Differential Cryptanalysis (DES versus AES)

Lorenzo Grassi

Eindhoven University of Technology, NL

December 2025

Outline

Symmetric-Key Ciphers

Differential Cryptanalysis

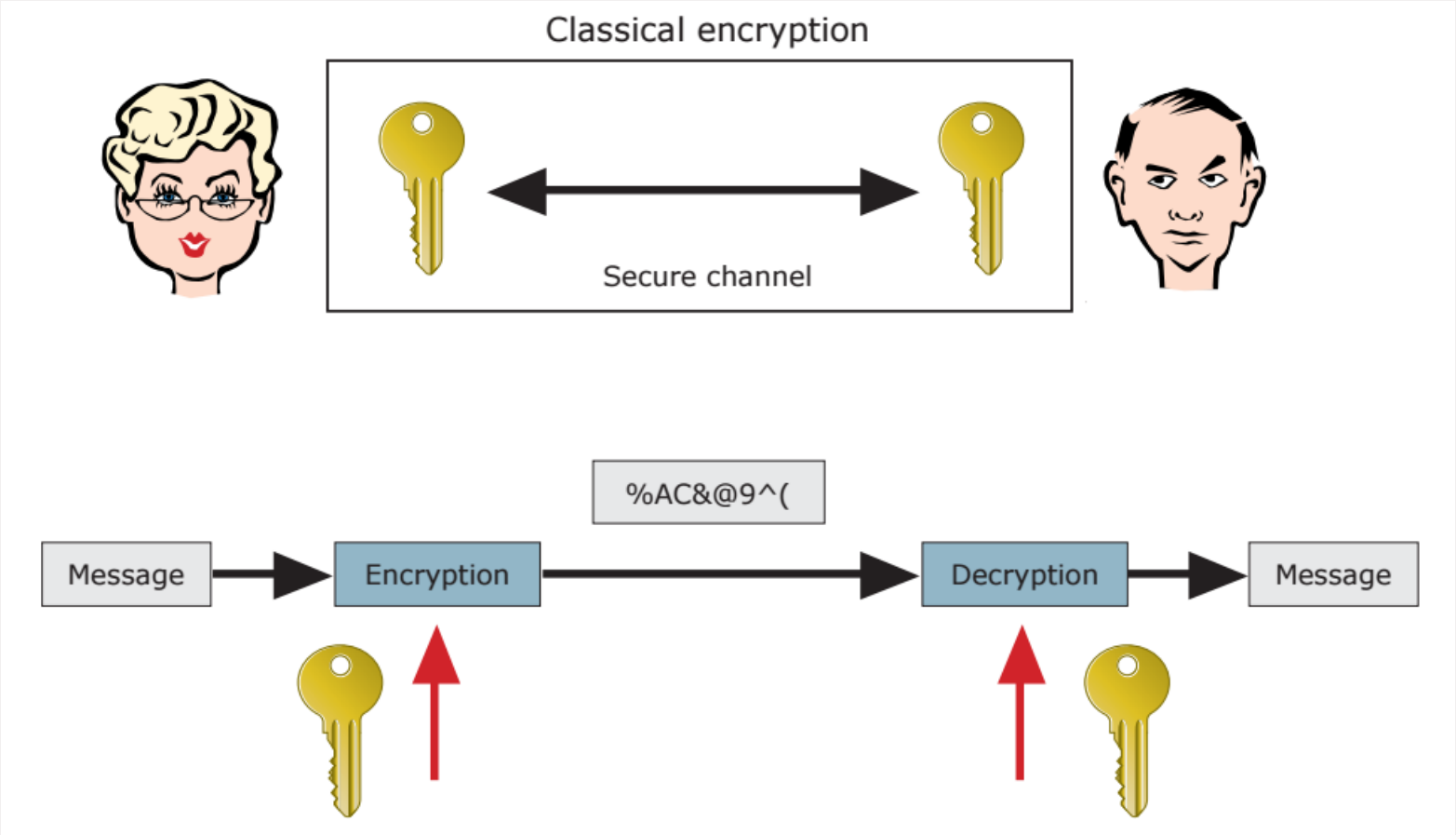
Attacks on DES

AES Block Cipher

Conclusion

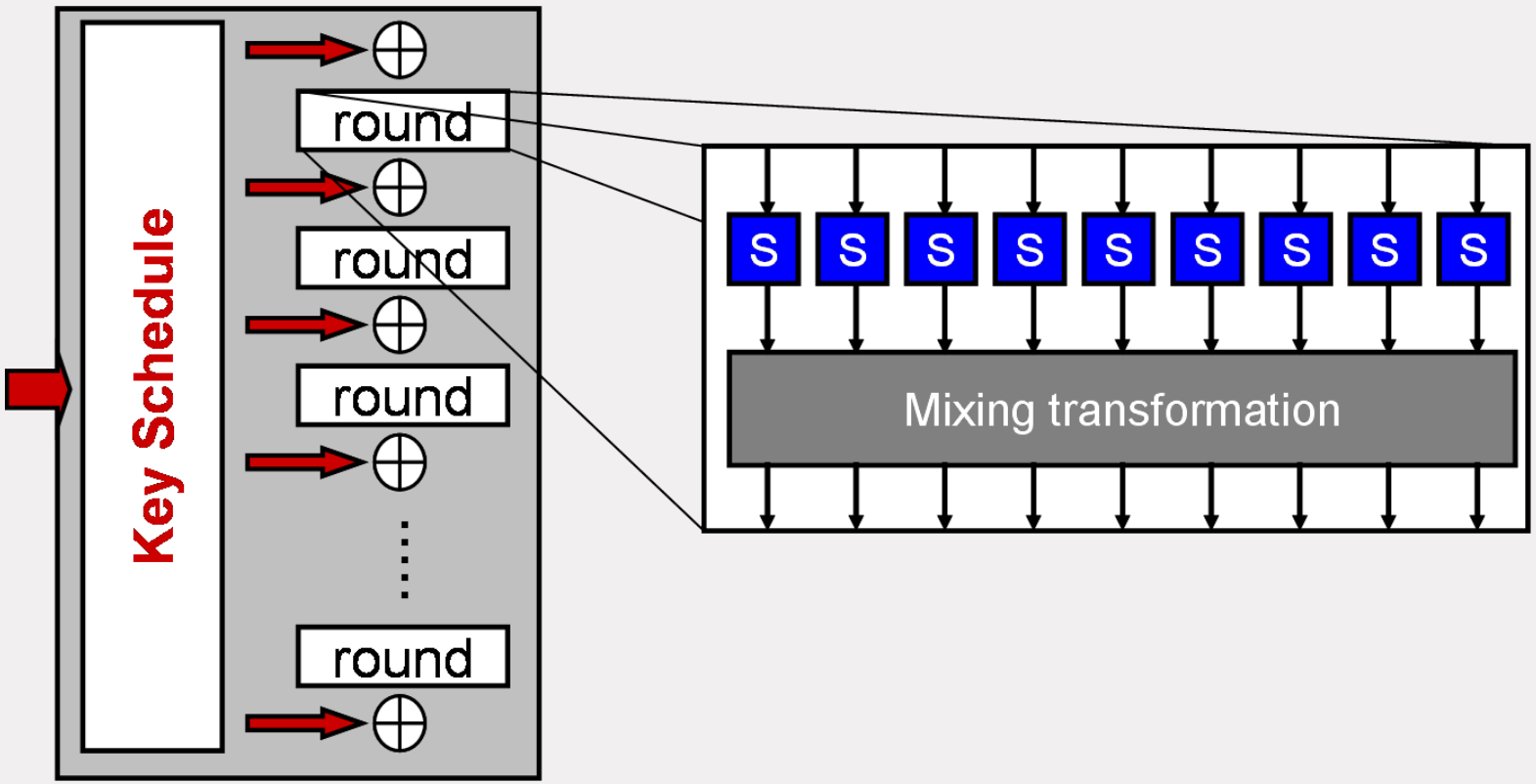
Symmetric-Key Ciphers

Symmetric-Key Cryptography



Reprinted from https://www.cosic.esat.kuleuven.be/summer_school_sardinia_2015/slides/LRKnudsen.pdf

SPN Cipher



TU/e

Security of Ciphers (and Symmetric Primitives)

When is a cipher secure?

Kerckhoffs' Principle: the security of a cryptosystem must lie in the choice of its keys only. *Everything else (including the algorithm itself) should be considered public knowledge.*

A symmetric primitive is secure if there is no attack better than brute force: a solid symmetric primitive must resist all known attacks published in the literature!

Security of Ciphers (and Symmetric Primitives)

When is a cipher secure?

Kerckhoffs' Principle: the security of a cryptosystem must lie in the choice of its keys only. *Everything else (including the algorithm itself) should be considered public knowledge.*

A symmetric primitive is secure if there is no attack better than brute force: a solid symmetric primitive must resist all known attacks published in the literature!

Security of Ciphers (and Symmetric Primitives)

When is a cipher secure?

Kerckhoffs' Principle: the security of a cryptosystem must lie in the choice of its keys only. *Everything else (including the algorithm itself) should be considered public knowledge.*

A symmetric primitive is secure if there is no attack better than brute force: a solid symmetric primitive must resist all known attacks published in the literature!

About Brute Force Attack (1/2)

Given a plaintext and the corresponding ciphertext, *brute-force attack* involves systematically checking all possible key combinations until the correct key is found.

Key Size	*Possible Combinations*
1 bit	2
2 bit	4
4 bit	16
8 bit	256
16 bit	65536
32 bit	$4.2 \cdot 10^9$
56 bit (DES)	$7.2 \cdot 10^{16}$
64 bit	$1.8 \cdot 10^{19}$
128 bit (AES-128)	$3.4 \cdot 10^{38}$
192 bit (AES-192)	$6.2 \cdot 10^{57}$
256 bit (AES-256)	$1.1 \cdot 10^{77}$

About Brute Force Attack (2/2)

Supercomputer (Sunway TaihuLight): 93.01 PetaFLOPS [Flops = Floating point operations per second]

Assume only 1 "floating point operation" is required per combination check (very optimistic!)

Number of combination checks per second: $93.01 \cdot 10^{15}$

Key Size	*Time to Crack*
56 bit (DES)	< 1 sec
128 bit (AES-128)	$1.2 \cdot 10^{14}$ years
192 bit (AES-192)	$2.1 \cdot 10^{33}$ years
256 bit (AES-256)	$3.8 \cdot 10^{52}$ years

For comparison, our universe was born about $1.37 \cdot 10^{10}$ years ago.

Differential Cryptanalysis

Differential Cryptanalysis: Idea

First published by Biham and Shamir to attack DES (around 1993):

> *Deduce information about the secret key by tracing differences between pairs of plaintexts during the encryption (and decryption).*

Consider $c = p \oplus k$. Impossibility to recover information if the key k is chosen at random and used only once.

What happens if we use the key twice? Assume we use k to encrypt p_1 and p_2:

$$c_1 \oplus c_2 = p_1 \oplus k \oplus p_2 \oplus k = p_1 \oplus p_2,$$

i.e., the attacker recovers the XOR (= difference) of the two plaintexts messages.

Differential Cryptanalysis: Idea

First published by Biham and Shamir to attack DES (around 1993):

> *Deduce information about the secret key by tracing differences between pairs of plaintexts during the encryption (and decryption).*

Consider $c = p \oplus k$. Impossibility to recover information if the key k is chosen at random and used only once.

What happens if we use the key twice? Assume we use k to encrypt p_1 and p_2:

$$c_1 \oplus c_2 = p_1 \oplus k \oplus p_2 \oplus k = p_1 \oplus p_2,$$

i.e., the attacker recovers the XOR (= difference) of the two plaintexts messages.

Differential Cryptanalysis: Idea

First published by Biham and Shamir to attack DES (around 1993):

> *Deduce information about the secret key by tracing differences between pairs of plaintexts during the encryption (and decryption).*

Consider $c = p \oplus k$. Impossibility to recover information if the key k is chosen at random and used only once.

What happens if we use the key twice? Assume we use k to encrypt p_1 and p_2:

$$c_1 \oplus c_2 = p_1 \oplus k \oplus p_2 \oplus k = p_1 \oplus p_2,$$

i.e., the attacker recovers the XOR (= difference) of the two plaintexts messages.

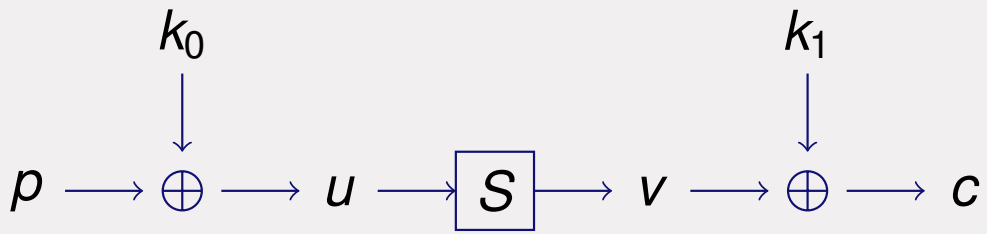
A Toy Block Cipher

Block cipher $E_{k_0 \| k_1} : \mathbb{F}_2^4 \to \mathbb{F}_2^4$ defined as

$$c = E_{k_0 \| k_1}(p) = S(p \oplus k_0) \oplus k_1 .$$

Let S be the 4-bit S-box defined as follows:

x	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
$S(x)$	6	4	c	5	0	7	2	e	1	f	3	d	8	a	9	b

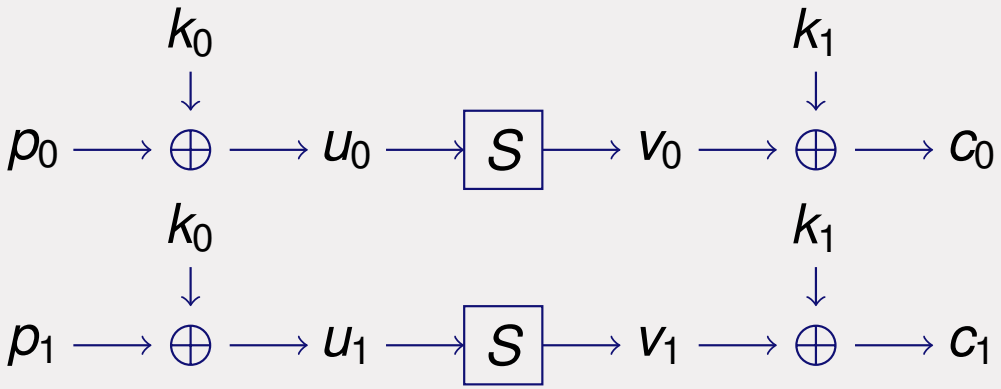


Given $(p_0, c_0) = (\mathtt{a}, 9)$ and $(p_1, c_1) = (5, 6)$, determine the key!?

Brute force (exhaustive search): try all $2^4 \cdot 2^4 = 256$ keys.

