

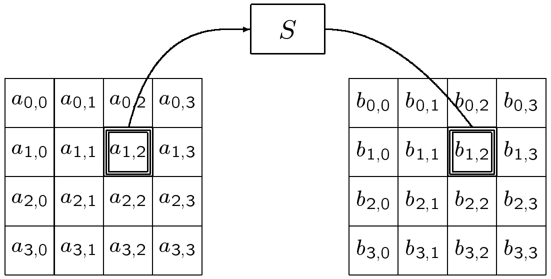
$$k_0k_1k_2k_3k_4k_5k_6k_7k_8k_9k_{10}k_{11}k_{12}k_{13}k_{14}k_{15}k_{16}k_{17}k_{18}k_{19}k_{20}k_{21}k_{22}k_{23}$$

Representation: rectangular byte arrays

p_0	p_4	p_8	p_{12}
p_1	p_5	p_9	p_{13}
p_2	p_6	p_{10}	p_{14}
p_3	p_7	p_{11}	p_{15}

k_0	k_4	k_8	k_{12}	k_{16}	k_{20}
k_1	k_5	k_9	k_{13}	k_{17}	k_{21}
k_2	k_6	k_{10}	k_{14}	k_{18}	k_{22}
k_3	k_7	k_{11}	k_{15}	k_{19}	k_{23}

Byte Substitution



Transformation is byte by byte
Invertible S-box
1 S-box for the whole cipher (simplicity)

MixColumns

$a_{0,0}$	$a_{0,1}$	$a_{0,2}$	$a_{0,3}$
$a_{1,0}$	$a_{1,1}$	$a_{1,2}$	$a_{1,3}$
$a_{2,0}$	$a_{2,1}$	$a_{2,2}$	$a_{2,3}$
$a_{3,0}$	$a_{3,1}$	$a_{3,2}$	$a_{3,3}$

$$\times \begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix}$$

$b_{0,0}$	$b_{0,1}$	$b_{0,2}$	$b_{0,3}$
$b_{1,0}$	$b_{1,1}$	$b_{1,2}$	$b_{1,3}$
$b_{2,0}$	$b_{2,1}$	$b_{2,2}$	$b_{2,3}$
$b_{3,0}$	$b_{3,1}$	$b_{3,2}$	$b_{3,3}$

Operations in Finite Field $GF(2^8)$
Bytes in columns are combined linearly
Good diffusion properties
Based on Maximal Distance codes

ShiftRows

a	b	c	d
e	f	g	h
i	j	k	l
m	n	o	p

$\longrightarrow$

a	b	c	d
f	g	h	e
k	l	i	j
p	m	n	o

Rows shifted over different offsets
Diffusion over the columns

AddRoundKey

$a_{0,0}$	$a_{0,1}$	$a_{0,2}$	$a_{0,3}$
$a_{1,0}$	$a_{1,1}$	$a_{1,2}$	$a_{1,3}$
$a_{2,0}$	$a_{2,1}$	$a_{2,2}$	$a_{2,3}$
$a_{3,0}$	$a_{3,1}$	$a_{3,2}$	$a_{3,3}$

$+$

$k_{0,0}$	$k_{0,1}$	$k_{0,2}$	$k_{0,3}$
$k_{1,0}$	$k_{1,1}$	$k_{1,2}$	$k_{1,3}$
$k_{2,0}$	$k_{2,1}$	$k_{2,2}$	$k_{2,3}$
$k_{3,0}$	$k_{3,1}$	$k_{3,2}$	$k_{3,3}$

$=$

$b_{0,0}$	$b_{0,1}$	$b_{0,2}$	$b_{0,3}$
$b_{1,0}$	$b_{1,1}$	$b_{1,2}$	$b_{1,3}$
$b_{2,0}$	$b_{2,1}$	$b_{2,2}$	$b_{2,3}$
$b_{3,0}$	$b_{3,1}$	$b_{3,2}$	$b_{3,3}$

Bit-wise XOR

Difference Propagation

SubBytes: differences remain contained

$\star$	$\star$		
	$\star$		

$\xrightarrow{\text{SubBytes}}$

$\star$	$\star$		
	$\star$		

Difference byte values: unpredictable
Difference pattern: fixed

Difference Propagation (2)