Starting Observation

Given two plaintext/ciphertext pairs (p_0, c_0) and (p_1, c_1):

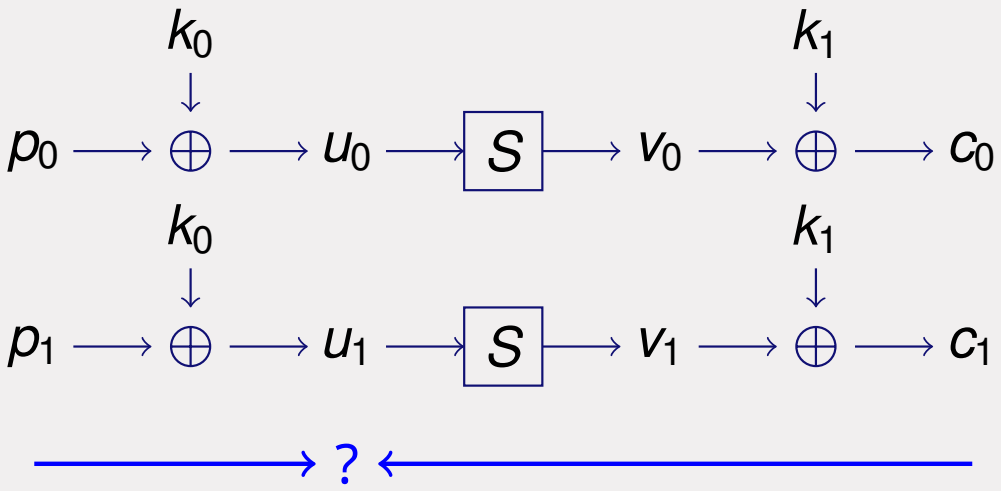


we know that

$$u_0 \oplus u_1 = (p_0 \oplus k_0) \oplus (p_1 \oplus k_0) = p_0 \oplus p_1$$
$$v_0 \oplus v_1 = (c_0 \oplus k_1) \oplus (c_1 \oplus k_1) = c_0 \oplus c_1,$$

even though we do not know k_0 and k_1.

Set up the Differential Attack

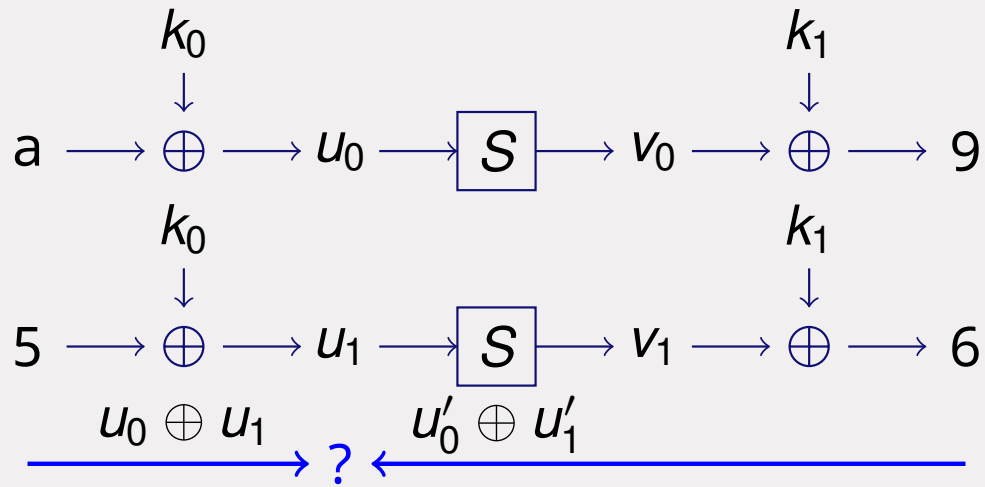


Strategy

1. compute $u_0 \oplus u_1$
2. guess k_1 (iterate over all values)
3. compute $u_0' = S^{-1}(c_0 \oplus k_1')$ and $u_1' = S^{-1}(c_1 \oplus k_1')$
4. check if $u_0 \oplus u_1 = u_0' \oplus u_1'$
5. if not: key guess was definitely wrong! (filtering)

Example

Given $(p_0, c_0) = (\mathrm{a}, 9)$ and $(p_1, c_1) = (5, 6)$

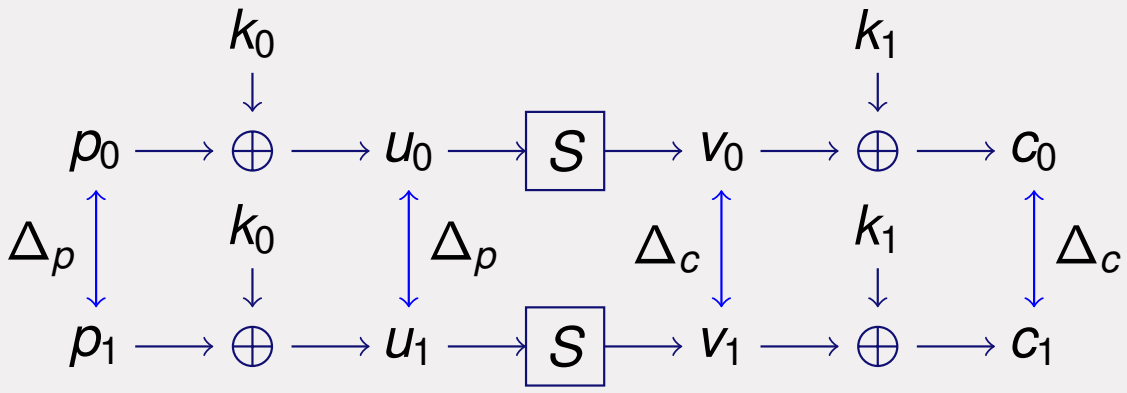


- compute $u_0 \oplus u_1 = p_0 \oplus p_1 = a \oplus 5 = f$
- guess k_1 and compute $u'_0 \oplus u'_1$:

k'_1	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
$u'_0 \oplus u'_1$	e	b	e	e	d	8	d	f	f	d	8	d	e	e	b	e

- only two candidates for k_1 are valid: $k_1 \in \{7, 8\}$

Differential Cryptanalysis



What happened?

- we can get information about the differences, even though we do not know the values;
- we can make a guess for the (last) key and verify it by computing backwards.

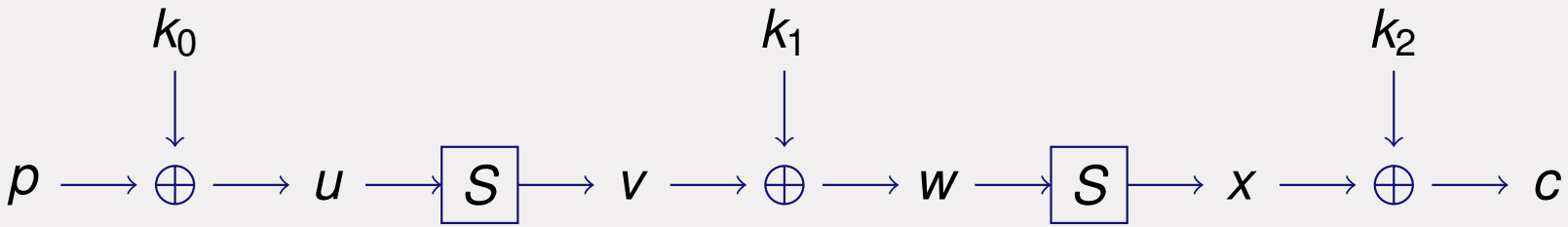
Extend it to a Two-Round Cipher

The block cipher $E_{k_0||k_1||k_2}(p)$ encrypts 4 bits of plaintext using three 4-bit keys:

$$c = E_{k_0||k_1||k_2}(p) = S(S(p \oplus k_0) \oplus k_1) \oplus k_2$$

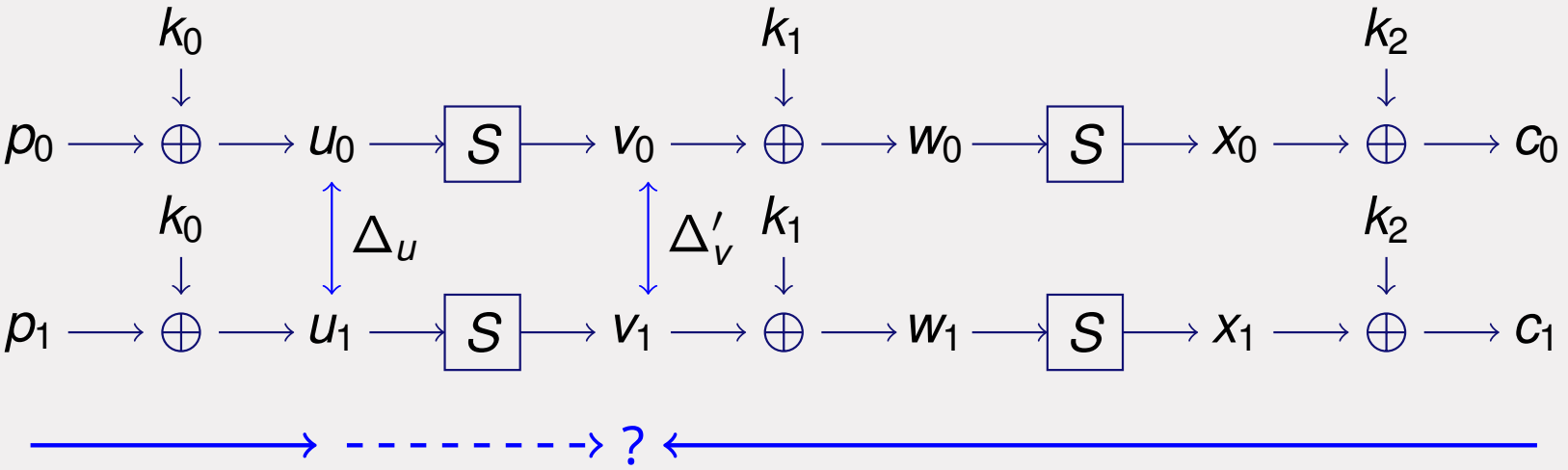
We use the same 4-bit S-box S:

x	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
$S(x)$	6	4	c	5	0	7	2	e	1	f	3	d	8	a	9	b



Brute force: $2^{4+4+4} = 4096$ keys.

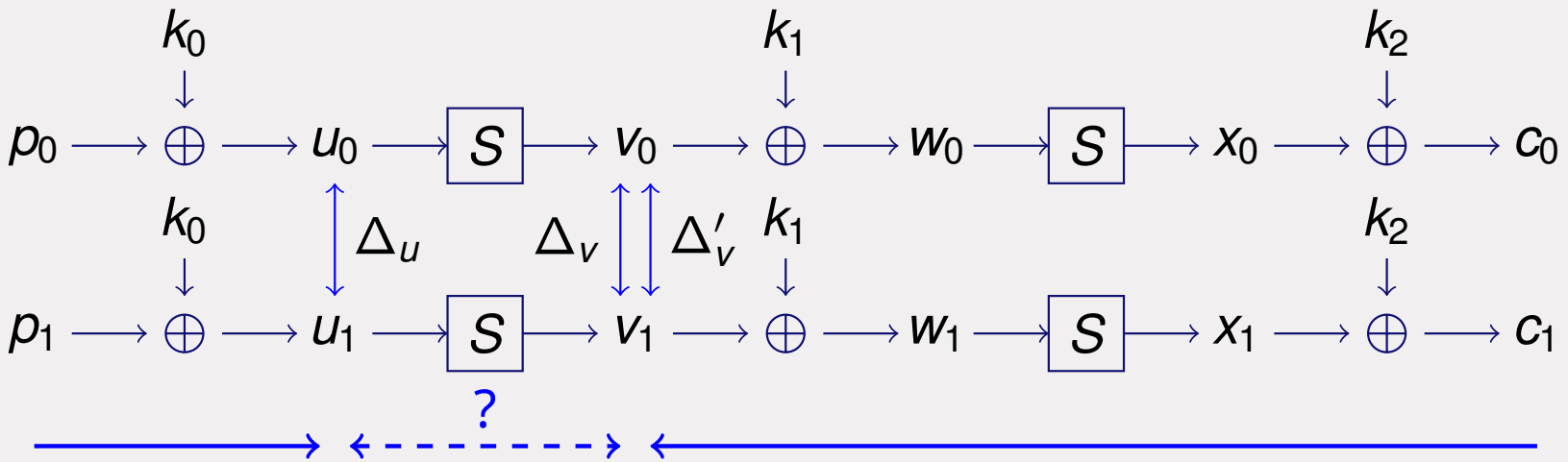
Differential Attack on a Two-Round Cipher



What can we compute?

- compute $\Delta u = \Delta p = p_0 \oplus p_1$;
- we guess the last round key k_2;
- compute x'_0, x'_1 and w'_0, w'_1;
- compute $\Delta v' = \Delta w' = w'_0 \oplus w'_1$;
- but: how can we check if our guess for k_2 was correct?

Differential Attack on a Two-Round Cipher



We want to verify our computed $\Delta v'$ (based on guess k_2'):

- can we get more information about the real Δv?
- using e.g. $\Delta u = \mathtt{f}$ but not knowing u_0 or u_1?

$\Rightarrow$ let's compute all possible differences for Δv.

TU/e

The Influence of the S-Box (1/2)

u_0	$u_1 = u_0 \oplus f$	$v_0 = S(u_0)$	$v_1 = S(u_1)$	$\Delta v = v_0 \oplus v_1$
0	f	6	b	d
1	e	4	9	d
2	d	c	a	6
3	c	5	8	d
4	b	0	d	d
5	a	7	3	4
6	9	2	f	d
7	8	e	1	f
8	7	1	e	f
9	6	f	2	d
a	5	3	7	4
b	4	d	0	d
c	3	8	5	d
d	2	a	c	6
e	1	9	4	d
f	0	b	6	d

Only 4 differences for Δv are possible (for the given $\Delta u = \mathsf{f}$).

One difference ($\Delta v = \mathsf{d}$) occurs very often.

TU/e

The Influence of the S-Box (2/2)

Observations

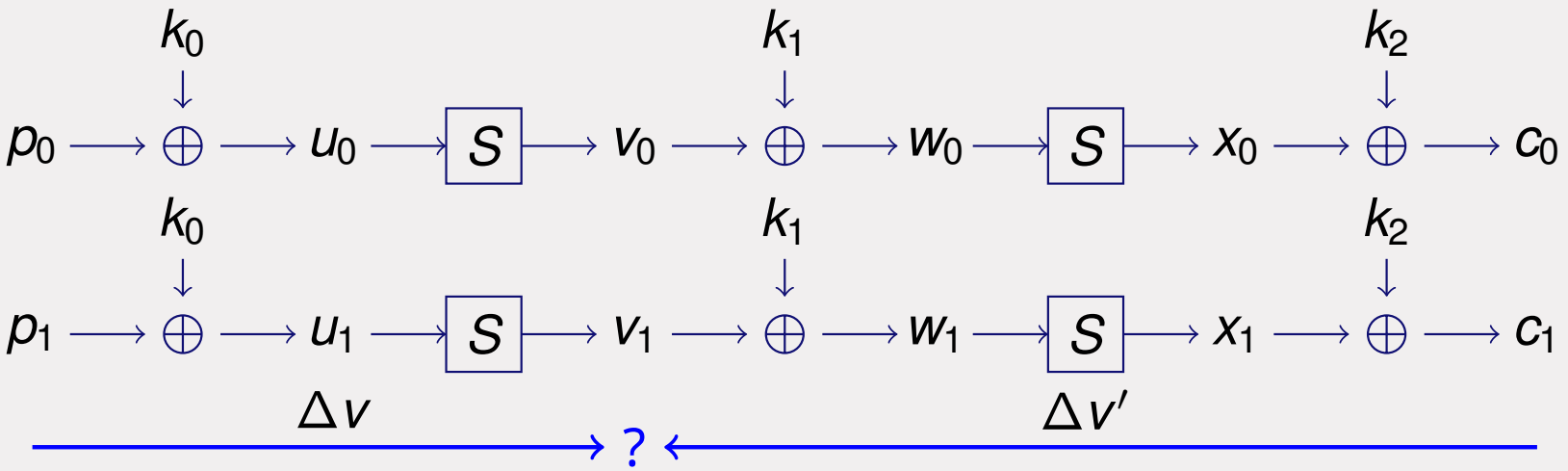
- the differences are unevenly distributed;
- the difference d occurs 10 out of 16 times;
- *not* all differences occur.

Assume that $\Delta v = \mathsf{d}$:

- we can verify our guess k_2' by checking whether $\Delta v'(= \Delta v) = \mathsf{d}$;
- our assumption is right with probability of 10/16.

$\Delta v = v_0 \oplus v_1$
d
d
6
d
d
4
d
f
f
d
4
d
d
6
d
d

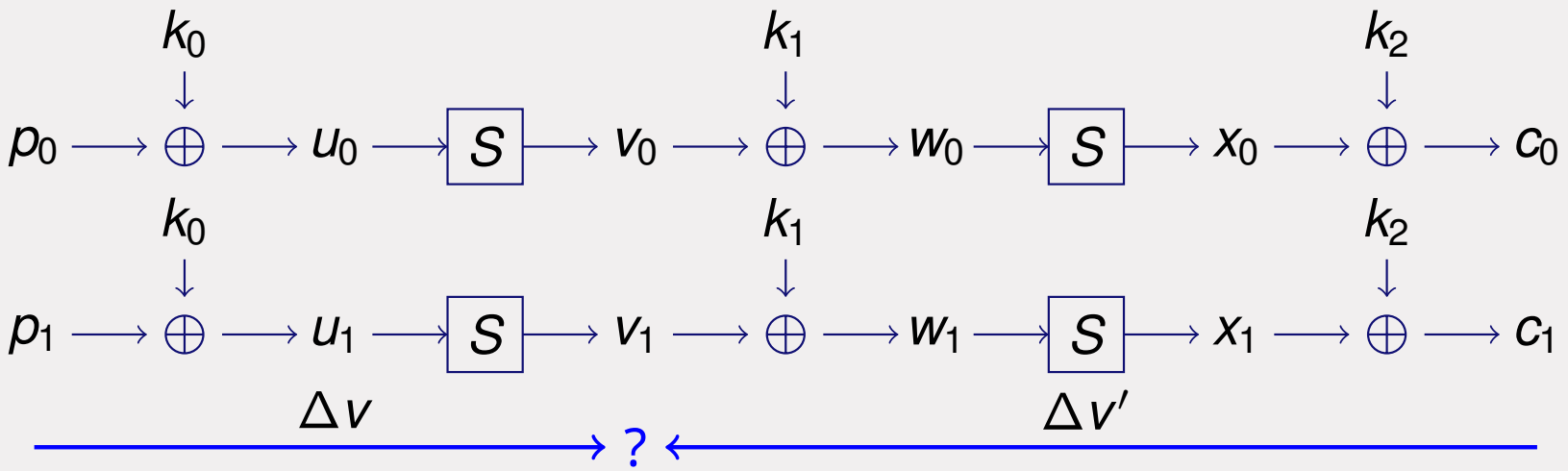
Differential Attack on a Two-Round Cipher



We filter wrong key guesses:

1. consider 16 plaintexts-ciphertexts pairs (p_0^i, c_0^i) and (p_1^i, c_1^i) s.t. $p_0^i \oplus p_1^i = \mathsf{f}$ for $i = 0, \ldots, 15$;
2. guess the last round key k_2 (iterate over all values);

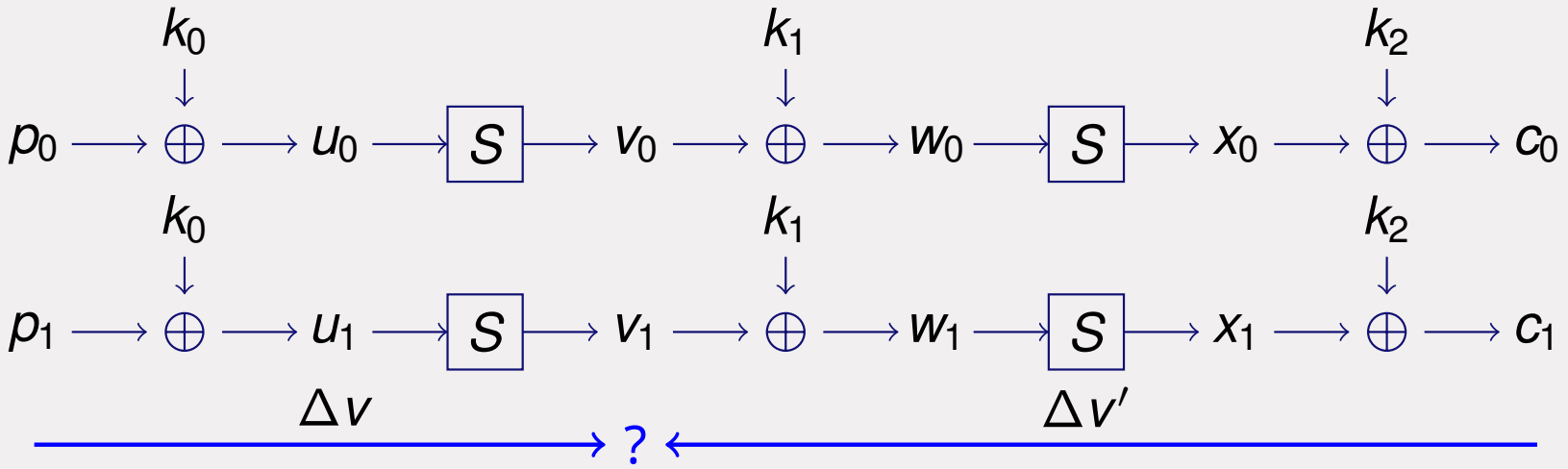
Differential Attack on a Two-Round Cipher



We filter wrong key guesses:

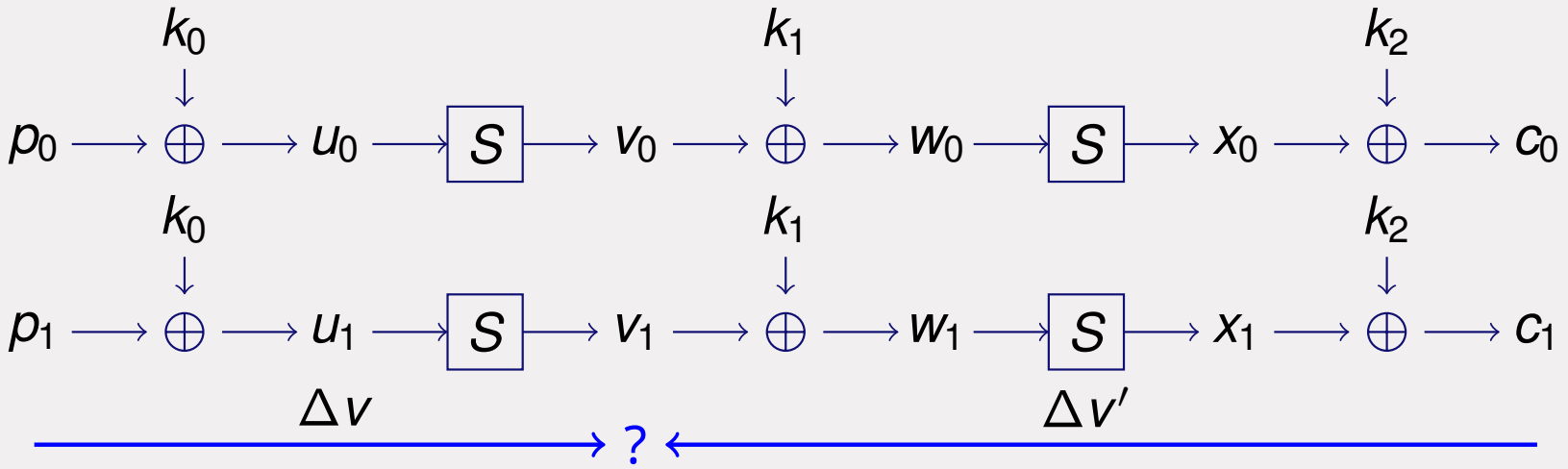
1. consider 16 plaintexts-ciphertexts pairs (p_0^i, c_0^i) and (p_1^i, c_1^i) s.t. $p_0^i \oplus p_1^i = \mathsf{f}$ for $i = 0, \ldots, 15$;
2. guess the last round key k_2 (iterate over all values);

Differential Attack on a Two-Round Cipher



3. for each plaintexts-ciphertexts pair:
   - compute x_0', x_1' and w_0', w_1';
   - count the number of pairs for which $\Delta v' = \Delta w' = \mathrm{d}$;
4. on average
   - for the *right key*: $16 \cdot \frac{10}{16} = 10$ pairs satisfy $\Delta v' = \mathrm{d}$;
   - for a *wrong key*: $16 \cdot \frac{1}{16} = 1$ pair satisfies $\Delta v' = \mathrm{d}$.

Differential Attack on a Two-Round Cipher



3. for each plaintexts-ciphertexts pair:
   - compute x_0', x_1' and w_0', w_1';
   - count the number of pairs for which $\Delta v' = \Delta w' = \mathrm{d}$;
4. on average
   - for the *right key*: $16 \cdot \frac{10}{16} = 10$ pairs satisfy $\Delta v' = \mathrm{d}$;
   - for a *wrong key*: $16 \cdot \frac{1}{16} = 1$ pair satisfies $\Delta v' = \mathrm{d}$.

TU/e

Why Prob. $1/16$ for a Wrong Guessed Key?

Observe:

$$\Delta v' \equiv \Delta w' = S^{-1}(c_0 \oplus k_2') \oplus S^{-1}(c_1 \oplus k_2') =$$
$$= S^{-1}(S(w_0) \oplus k_2 \oplus k_2') \oplus S^{-1}(S(w_1) \oplus k_2 \oplus k_2')$$

- If $k_2 = k_2'$, then $\Delta v' = w_0 \oplus w_1 = \Delta v$, that is,

$$\text{Prob}(\Delta v' = \text{d}) = 10/16;$$

- Otherwise, *since k_2 is unknown and uniform distributed*, it follows that $\Delta v'$ is uniform distributed, that is,

$$\text{Prob}(\Delta v' = \text{d}) = 1/16.$$

Why Prob. $1/16$ for a Wrong Guessed Key?

Observe:

$$\Delta v' \equiv \Delta w' = S^{-1}(c_0 \oplus k_2') \oplus S^{-1}(c_1 \oplus k_2') =$$
$$= S^{-1}(S(w_0) \oplus k_2 \oplus k_2') \oplus S^{-1}(S(w_1) \oplus k_2 \oplus k_2')$$

- If $k_2 = k_2'$, then $\Delta v' = w_0 \oplus w_1 = \Delta v$, that is,

$$\text{Prob}(\Delta v' = \text{d}) = 10/16;$$

- Otherwise, *since k_2 is unknown and uniform distributed*, it follows that $\Delta v'$ is uniform distributed, that is,

$$\text{Prob}(\Delta v' = \text{d}) = 1/16.$$

Why Prob. $1/16$ for a Wrong Guessed Key?

Observe:

$$\Delta v' \equiv \Delta w' = S^{-1}(c_0 \oplus k_2') \oplus S^{-1}(c_1 \oplus k_2') =$$
$$= S^{-1}(S(w_0) \oplus k_2 \oplus k_2') \oplus S^{-1}(S(w_1) \oplus k_2 \oplus k_2')$$

- If $k_2 = k_2'$, then $\Delta v' = w_0 \oplus w_1 = \Delta v$, that is,

$$\text{Prob}(\Delta v' = \text{d}) = 10/16;$$

- Otherwise, *since k_2 is unknown and uniform distributed*, it follows that $\Delta v'$ is uniform distributed, that is,

$$\text{Prob}(\Delta v' = \text{d}) = 1/16.$$

Difference Distribution Table

How can we find differences with a good probability?

- Compute all possible output differences for all input differences of an S-box;

- Equivalently: compute the number of solutions x to the equation

$$S(x \oplus \Delta u) \oplus S(x) = \Delta v\,.$$

Definition: Let F be a n to m bit function. The *difference distribution table* of F is an $2^n \times 2^m$ table whose entries are the number of valid solutions x for each differential $\Delta u \to \Delta v$.

Difference Distribution Table

How can we find differences with a good probability?

- Compute all possible output differences for all input differences of an S-box;

- Equivalently: compute the number of solutions x to the equation

$$S(x \oplus \Delta u) \oplus S(x) = \Delta v\,.$$

Definition: Let F be a n to m bit function. The *difference distribution table* of F is an $2^n \times 2^m$ table whose entries are the number of valid solutions x for each differential $\Delta u \to \Delta v$.