MixColumns: differences spread in the columns

$$\begin{array}{|c|c|c|c|}\hline \star & \star & & \\ \hline & & & \\ \hline & \star & & \\ \hline & & & \\ \hline \end{array} \quad \xrightarrow{\text{MixColumns}} \quad \begin{array}{|c|c|c|c|}\hline \star & \star & & \\ \hline \star & & & \\ \hline \star & \star & & \\ \hline \star & \star & & \\ \hline \end{array}$$

Difference byte values: predictable
Difference pattern: use of MDS codes guarantees 'wide trails'

Difference Propagation (3)

ShiftRows: spreading over columns

Every '$\star$' adds unpredictability
The S-box is optimal in this respect (design by K. Nyberg)

Linear components:
Theorem: in every 4 rounds, at least 25 $\star$'s

Similar theorem for linear cryptanalysis

The S-box

[Nyberg'94]: use $x \to x^{-1}$ over $GF(2^8)$

1. optimal spreading of (δ, ϵ)
2. minimal input-output bit correlations
3. maximal nonlinear order of the output
4. 'easy' description

$\to$ Add affine mapping to remediate (4)

The S-box (2)

Affine mapping

$$\begin{bmatrix} y_7 \\ y_6 \\ y_5 \\ y_4 \\ y_3 \\ y_2 \\ y_1 \\ y_0 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_7 \\ x_6 \\ x_5 \\ x_4 \\ x_3 \\ x_2 \\ x_1 \\ x_0 \end{bmatrix} \oplus \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

Implementation

Exors and small bit permutations can be implemented quite efficiently
MDS code based mappings require operations in a finite field

Efficient implementation on 32-bit CPU: combine non-linear substitution and mixing layer in 1 set of table look-ups

Non-Linear Substitution and Mixing

Substitution

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} \to \begin{bmatrix} S[a_0] \\ S[a_1] \\ S[a_2] \\ S[a_3] \end{bmatrix}$$

Mixing

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} \to \begin{bmatrix} c_{00} & c_{01} & c_{02} & c_{03} \\ c_{10} & c_{11} & c_{12} & c_{13} \\ c_{20} & c_{21} & c_{22} & c_{23} \\ c_{30} & c_{31} & c_{32} & c_{33} \end{bmatrix} \times \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

Combined Table Look-Up

$b =$ mixing (substitution(a)):

$$\begin{array}{lcccccccc} b_0 & = & c_{00} \cdot S[a_0] & + & c_{01} \cdot S[a_1] & + & c_{02} \cdot S[a_2] & + & c_{03} \cdot S[a_3] \\ b_1 & = & c_{10} \cdot S[a_0] & + & c_{11} \cdot S[a_1] & + & c_{12} \cdot S[a_2] & + & c_{13} \cdot S[a_3] \\ b_2 & = & c_{20} \cdot S[a_0] & + & c_{21} \cdot S[a_1] & + & c_{22} \cdot S[a_2] & + & c_{23} \cdot S[a_3] \\ b_3 & = & c_{30} \cdot S[a_0] & + & c_{31} \cdot S[a_1] & + & c_{32} \cdot S[a_2] & + & c_{33} \cdot S[a_3] \end{array}$$

$$T_0[x] = \begin{bmatrix} c_{00} \cdot S[x] \\ c_{10} \cdot S[x] \\ c_{20} \cdot S[x] \\ c_{30} \cdot S[x] \end{bmatrix}, T_1[x] = \begin{bmatrix} c_{01} \cdot S[x] \\ c_{11} \cdot S[x] \\ c_{21} \cdot S[x] \\ c_{31} \cdot S[x] \end{bmatrix}, \ldots$$

$$\begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} = T_0[a_0] + T_1[a_1] + T_2[a_2] + T_3[a_3]$$

Implementation of the Inverse Cipher

Combined table look-up only works when the mixing step immediately follows after the nonlinear substitution
$\Rightarrow$ steps need to be reordered in the inverse operation

8-bit Processors

Implementation on processor with small RAM:
No combined table lookup possible, matrix multiplication has to be performed explicitly every time
⇒ select matrix coefficients c_{ij} to simplify this

Rijndael Key Schedule

Different key scheduling for different key lengths (extendible for non-AES key lengths)

Fast and light, because in many applications, the key scheduling is a bottleneck: short messages, hashing, . . .

Rijndael Key Schedule (2)

k_0	k_1	k_2	k_3	k_4	k_5	k_6	k_7	k_8	k_9	k_{10}	k_{11}	k_{12}	k_{13}	k_{14}	k_{15}	$\cdots$

Round key 0	Round key 1	Round key 2	$\cdots$

$$\begin{aligned} k_{6n} &= k_{6n-6} + f(k_{6n-1}) \\ k_i &= k_{i-6} + k_{i-1} \end{aligned}$$

Cipher key expansion can be done just-in-time
No extra storage required

Rijndael: Security

- provable bounds against differential and linear cryptanalysis
- dedicated attack for up to 6-7 rounds
- simple basic operations, because we think that makes it easier to protect against implementation attacks (timing, power analysis)

Square Attack (1) [Knudsen'97]

- applied to the block cipher Square
- exploits byte-oriented structure
- aka structural attack, saturation attack, integral attack

Λ-set: set of 256 states that are

- active state bytes: all different values
- passive state bytes: all equal

summation property Σ

$$\bigoplus_{x \in \Lambda} x_{i,j} = 0, \ \forall i, j$$

Square Attack (2)

applying:

- SubBytes leaves a Λ-set invariant
- AddRoundKey leaves a Λ-set invariant
- ShiftRows moves the active bytes of a Λ-set
- MixColumns does *not necessarily* leave a Λ-set invariant

But: applying MixColumns on a column with one active byte results in a Λ-set with all 4 bytes active.
Proof: every output byte of MixColumns is a linear combination with invertible coefficients of the 4 input bytes

$$b_{i,j} = \texttt{MixColumns}(a_{i,j}) = 02 \cdot a_{i,j} \oplus 03 \cdot a_{i+1,j} \oplus a_{i+2,j} \oplus a_{i+3,j}$$

Square Attack (3): 4 rounds

Consider a Λ-set with one active byte

input round 1:

Σ	C	C	C
C	C	C	C
C	C	C	C
C	C	C	C

output round 1:

Σ	C	C	C
Σ	C	C	C
Σ	C	C	C
Σ	C	C	C

output round 2:

Σ	Σ	Σ	Σ
Σ	Σ	Σ	Σ
Σ	Σ	Σ	Σ
Σ	Σ	Σ	Σ

in round 3:
SubBytes and AddRoundKey leave the Σ property invariant

effect of MixColumns?

Square Attack (4): 4 rounds

effect of MixColumns in round 3:

$$\begin{aligned}
\bigoplus_l b^l_{i,j} &= \bigoplus_l \texttt{MixColumns}(a^l_{i,j}) \\
&= \bigoplus_l (02 \cdot a^l_{i,j} \ \oplus \ 03 \cdot a^l_{i+1,j} \ \oplus \ a^l_{i+2,j} \ \oplus \ a^l_{i+3,j}) \\
&= 02 \cdot \bigoplus_l a^l_{i,j} \ \oplus \ 03 \cdot \bigoplus_l a^l_{i+1,j} \ \oplus \ \bigoplus_l a^l_{i+2,j} \ \oplus \ \bigoplus_l a^l_{i+3,j} \\
&= 0 \oplus 0 \oplus 0 \oplus 0 = 0
\end{aligned}$$

All input bytes of round 4 sum to 0 (Σ)

Round4: $\texttt{AddRoundKey}(\texttt{ShiftRows}(\texttt{SubBytes}(C_4)), K_4)$
SubBytes in round 4 destroys the Σ property

Square Attack (5): 4 rounds

Round 4 maps single input bytes to single output bytes: one can guess K_4 (byte by byte) and check upon decryption of 1 round that the Σ property holds.

$$\bigoplus_{l} S^{-1}[C_{i,j-\delta_i} \oplus K_{i,j-\delta_i}] = 0$$

Effort: $2^9 \cdot 2^8$ XORs and $2^8 \cdot 2^8$ table lookups per key byte
Total effort for 16 key bytes: 2^{21} XORs and 2^{20} table lookups, or (roughly) 2^{10} 4-round encryptions

Second Λ-set required to eliminate false positives

Note: if round 4 would have had MixColumns, it could have been moved after AddRoundKey and subsequently removed

Square Attack (6): 5-6 rounds

round at the end: guess 4 extra key bytes 2^{40} steps and need 5 Λ-sets.

round at the beginning: let one column take on all 2^{32} 32-bit values; output of round 1 consists then of 2^{24} Λ-sets

Attack	# plaintexts	# encryptions	memory
4 rounds	2^9	2^{14}	small
4+1 rounds	2^{11}	2^{46}	small
1+4 rounds	2^{33}	2^{38}	2^{32}
1+4+1 rounds	2^{35}	2^{46}	2^{32}
1+4+2 rounds	2^{32}	2^{189}	2^{32}

last attack: Gilbert-Minier, FSE 2000, based on a 4-round distinguisher

Algebraic Properties of AES

AES is completely defined over $GF(2^8)$, except for the affine transformation of the S-box (defined over $GF(2)$).