Difference Distribution Table

$\Delta_{in} \setminus \Delta_{out}$	0	1	2	3	4	5	6	7	8	9	a	b	c	*d*	e	f
0	16	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
1	-	-	6	-	-	-	-	2	-	2	-	-	2	-	4	-
2	-	6	6	-	-	-	-	-	-	2	2	-	-	-	-	-
3	-	-	-	6	-	2	-	-	2	-	-	-	4	-	2	-
4	-	-	-	2	-	2	4	-	-	2	2	2	-	-	2	-
5	-	2	2	-	4	-	-	4	2	-	-	2	-	-	-	-
6	-	-	2	-	4	-	-	2	2	-	2	2	2	-	-	-
7	-	-	-	-	-	4	4	-	2	2	2	2	-	-	-	-
8	-	-	-	-	-	2	-	2	4	-	-	4	-	2	-	2
9	-	2	-	-	-	2	2	2	-	4	2	-	-	-	-	2
a	-	-	-	-	2	2	-	-	-	4	4	-	2	2	-	-
b	-	-	-	2	2	-	2	2	2	-	-	4	-	-	2	-
c	-	4	-	2	-	2	-	-	2	-	-	-	-	-	6	-
d	-	-	-	-	-	-	2	2	-	-	-	-	6	2	-	4
e	-	2	-	4	2	-	-	-	-	-	2	-	-	-	-	6
f	-	-	-	-	2	-	2	-	-	-	-	-	-	*10*	-	2

Maximum Differential Probability

Let F be defined as before. The *Differential Probability* of F is defined as

$$DP^{\mathsf{F}}(\Delta u \to \Delta v) := \text{Prob}[\mathsf{F}(x \oplus \Delta u) \oplus \mathsf{F}(x) = \Delta v] = \\ = \frac{|\{x \mid \mathsf{F}(x \oplus \Delta u) \oplus \mathsf{F}(x) = \Delta v\}|}{2^n}$$

Definition: The *Maximum Differential Probability* is defined as

$$DP^{\mathsf{F}}_{\max} := \max_{\Delta u \neq 0, \Delta v} DP^{\mathsf{F}}(\Delta u \to \Delta v).$$

In the previous example:

$$DP^{S}_{\max} = \max_{\Delta u \neq 0, \Delta v} \frac{|\{x \mid S(x \oplus \Delta u) \oplus S(x) = \Delta v\}|}{16} = \frac{10}{16}$$

Maximum Differential Probability

Let F be defined as before. The *Differential Probability* of F is defined as

$$DP^{\mathsf{F}}(\Delta u \to \Delta v) := \text{Prob}[\mathsf{F}(x \oplus \Delta u) \oplus \mathsf{F}(x) = \Delta v] = \\ = \frac{|\{x \mid \mathsf{F}(x \oplus \Delta u) \oplus \mathsf{F}(x) = \Delta v\}|}{2^n}$$

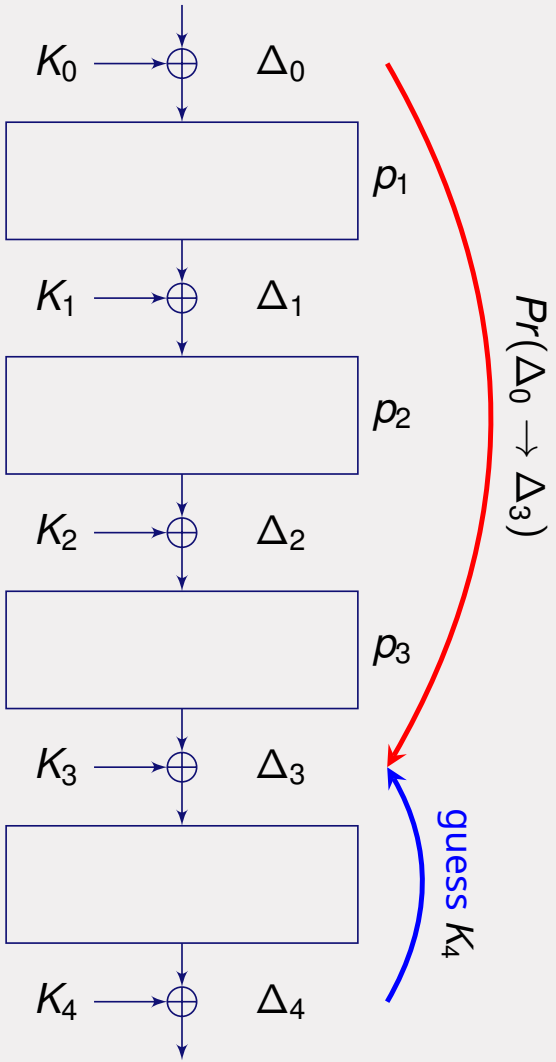
Definition: The *Maximum Differential Probability* is defined as

$$DP^{\mathsf{F}}_{\max} := \max_{\Delta u \neq 0, \Delta v} DP^{\mathsf{F}}(\Delta u \to \Delta v).$$

In the previous example:

$$DP^{S}_{\max} = \max_{\Delta u \neq 0, \Delta v} \frac{|\{x \mid S(x \oplus \Delta u) \oplus S(x) = \Delta v\}|}{16} = \frac{10}{16}$$

Basic Approach of a Differential Attack



1. Find "good" differential characteristic

$$\Delta_0 \rightarrow \Delta_1 \rightarrow \Delta_2 \rightarrow \Delta_3$$

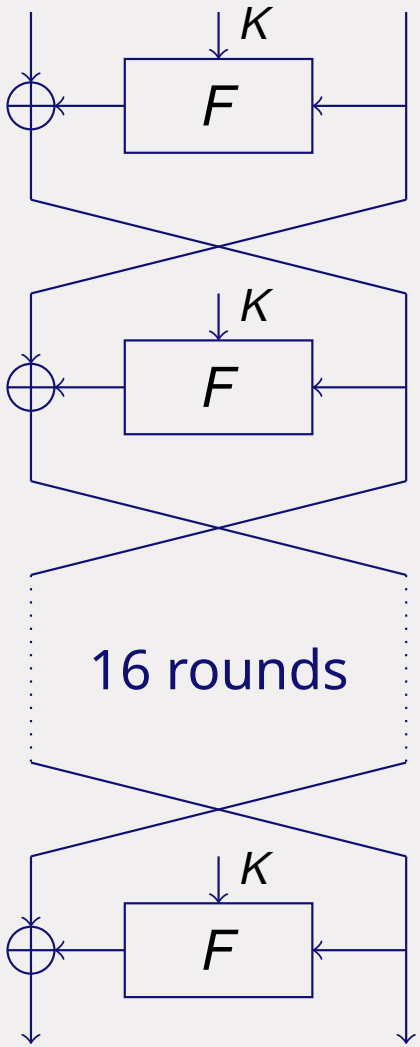
2. Guess final key K_4' and compute backward through the S-boxes to determine Δ_3'

3. The right key satisfies $\Delta_3' = \Delta_3$ with probability $\mathsf{Prob}(\Delta_0 \rightarrow \Delta_3)$, while a wrong key satisfies $\Delta_3' = \Delta_3$ with probability $|\mathcal{P}|^{-1}$.

TU/e

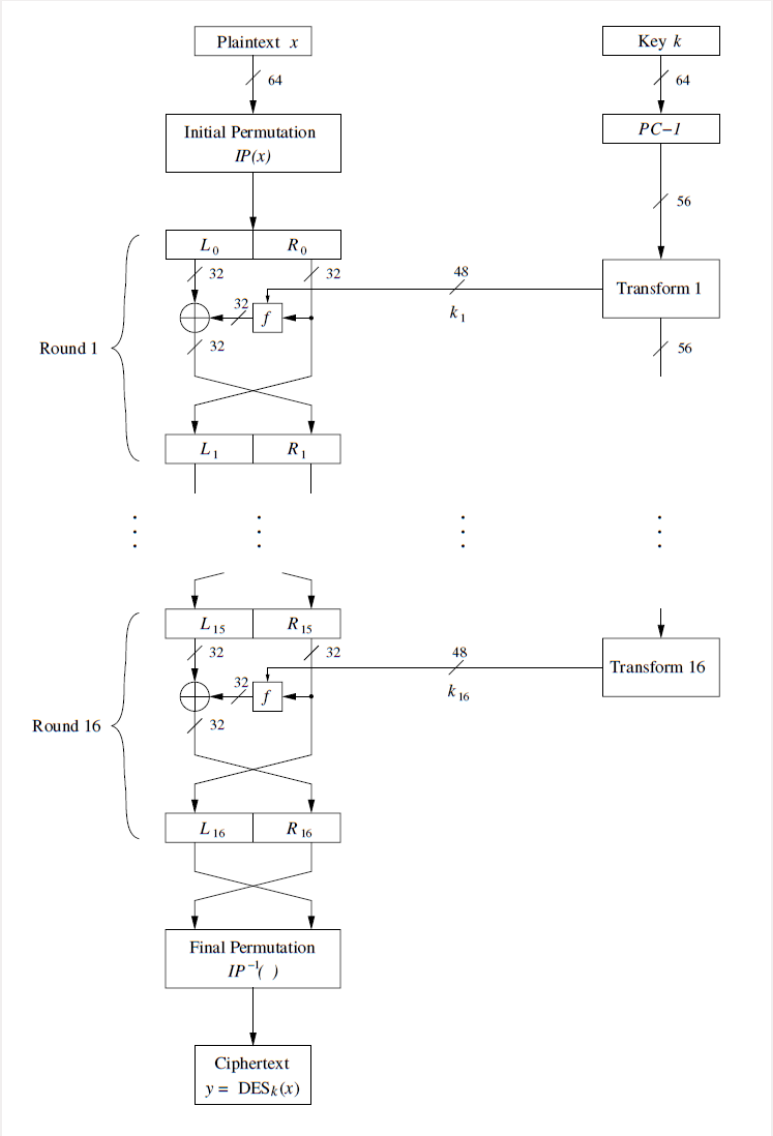
Attacks on DES

Data Encryption Standard (DES)

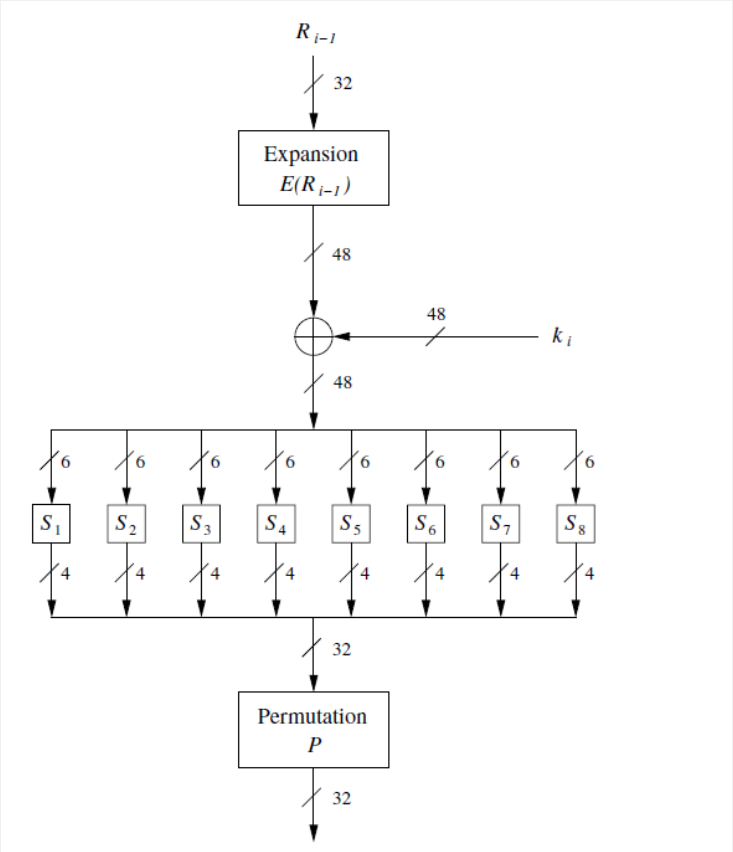


- 1970 need for an encryption standard
- 1973-1977 development of a block cipher DES
  - IBM together with NBS (= today's NIST)
  - consulted by NSA
- Encrypts blocks of 64 bits
- Effective key length of 56 bits
- Structure:
  - Initial bit shuffle
  - 16 iterations of a round transformation (Feistel)
  - Inverse bit shuffle

DES in Details

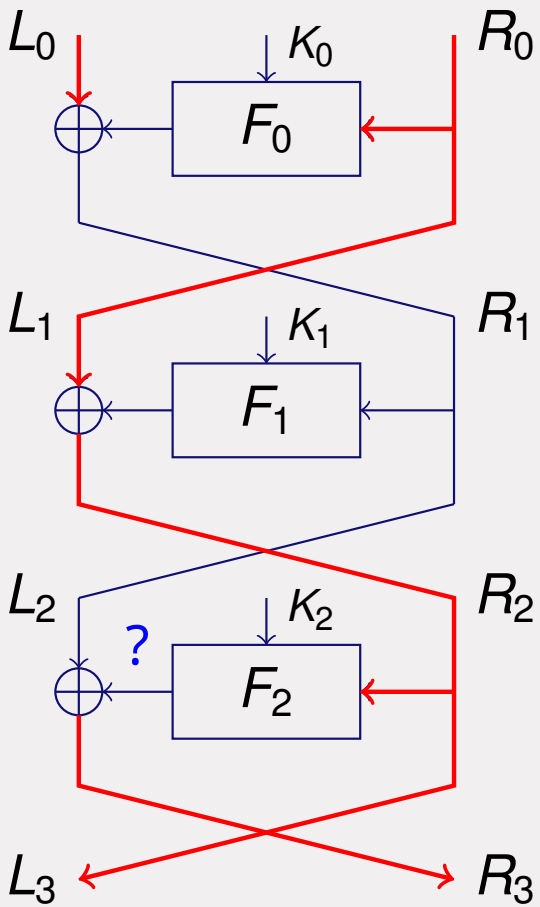


The DES Round Function



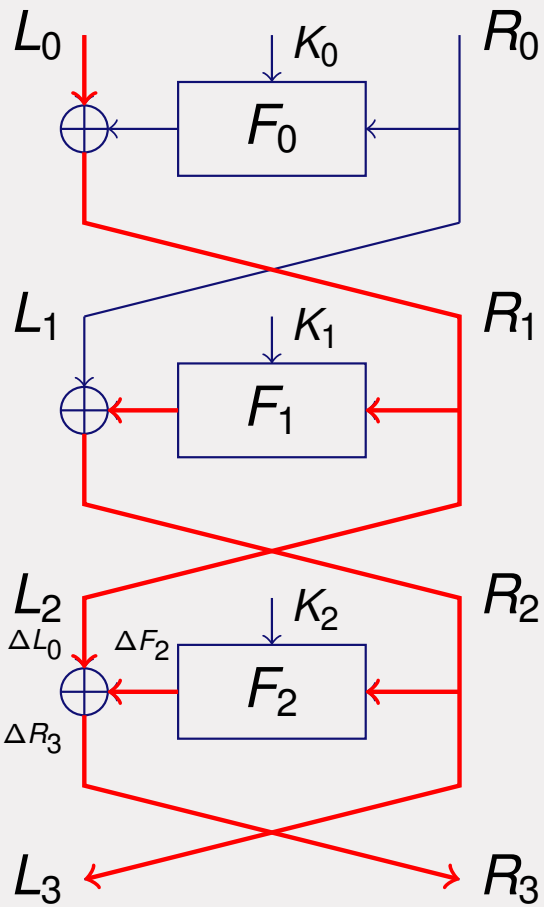
Note: the 8 S-boxes $S_0, \ldots, S_7$ (6 to 4 bits) are the only *non-linear* operations.

Attack on 3 rounds of DES



- Plaintext and ciphertext pairs known
  - propagate known differences backward and forward
  - input to F_2 and S-boxes known in last round
- but: output difference of F_2 in last round unknown

Attack on 3 rounds of DES



Remember: DC is a chosen plaintext attack.

Choose $\Delta R_0 = 0$:

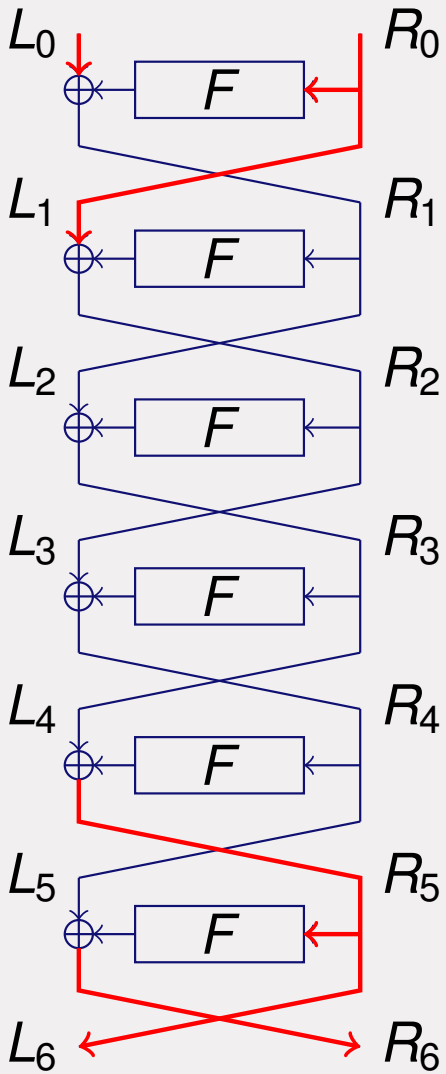
- get $\Delta F_2 = \Delta R_3 \oplus \Delta L_0$;
- input and output differences of S-boxes in F_2 known;

$\Rightarrow$ the set of possible key values for K_2 can be determined as in the basic attack.

Attack on 3 rounds of DES

- The last round key can be determined by using several pairs of plaintexts with $\Delta R_0 = 0$.
- The remaining key bits can be determined by brute force.

Attack on 6 rounds of DES



- Input difference of F in the last round known as before.
- Output difference of F in the last round cannot be calculated as easily as before...

$\Rightarrow$ we need a probabilistic approach again

Differential Characteristics

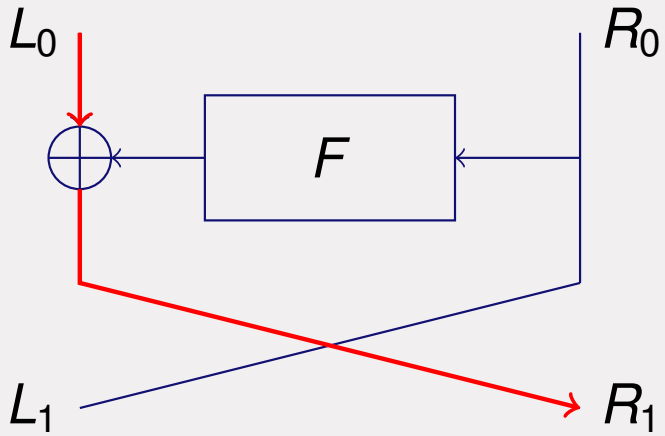
Differential characteristic:

- ΔL_0, ΔR_0;
- after 1 round: ΔL_1, ΔR_1 with prob. p_1;
- after 2 rounds: ΔL_2, ΔR_2 with extra prob. p_2;

  $\vdots$
- after n rounds: ΔL_n, ΔR_n with extra prob. p_n.

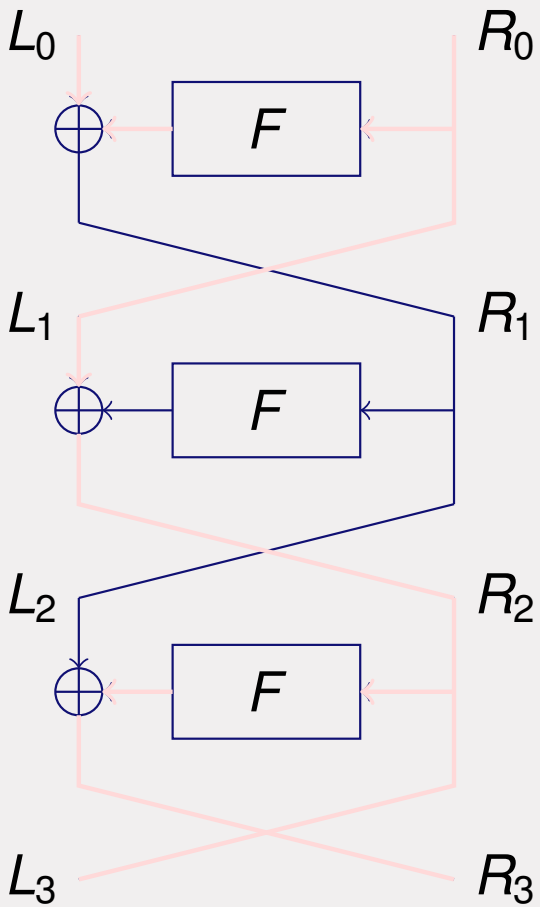
Note: p_i is the probability that $\Delta L_{i-1}, \Delta R_{i-1}$ is mapped to $\Delta L_i, \Delta R_i$ in round i.

A 1-round Characteristic



- $\Delta L_0 =$ arbitrary, $\Delta R_0 = 00000000$
- $\Delta L_1 = 0000000$, $\Delta R_1 = \Delta L_0$
- $p_1 = 1$