AES S-box: $z = S(x) = f(g(x)) = f(y)$ with
$f(y)$ an affine mapping over $GF(2)$
$y = g(x) = x^{-1}$, or $x \cdot y = 1$ if $x \neq 0$

With defining polynomial $GF(2^8)$: $X^8 + X^4 + X^3 + X + 1$ (not essential for properties)

Then $f(y) = 63 + 05y + 09y^2 + \mathtt{f9}y^4 + 25y^8 + \mathtt{f4}y^{16} + 01y^{32} + \mathtt{b5}y^{64} + \mathtt{8f}y^{128}$

Algebraic Properties of AES (2)

[Courtois-Goubin-Patarin'0x] The AES S-box has 8 bi-affine equations between the coordinates x_i and z_i (1 of these holds with probability 255/256 only):

$$0 = z_1x_0 + z_4x_0 + z_6x_0 + z_0x_1 + z_2x_2 + z_4x_2 + z_7x_2 + z_1x_3 + z_3x_3 + z_6x_3 + z_0x_4 + z_1x_4 + z_2x_4 + z_4x_4 + z_5x_4 + z_6x_4 + z_0x_5 + z_3x_5 + z_5x_5 + z_6x_5 + z_7x_5 + z_2x_6 + z_4x_6 + z_5x_6 + z_7x_6 + z_3x_7 + z_5x_7 + z_3x_1 + {} + z_5x_1 + x_5 + x_7$$

We also have: $x = x^2 \cdot y$; results in 8 additional equations

Similarly: $y = y^2 \cdot x$; results in 8 additional equations

In total, there are 23 bi-affine quadratic equations that hold with probability 1.

There are also 39 fully quadratic equations (with terms x_ix_j and z_iz_j).

Algebraic Properties of AES (3)

AES (128-bit block length) with a 128-bit key: 8000 quadratic equations with 1600 variables (strongly overdetermined)

Can we solve these? (MQ problem)

- NP hard if not overdetermined
- May be easier if overdetermined: Shamir-Kipnis, Crypto'99 (relinearization) and Shamir-Patarin-Courtois-Klimov, Eurocrypt '00 (XL)
- Even easier if structured/sparse: Courtois-Pieprzyk, Asiacrypt'02 (XSL)

Algebraic Properties of AES (4)

[Murphy-Robshaw, Crypto'02]: AES can be embedded in a $8 \cdot 128$-bit block cipher, which is completely defined over $GF(2^8)$. This may further simplify the equations.

Does there exist any method to recover the key faster than by exhaustive search?

The jury is still out!