A 3-round Characteristic



- $\Delta L_0 = 40080000, \Delta R_0 = 04000000$

  $p_1 = 1/4$

- $\Delta L_1 = 04000000, \Delta R_1 = 00000000$

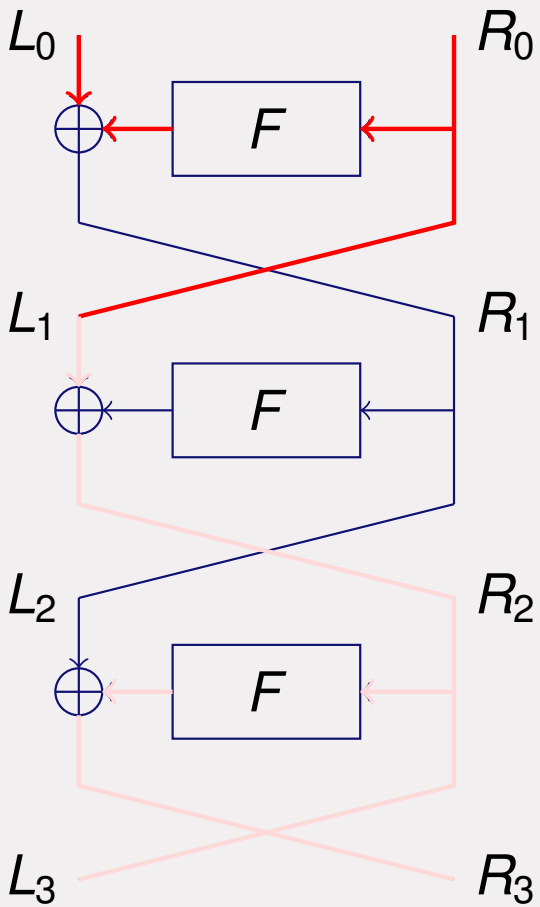
  $p_2 = 1$

- $\Delta L_2 = 00000000, \Delta R_2 = 04000000$

  $p_3 = 1/4$

- $\Delta L_3 = 04000000, \Delta R_3 = 40080000$

A 3-round Characteristic



- $\Delta L_0 = 40080000, \Delta R_0 = 04000000$

  $p_1 = 1/4$

- $\Delta L_1 = 04000000, \Delta R_1 = 00000000$

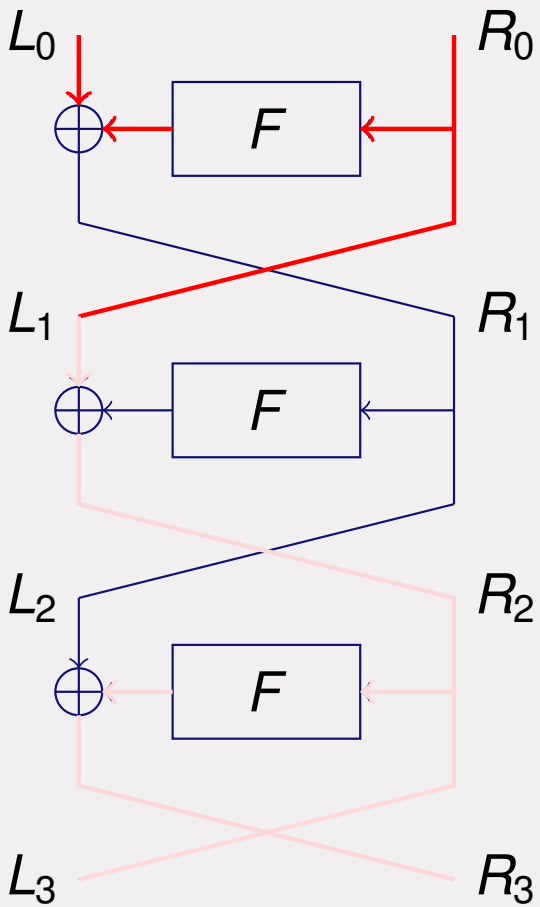
  $p_2 = 1$

- $\Delta L_2 = 00000000, \Delta R_2 = 04000000$

  $p_3 = 1/4$

- $\Delta L_3 = 04000000, \Delta R_3 = 40080000$

A 3-round Characteristic



- $\Delta L_0 = 40080000, \Delta R_0 = 04000000$

  $p_1 = 1/4$

- $\Delta L_1 = 04000000, \Delta R_1 = 00000000$

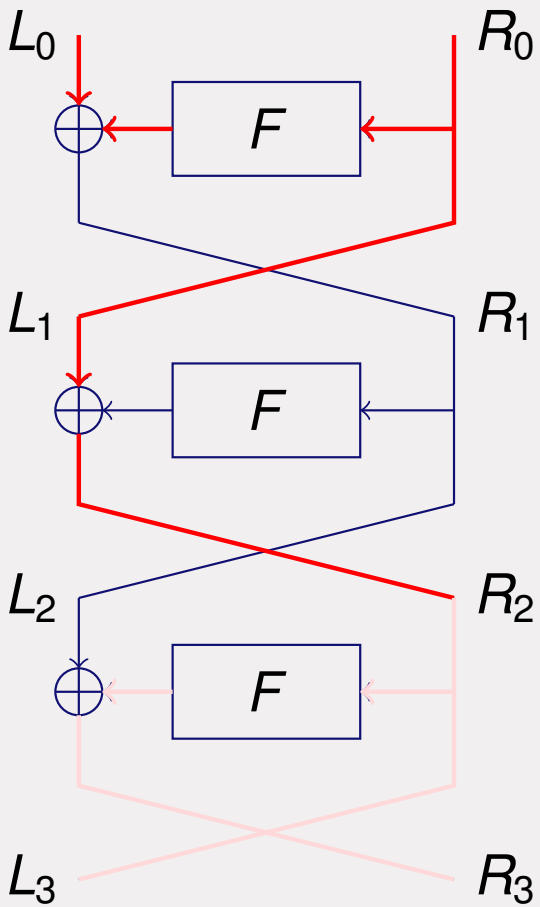
  $p_2 = 1$

- $\Delta L_2 = 00000000, \Delta R_2 = 04000000$

  $p_3 = 1/4$

- $\Delta L_3 = 04000000, \Delta R_3 = 40080000$

A 3-round Characteristic



- $\Delta L_0 = 40080000, \Delta R_0 = 04000000$

  $p_1 = 1/4$

- $\Delta L_1 = 04000000, \Delta R_1 = 00000000$

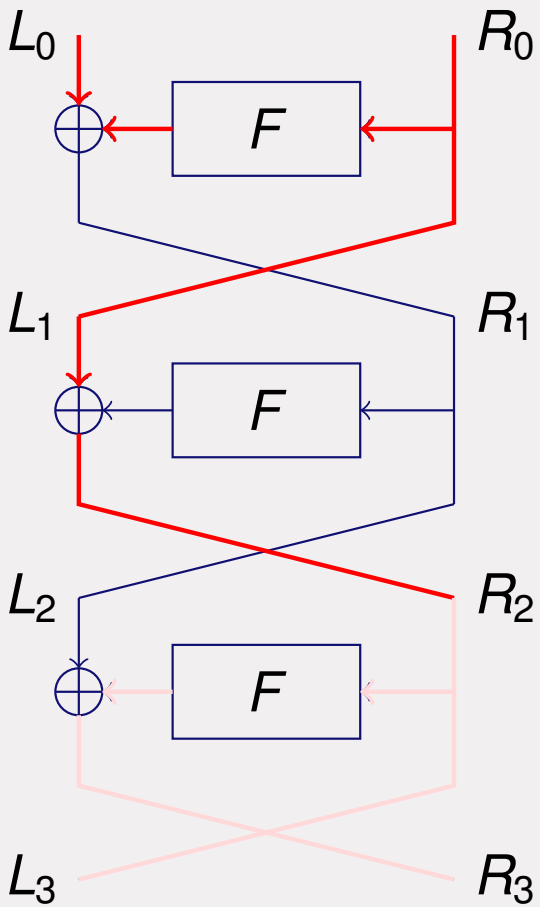
  $p_2 = 1$

- $\Delta L_2 = 00000000, \Delta R_2 = 04000000$

  $p_3 = 1/4$

- $\Delta L_3 = 04000000, \Delta R_3 = 40080000$

A 3-round Characteristic



- $\Delta L_0 = 40080000, \Delta R_0 = 04000000$

  $p_1 = 1/4$

- $\Delta L_1 = 04000000, \Delta R_1 = 00000000$

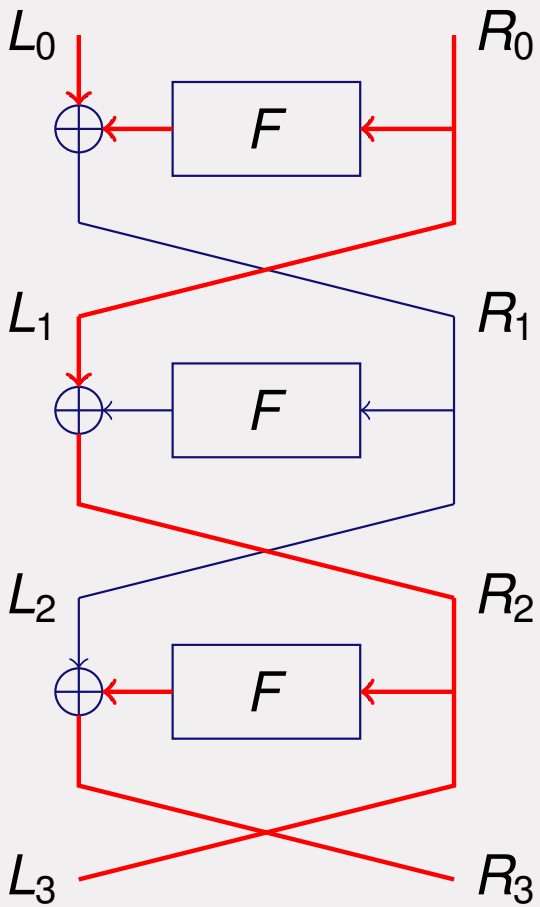
  $p_2 = 1$

- $\Delta L_2 = 00000000, \Delta R_2 = 04000000$

  $p_3 = 1/4$

- $\Delta L_3 = 04000000, \Delta R_3 = 40080000$

A 3-round Characteristic



- $\Delta L_0 = 40080000$, $\Delta R_0 = 04000000$

  $p_1 = 1/4$

- $\Delta L_1 = 04000000$, $\Delta R_1 = 00000000$

  $p_2 = 1$

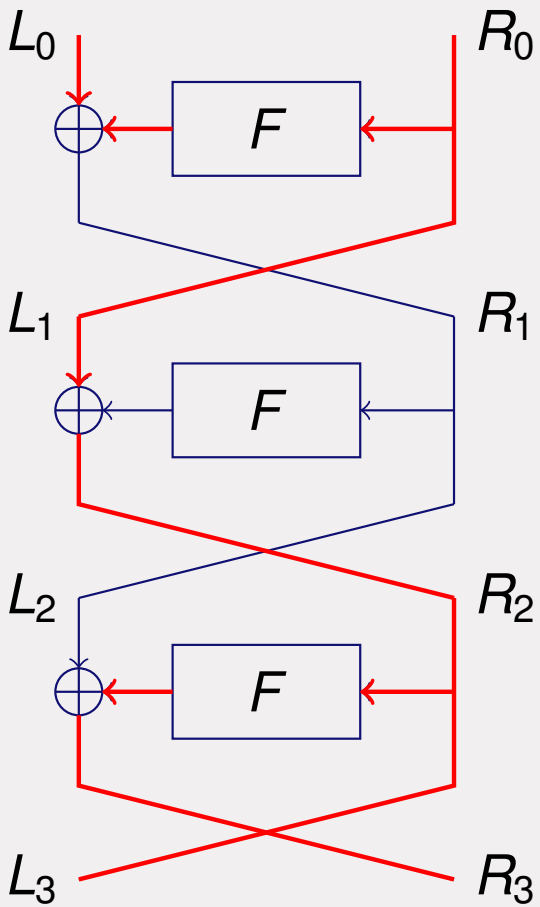
- $\Delta L_2 = 00000000$, $\Delta R_2 = 04000000$

  $p_3 = 1/4$

- $\Delta L_3 = 04000000$, $\Delta R_3 = 40080000$

TU/e

A 3-round Characteristic



- $\Delta L_0 = 40080000, \Delta R_0 = 04000000$

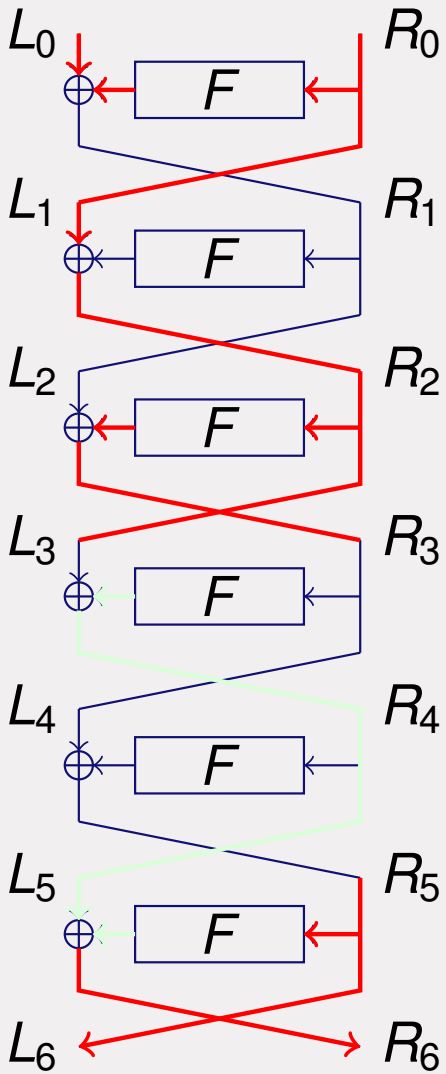
  $p_1 = 1/4$
- $\Delta L_1 = 04000000, \Delta R_1 = 00000000$

  $p_2 = 1$
- $\Delta L_2 = 00000000, \Delta R_2 = 04000000$

  $p_3 = 1/4$
- $\Delta L_3 = 04000000, \Delta R_3 = 40080000$

TU/e

Attack on 6 rounds of DES



- Use the 3-round characteristic to (partly) determine ΔF in the last round
- $\Delta L_3 = 04000000$ and $\Delta R_3 = 40080000$
  - ΔR_3 is chosen in such a way that there are no differences at the input of S_1, S_4, S_5, S_6 and S_7 in round 4

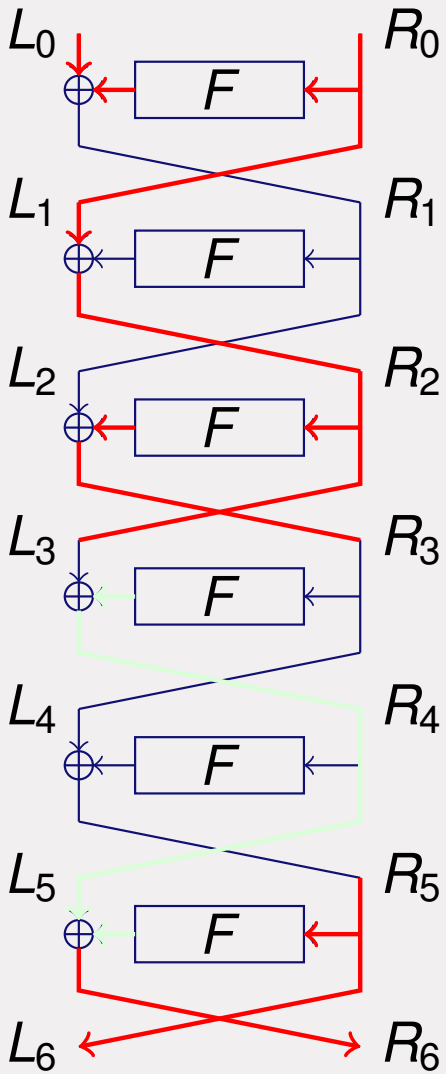
  $\Rightarrow \Delta L_5 = \Delta R_4 = P(?0??0000) \oplus 04000000$
- We get the following differences at the output of the S-boxes in the last round:

  $P^{-1}(\Delta L_5 \oplus \Delta R_6) = P^{-1}(\Delta R_6 \oplus 04000000) \oplus ?0??0000$
- We know ΔR_6, so only the output differences of S_0, S_2 and S_3 are unknown

  $\Rightarrow$ The key bytes K_1, K_4, K_5, K_6 and K_7 of the last subkey can be determined!

TU/e

Attack on 6 rounds of DES



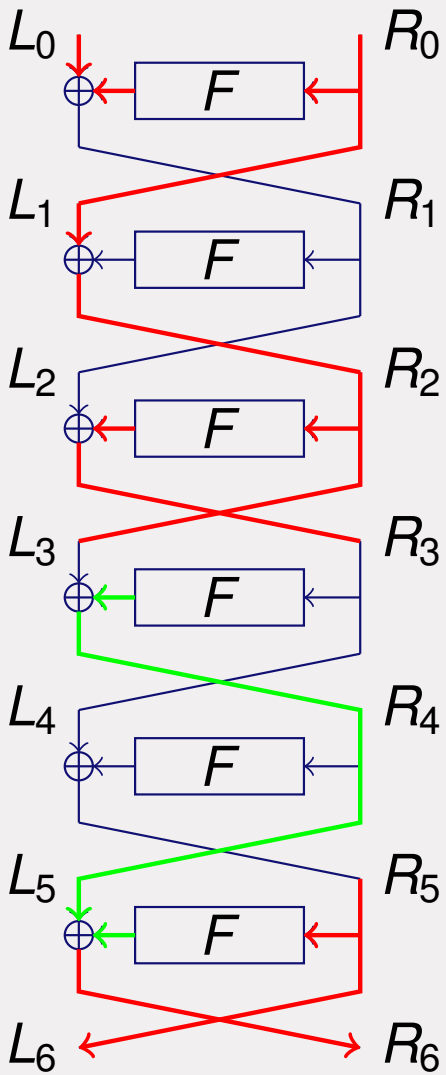
- Use the 3-round characteristic to (partly) determine ΔF in the last round
- $\Delta L_3 = 04000000$ and $\Delta R_3 = 40080000$
  - ΔR_3 is chosen in such a way that there are no differences at the input of S_1, S_4, S_5, S_6 and S_7 in round 4

  $\Rightarrow \Delta L_5 = \Delta R_4 = P(?0??0000) \oplus 04000000$
- We get the following differences at the output of the S-boxes in the last round:

  $P^{-1}(\Delta L_5 \oplus \Delta R_6) = P^{-1}(\Delta R_6 \oplus 04000000) \oplus ?0??0000$
- We know ΔR_6, so only the output differences of S_0, S_2 and S_3 are unknown

  $\Rightarrow$ The key bytes K_1, K_4, K_5, K_6 and K_7 of the last subkey can be determined!

Attack on 6 rounds of DES



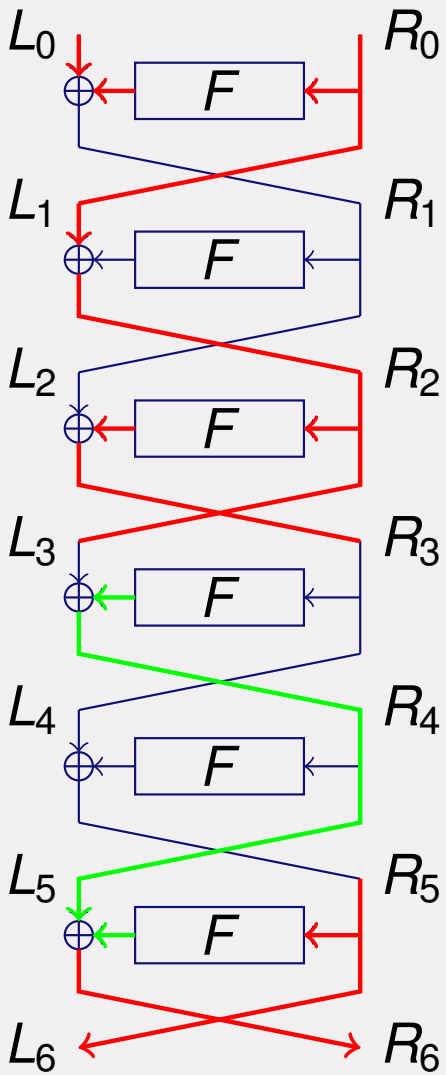
- Use the 3-round characteristic to (partly) determine ΔF in the last round
- $\Delta L_3 = 04000000$ and $\Delta R_3 = 40080000$
  - ΔR_3 is chosen in such a way that there are no differences at the input of S_1, S_4, S_5, S_6 and S_7 in round 4

  $\Rightarrow \Delta L_5 = \Delta R_4 = P(?0??0000) \oplus 04000000$
- We get the following differences at the output of the S-boxes in the last round:

  $P^{-1}(\Delta L_5 \oplus \Delta R_6) = P^{-1}(\Delta R_6 \oplus 04000000) \oplus ?0??0000$
- We know ΔR_6, so only the output differences of S_0, S_2 and S_3 are unknown

  $\Rightarrow$ The key bytes K_1, K_4, K_5, K_6 and K_7 of the last subkey can be determined!

Attack on 6 rounds of DES



- Use the 3-round characteristic to (partly) determine ΔF in the last round
- $\Delta L_3 = 04000000$ and $\Delta R_3 = 40080000$
  - ΔR_3 is chosen in such a way that there are no differences at the input of S_1, S_4, S_5, S_6 and S_7 in round 4

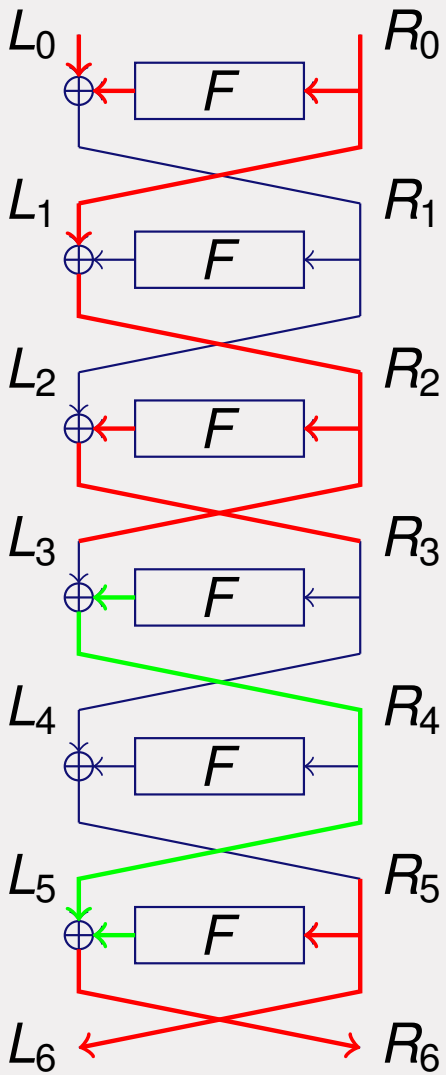
  $\Rightarrow \Delta L_5 = \Delta R_4 = P(?0??0000) \oplus 04000000$
- We get the following differences at the output of the S-boxes in the last round:

  $P^{-1}(\Delta L_5 \oplus \Delta R_6) = P^{-1}(\Delta R_6 \oplus 04000000) \oplus ?0??0000$
- We know ΔR_6, so only the output differences of S_0, S_2 and S_3 are unknown

  $\Rightarrow$ The key bytes K_1, K_4, K_5, K_6 and K_7 of the last subkey can be determined!

TU/e

Attack on 6 rounds of DES



- Use the 3-round characteristic to (partly) determine ΔF in the last round
- $\Delta L_3 = 04000000$ and $\Delta R_3 = 40080000$
  - ΔR_3 is chosen in such a way that there are no differences at the input of S_1, S_4, S_5, S_6 and S_7 in round 4

  $\Rightarrow \Delta L_5 = \Delta R_4 = P(?0??0000) \oplus 04000000$
- We get the following differences at the output of the S-boxes in the last round:

  $P^{-1}(\Delta L_5 \oplus \Delta R_6) = P^{-1}(\Delta R_6 \oplus 04000000) \oplus ?0??0000$
- We know ΔR_6, so only the output differences of S_0, S_2 and S_3 are unknown

  $\Rightarrow$ The key bytes K_1, K_4, K_5, K_6 and K_7 of the last subkey can be determined!

TU/e

Determining the Secret Key

- Check for possible key candidates at S-boxes S_j with $j = 1, 4, 5, 6, 7$ to determine the corresponding keys K_j of the last subkey

- Only works for

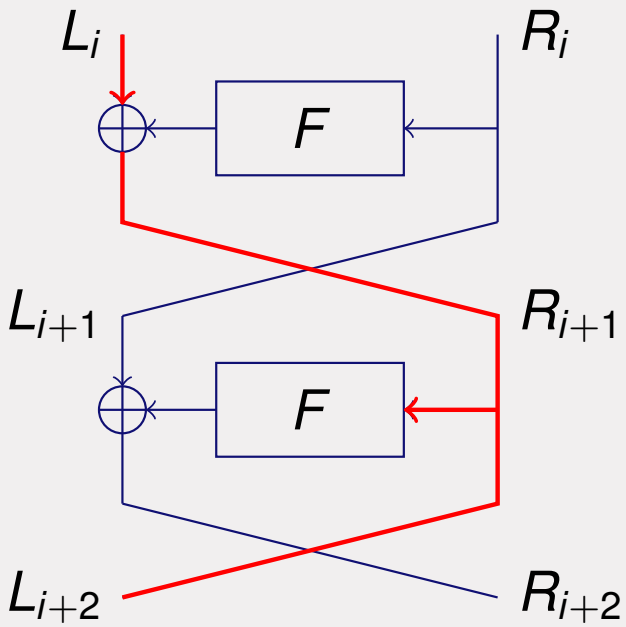
$$p_1 \times p_2 \times p_3 = \frac{1}{4} \times 1 \times \frac{1}{4} = \frac{1}{16}$$

  of the plaintext pairs

- Not all bad pairs are filtered (we get wrong key suggestions):

  Use several pairs of plaintexts to reduce the set of possible values to one element for K_1, K_4, K_5, K_6 and K_7 before combining the parts to a key string

Iterative Characteristic for 2-round DES



- Iterative characteristics:

$$\Delta L_i = \Delta R_{i+1} = \Delta L_{i+2} = 19600000$$
$$\Delta R_i = \Delta L_{i+1} = \Delta R_{i+2} = 0$$

  best characteristics for ≥ 8 rounds
- 3 active S-boxes *per* 2 rounds
- Probability $= 1/234$ over 2 rounds

Summary of the attack

1. For a given cipher of r rounds, *assume* the attacker knowns a characteristic $(\Delta_0, \Delta_1, ..., \Delta_{r-1})$ over $r-1$ rounds;

2. In an attack, the cryptanalyst chooses messages pairs with difference Δ_0 as specified by the characteristic;

3. Using the expected difference in the partially encrypted ciphertexts after $r-1$ rounds, the attacker will try to identify (some bits of) the last subkey k_r.

4. *problem: A "good" characteristic needs to be found in order to get there!*

AES Block Cipher

Defending against Differential Cryptanalysis

- To prevent Differential Cryptanalysis: keep the probability of each characteristic/differential as low as possible.
- Difficult to compute the exact probability: compute “bounds”.
- Strategy:
  - compute “Maximum Differential Probability” of the S-Box $DP_{\max}^{\text{S-Box}}$;
  - compute bound on number of active S-Boxes for each round.

$\Rightarrow$ *Design S-Boxes with low maximum values* DP_{max}, and

$\Rightarrow$ *Design mixing layers which result in many active S-Boxes.*

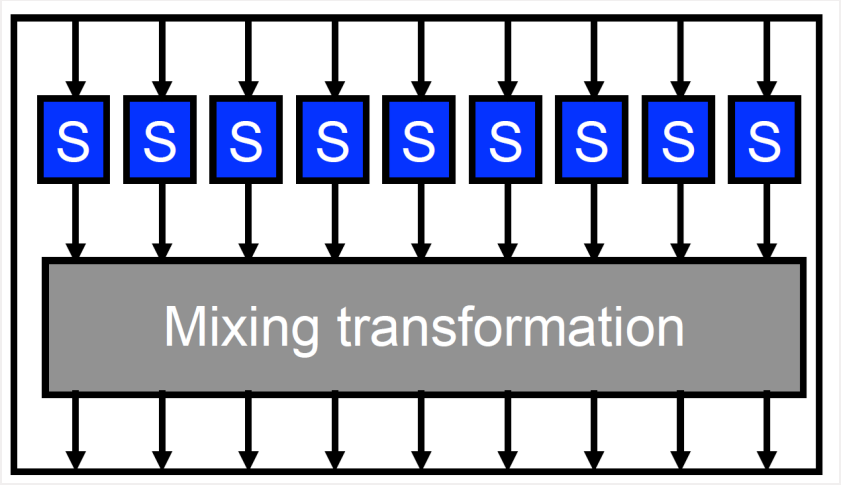
Defending against Differential Cryptanalysis

- To prevent Differential Cryptanalysis: keep the probability of each characteristic/differential as low as possible.
- Difficult to compute the exact probability: compute “bounds”.
- Strategy:
  - compute “Maximum Differential Probability” of the S-Box $DP_{\max}^{\text{S-Box}}$;
  - compute bound on number of active S-Boxes for each round.

$\Rightarrow$ *Design S-Boxes with low maximum values* DP_{max}, and

$\Rightarrow$ *Design mixing layers which result in many active S-Boxes*.

SPN: Single-Round Optimization



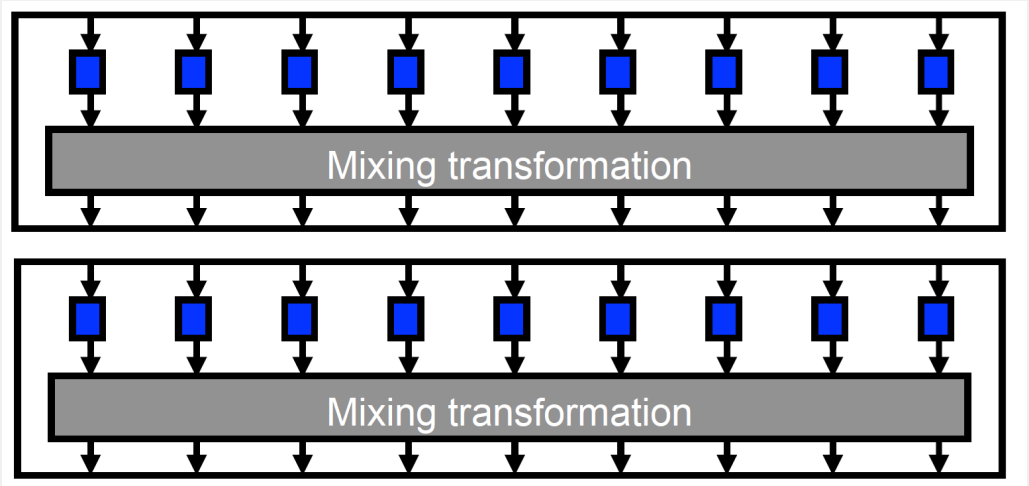
Relevant:

- Number of active components (S-Boxes) in input;
- Worst-case (max) differential probability in S-Box.

Result:

- bound of 1 active S-Box per round (= minimum number of active S-Boxes);
- design large S-Boxes with small $DP_{\max}$.

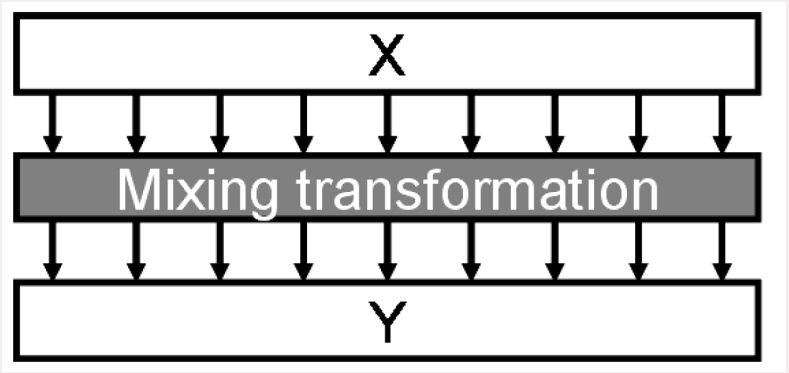
SPN: Two-Round Optimization



- Relevant: number of active S-Boxes in input and after first round.
- The number of active S-Boxes after the first round depends on the (linear) *mixing transformation*:
  - *Branch number $\mathcal{B}$: minimum number of active S-Boxes over two consecutive rounds*.

$\Rightarrow$ Provides a bound of $\mathcal{B}$ active S-Boxes per two rounds.

SPN: Designing the Mixing Transformation



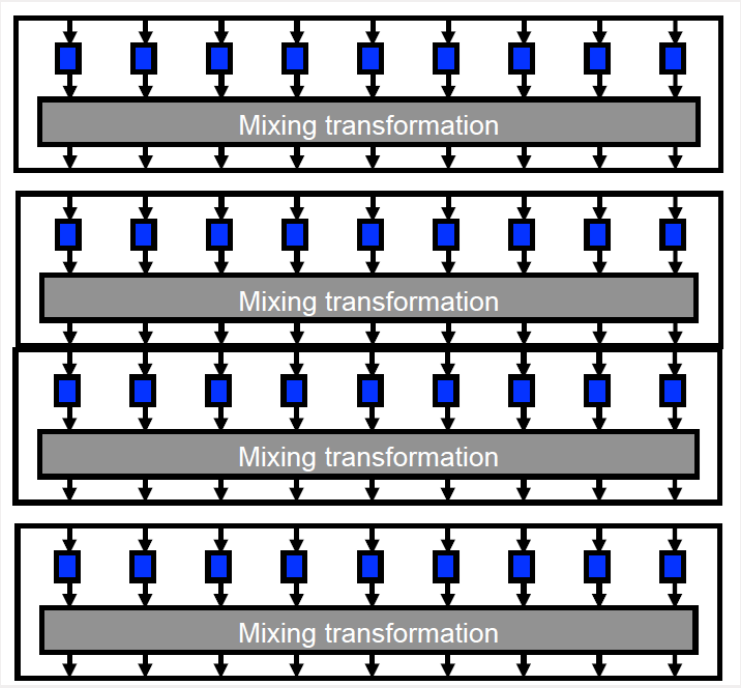
Given $Y = m(X)$, then

$\mathcal{B} \leq 1 +$ total number of components (= number of S-Boxes) in Y.

$\Rightarrow$ Design a Mixing Transformation m that maximizes $\mathcal{B}$.

A linear transformation that maximes $\mathcal{B}$ is called MDS (= Maximum Distance Separable).

SPN: Four-Round Optimization



Apply previous result on 2-round recursively:

- we can square the branch number (at almost no cost!);
- $\mathcal{B}^2$ active S-Boxes.

AES: Iterated Block Cipher

Advanced Encryption Standard (128-, 192-, 256-bit keys):

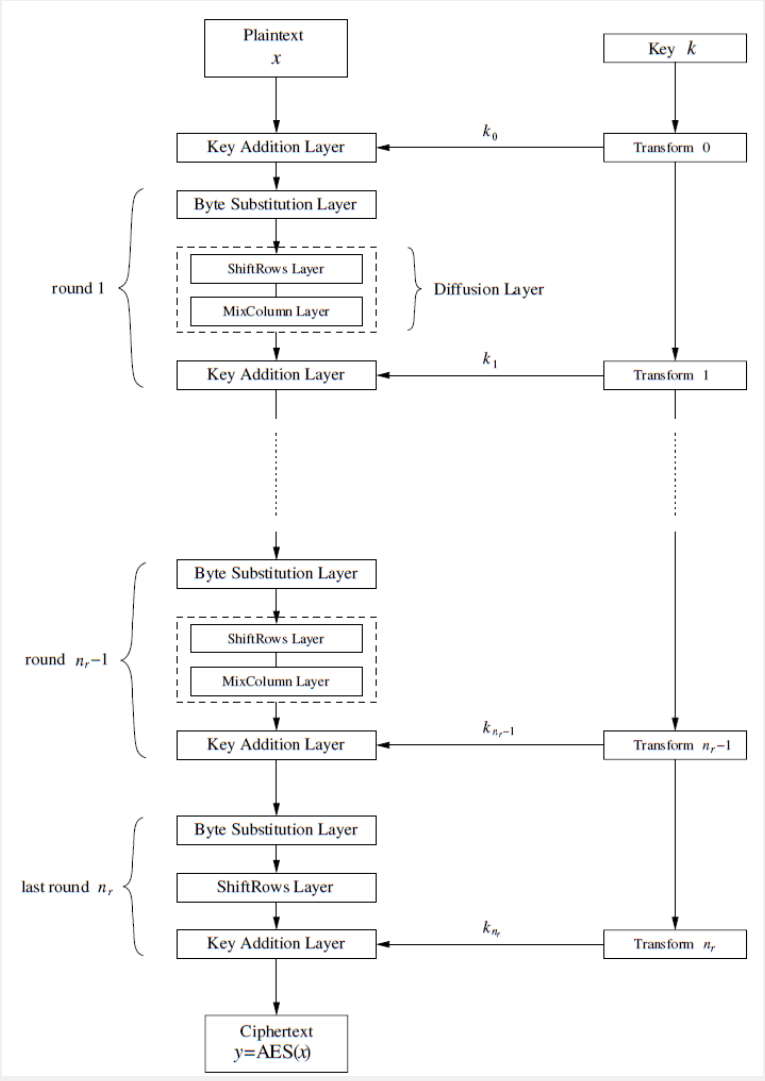
- 10, 12, 14 times applying the same round function.
- State of 4×4 bytes (128 bits).
- Round function: composed of 4 steps