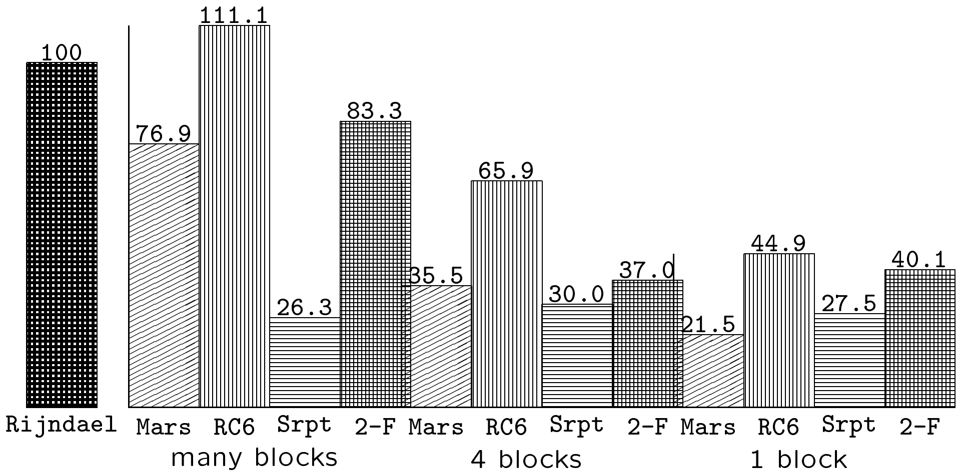
Performance-Security Trade-off

(Serpent designers:) Security weighted performance

$\rightarrow$ Performance-weighted Security

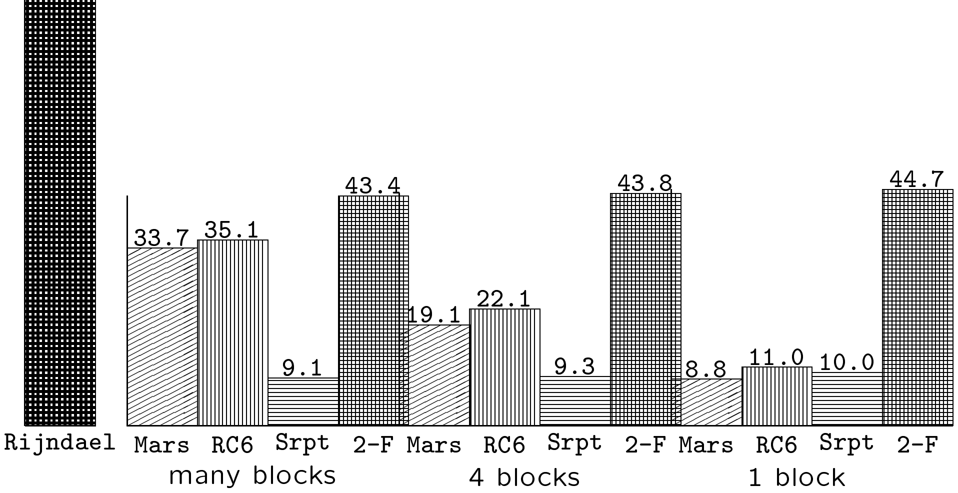
Performance of the Finalists

% executed when Rijndael finishes (Pentium II/Pro)



Performance of the Finalists

% executed when Rijndael finishes (Motorola 6805)



Concluding Remarks

- Design theory for block ciphers is quite recent ($\leftrightarrow$ stream ciphers)
- We know how to construct block ciphers that nobody today can break
- Good performance on many platforms: difficult

Triple-DES

$E_K(\cdot)$ single encryption, $|K| = 2^k$

- Double encryption: $C = E_{K2}(E_{K1}(P))$
- Triple encryption: $C = E_{K3}(E_{K2}(E_{K1}(P)))$
- Both at least as secure as E
- Two-key Triple Encryption: $C = E_{K1}(D_{K2}(E_{K1}(P)))$

Triple-DES: Best Known Attacks

	Time	Plaintexts	Memory	Full cost
Double	2^{56}	2	2^{56}	$2^{6k/5}$
Double	2^{72}	2	2^{30}	
2K3	2^{56}	2^{56}	2^{56}	$2^{4k/3}(**)$
2K3	$2^{120-m}(*)$	2^m	2^m	
3K3	2^{112}	3	2^{56}	$2^{11k/5}$
3K3	2^{104}	2^{32}	2^{72}(?)	

(*) $\min(2^{112}, 2^{120-m})$

(**) needs huge number of chosen plaintexts (2^k)

+ matching ciphertext attack with 2^{32} blocks

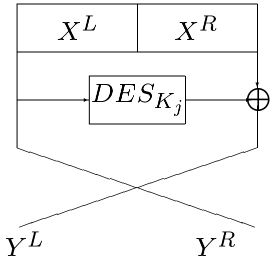
DES-X

Exor plaintext and ciphertext with 64-bit key.

Security level: 2^{119-t}, for 2^t chosen/known plaintexts

Best known attack: known plaintexts [Biryukov-Wagner]

DEAL (Knudsen 97)



- 6 rounds, choice of 128-bit, 192-bit and 256-bit keys
- As fast as triple-DES
- Matching ciphertext attack requires 2^{64} ciphertext blocks
- Best known attack runs in time 2^{121}

Triple DES - Status

- Slow in software
- No efficient shortcut attacks
- Best known attack on 2K-3DES: 2^t known texts, 2^{120-t} time
- Best known attack on 3K-3DES: 2^{32} known texts, 2^{104} encryptions [Lucks00] (high full cost; chosen plaintexts and ciphertexts)
- Blocks "only" 64 bits
- Alternative: DEAL
- 3K-3DES vulnerable to related key attacks

Extensions & Improvements

- find good characteristics and linear approximation trails (or differentials and linear approximations) problem for designer and cryptanalyst!
- DC
  - multiple characteristics
  - key dependent characteristics
  - higher order differentials
  - partial differentials
  - differential attacks when only part of the output is known (DES in CFB mode and with CBC-MAC) – maximum likelihood
  - hash functions based on block ciphers

Extensions & Improvements (2)

- LC
  - multiple approximations
  - more information from data
  - ciphertext only
  - chosen plaintext
- linear-differential [Langford-Hellman'94]:
  - chosen plaintext
  - 8-round DES - doesn't seem to extend easily
  - 768 chosen plaintexts
    (compared to 5 000 previously)
- impossible differentials [Biham-Shamir-Biryukov]
- interpolation attack [Jakobsen-Knudsen]
- integral/square attack [Daemen-Rijmen-Knudsen]
- non-surjective attack [De Win-Preneel-Rijmen]

Related Key Attacks

Attacker exploits relation between

$E_K(\cdot), E_{K+a_1}(\cdot), \ldots, E_{K+a_n}(\cdot)$

and finds K, where $a_1, \ldots, a_n$ are known

Variants: Attacker gets

1) encryptions $E_K(\cdot)$
2) encryptions $E_K(\cdot), E_{K+a_1}(\cdot), \ldots, E_{K+a_n}(\cdot)$
3) variant: 2) where $a_1, \ldots, a_n$ are unknown

Differentials

In basic DC intermediate differences not used

- Charact. $\Phi = (\Delta P, \Delta C_1, \ldots \Delta C_{r-2}, \Delta C_{r-1})$
- Differential $\Omega = (\Delta P, \Delta C_{r-1})$
- $\Pr(\Omega) \geq \Pr(\Phi)$
- $\Pr(\Omega)$ probability attacker can expect in attack

Prob. of characteristic (Markov ciphers)

$$\Pr(\Delta C_s = \alpha_s, \ldots\ldots, \Delta C_1 = \alpha_1 | \Delta P = \alpha_0) = \prod_{i=1}^{s} \Pr(\Delta C_1 = \alpha_i | \Delta P = \alpha_{i-1})$$

Prob. of differential (Markov ciphers)

$$\Pr(\Delta C_s = \beta_s \mid \Delta P = \beta_0) = \sum_{\beta_1} \cdots \sum_{\beta_{s-1}} \prod_{i=1}^{s} \Pr(\Delta C_i = \beta_i \mid \Delta C_{i-1} = \beta_{i-1})$$

where $\Delta C_0 = \Delta P$

Differentials and Probabilities

Probability of differentials taken over all plaintexts and all keys

For Markov cipher only over all keys

Probability is an average over all keys

In attack, one key is used. Probability?

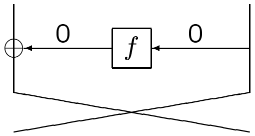
Hypothesis of stochastic equivalence
For virtually all high probability $(r-1)$-round differentials (α, β)

$\Pr_P(\Delta C_1 = \beta \mid \Delta P = \alpha,\ K = k) \approx$
$\Pr_{P,K}(\Delta C_1 = \beta \mid \Delta P = \alpha)$

holds for substantial fraction of key values k

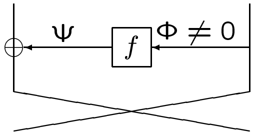
Proven Average Resistance Against DC

A trivial one-round differential with $p = 1$



A non-trivial one-round differential with

$$0 < p < 1$$



Proven Average Resistance Against DC

[Knudsen-Nyberg'92]

Let p_{max} denote the probability of the most likely non-trivial one-round differential

Theorem

Consider an r-round Feistel cipher with independent round keys. Any s-round differential, $s \geq 4$, has a probability of at most $2p_{max}^2$

Proven Average Resistance Against DC (2)

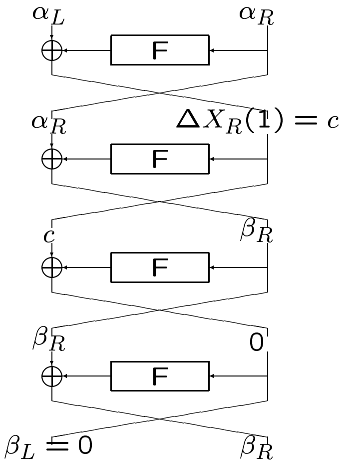
Proof. First note that if Theorem holds for $s = 4$, then it holds for $s \geq 4$

$$\begin{aligned}
&\Pr(\Delta X(s) = \beta \,|\, \Delta X(0) = \alpha) = \\
&\sum_{\Delta X(s-4)} \Pr(\Delta X(s-4) \,|\, \Delta X(0) = \alpha) \times \\
&\Pr(\Delta X(s) = \beta \,|\, \Delta X(0) = \alpha, \Delta X(s-4)) \leq \\
&\sum_{\Delta X(s-4)} \Pr(\Delta X(s-4) \,|\, \Delta X(0) = \alpha) \times 2p_{max}^2 \\
&= 2p_{max}^2
\end{aligned}$$

Proof ($\beta_L = 0$)

$$\begin{aligned}
&\mathrm{P}(\Delta X(4) = \beta \,|\, \Delta X(0) = \alpha) \\
&= \sum_{\Delta X_R(1)} \mathrm{P}(\Delta X_R(1) \,|\, \Delta X(0) = \alpha) \\
&\quad \times \mathrm{P}(\Delta X(4) = \beta \,|\, \Delta X(0) = \alpha, \Delta X_R(1)) \\
&= \mathrm{P}(\Delta X_R(1) = 0 \,|\, \Delta X(0) = \alpha) \\
&\quad \times \mathrm{P}(\Delta X(4) = \beta \,|\, \Delta X(0) = \alpha, \Delta X_R(1) = 0) \\
&\quad + \sum_{\Delta X_R(1) \neq 0} \mathrm{P}(\Delta X_R(1) \,|\, \Delta X(0) = \alpha) \\
&\quad \times \mathrm{P}(\Delta X(4) = \beta \,|\, \Delta X(0) = \alpha, \Delta X_R(1)) \\
&\leq p_{max}^2 + \\
&\quad \sum_{\Delta X_R(1) \neq 0} \mathrm{P}(\Delta X_R(1) \,|\, \Delta X(0) = \alpha) \times p_{max}^2 \\
&\leq 2p_{max}^2
\end{aligned}$$

Proof (picture)



Proven Average Resistance Against LC

Let $|p_f - 1/2|$ be the highest bias of a non-trivial linear characteristic for the function f in a Feistel cipher

Let $q_f = (2p_f - 1)^2$

Theorem [Nyberg-Knudsen-Matsui]

Consider an r-round Feistel cipher with independent round keys.
Any s-round linear characteristic, $s \geq 4$, has a probability p_L, such that

$$|p_L - 1/2|^2 \leq 8|p_f - 1/2|^4$$

or equivalently,

$$q_L \leq 2 \times q_f^2$$

Comparison DC-LC

round characteristics $p(a_i, b_i)$
round approximations $c(a_i, b_i)$

characteristic ψ: $p_\psi = \prod_{i=1}^{r} p(a_i, b_i)$
linear trail λ: $c_\lambda = \prod_{i=1}^{r} c(a_i, b_i)$

differential $\chi(a, b)$: $p_{\chi(a,b)} = \sum p_\psi$
linear approximation $\Lambda(a, b)$: $c^2_{\Lambda(a,b)} = \sum c^2_\lambda$

if round keys chosen independently
and uniformly at random
$E_{\mathcal{K}}[\Pr_X\{\Delta X(r-1) = b \mid \Delta X(0) = a\}] = p_{\chi(a,b)}$
$E_{\mathcal{K}}[| \Pr_X\{b \cdot X(r-1) = a \cdot X(0)\} - \frac{1}{2} |^2] = c^2_{\Lambda(a,b)}$
(if plaintext uniform)

proven average resistance
$\delta_f(a, b)$ small, $\forall a (a \neq 0)$, b
$c_f(a, b)$ small, $\forall a$, $b (b \neq 0)$

Exercises

1. Prove the DES complementation property:
   $\forall\, P, K : C = \text{DES}_K(P) \Longleftrightarrow \overline{C} = \text{DES}_{\overline{K}}(\overline{P})$
2. Show that the all-zero key is a weak DES key:
   $\text{DES}_K(\text{DES}_K(P)) = P$, $\forall\, P$
3. Compute the probability that a random permutation has one (or more) fixed points. What are the implications on block ciphers?
4. Show that for a DES weak key, there are at least 2^{32} fixed points.
5. Find an attack on double-DES.
6. Find the first attack on two-key triple DES (hint: consider all plaintexts for which the value after the 1st encryption equals 0).