$$R_k(\cdot) = k \oplus MC \circ SR \circ S - Box(\cdot)$$

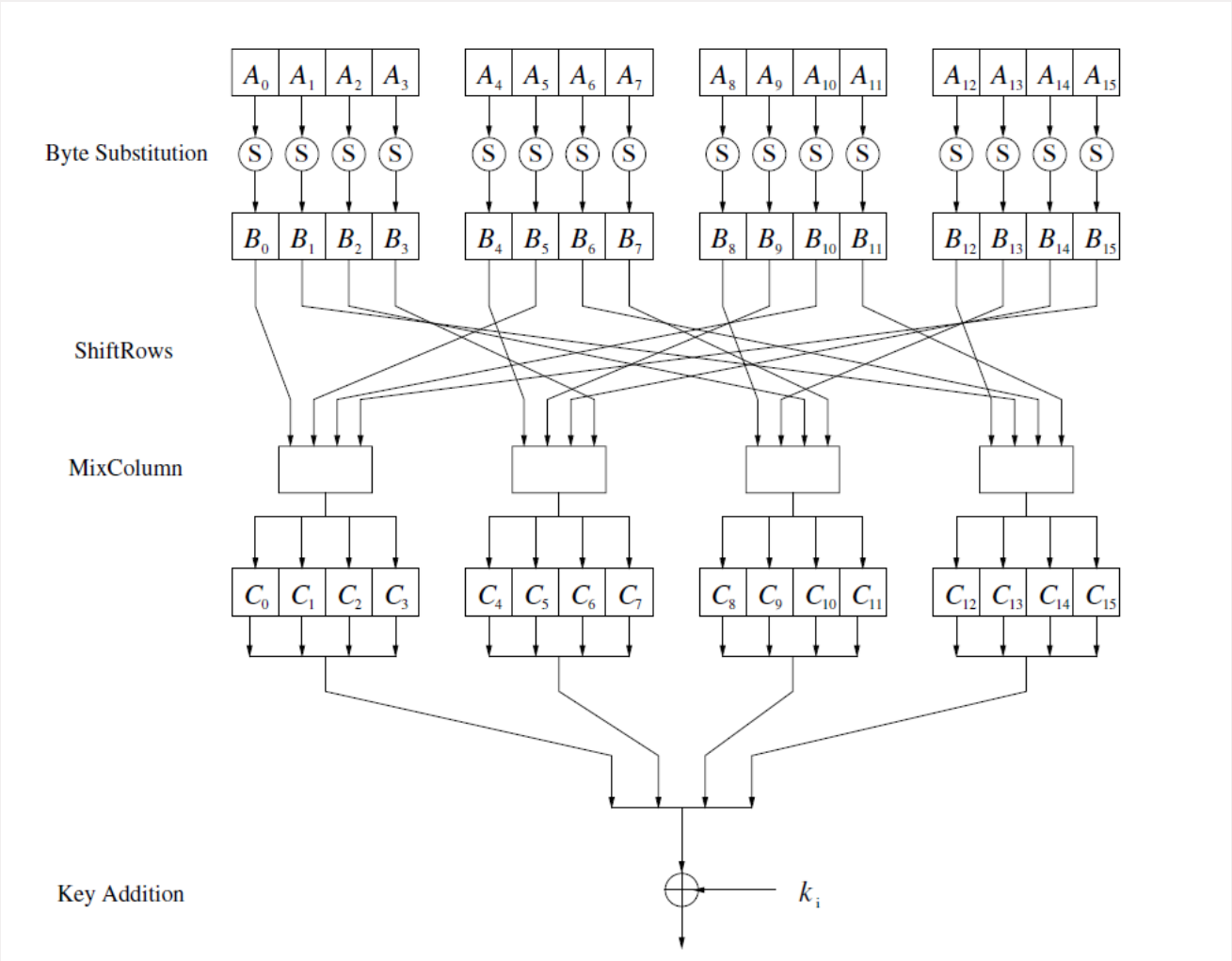
  where each step has its own particular functionality!

- Sequential and light-weight key schedule.

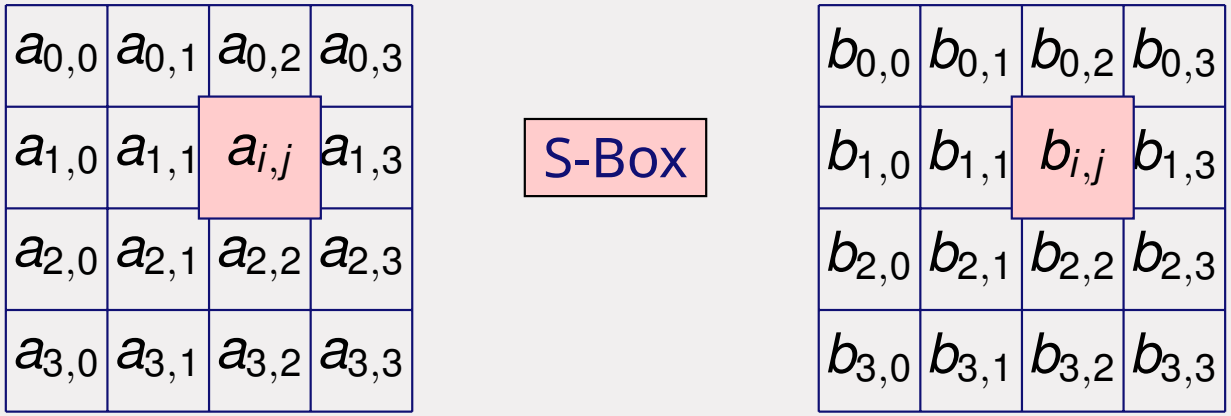
AES


Plaintext
x
Key k
Key Addition Layer
k0
Transform 0
Byte Substitution Layer
round 1
ShiftRows Layer
MixColumn Layer
Diffusion Layer
Key Addition Layer
k1
Transform 1
Byte Substitution Layer
round nr−1
ShiftRows Layer
MixColumn Layer
Key Addition Layer
knr−1
Transform nr−1
Byte Substitution Layer
last round nr
ShiftRows Layer
Key Addition Layer
knr
Transform nr
Ciphertext
y=AES(x)

The AES Round



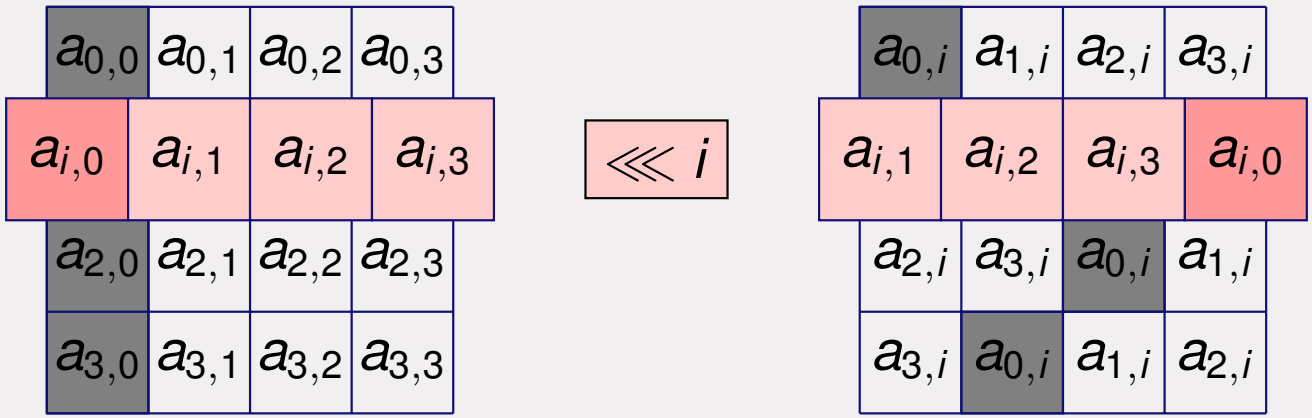
SubBytes



- Bytes are transformed by invertible S-Box with $b_{i,j} = S(a_{i,j})$.
- Same S-Box (lookup table) for the whole cipher:
  - based on multiplicative inverse in $GF(2^8)$;
  - high non-linearity *(see Lecture 8 of AK1 for details)*.
  - What about DP_{max}?

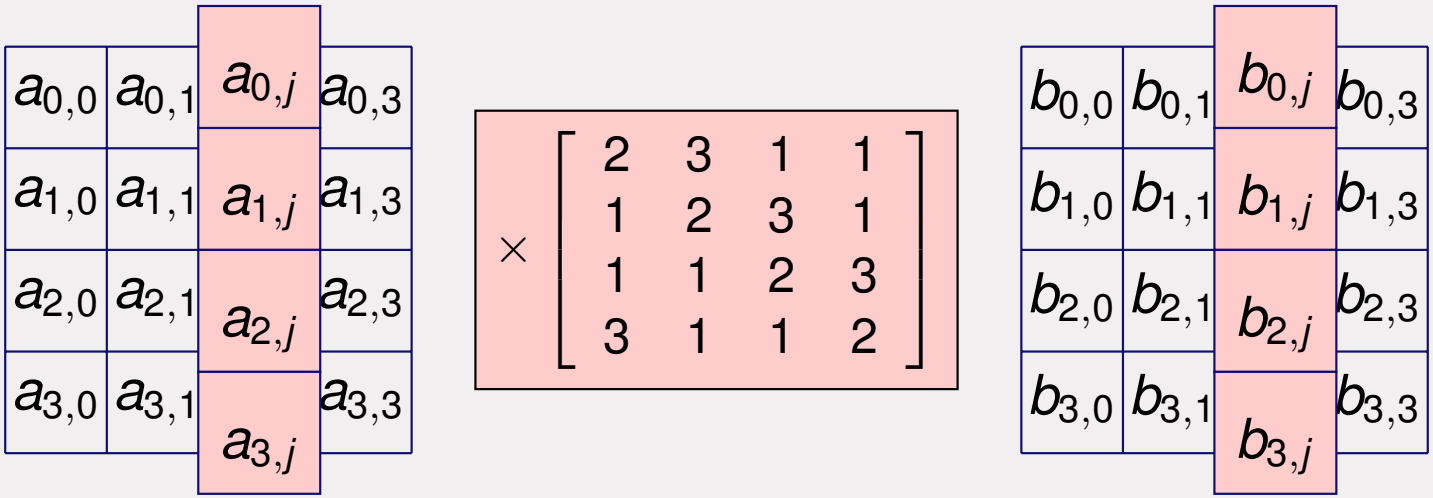
$$DP_{max}(\text{AES S-Box}) = \frac{4}{256}.$$

ShiftRows



- Rows are rotated over 4 different offsets.

MixColumns



- Columns transformed by 4×4 matrix over $GF(2^8)$;
- Linear function with Branch number $\mathcal{B} = 5$:
  - *MDS* (= Maximum Distance Separable) *matrix* (i.e., linear function that maximizes $\mathcal{B}$).
- Together with ShiftRows, *high diffusion* over multiple rounds:
  - *min.* $\mathcal{B}^2 = 25$ *active S-Boxes for 4 rounds*.

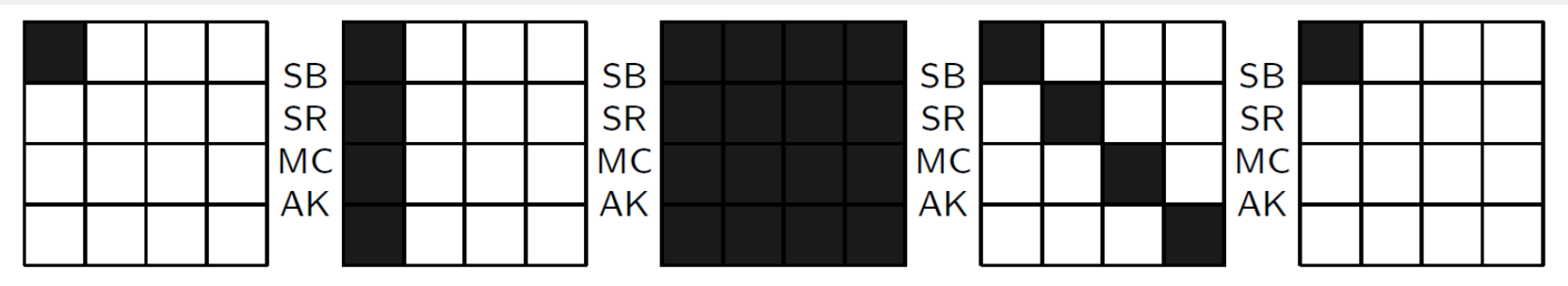
Bounds in AES (1/2)

Diffusion in AES:

- MixColumns: branch number 5;
- ShiftRows: Diffusion-optimal;

$\Rightarrow$ lower bound for number of active S-Boxes per 4 rounds: 25.

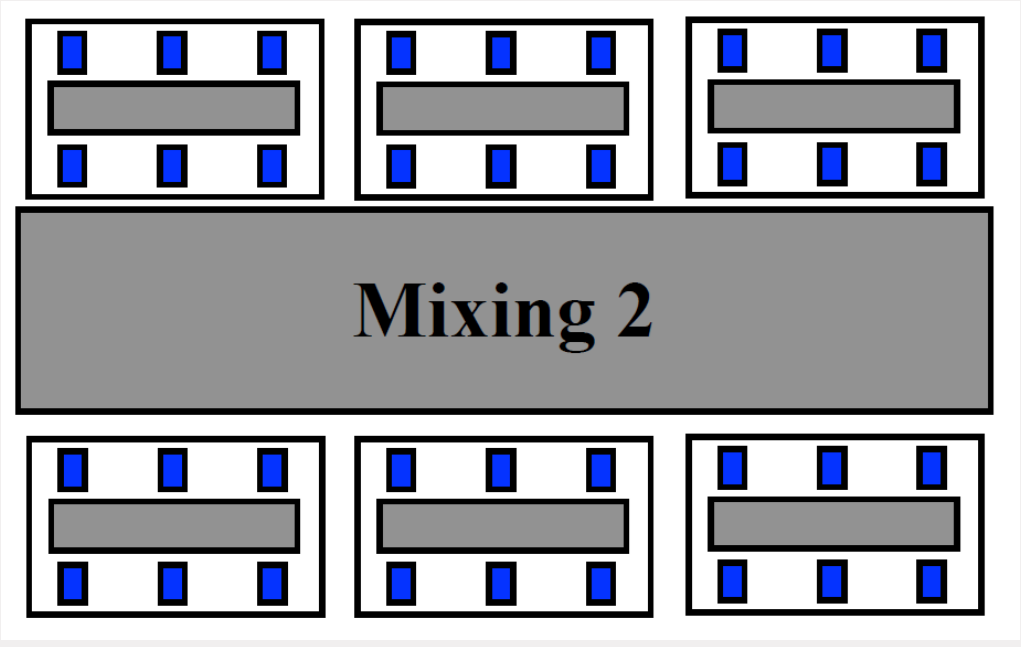
Example



where: *white byte = zero XOR-difference & black byte = non-zero XOR-difference*.

Proof via the Super-Sbox (1/2)

Re-write 4-round AES



using the *super-Sbox notation*

$$\textit{super-Sbox}(\cdot) = \text{S-Box} \circ \textit{ARK} \circ \textit{MC} \circ \text{S-Box}(\cdot).$$

Proof via the Super-Sbox (2/2)

Using the *super-Sbox notation*

$$R^4(\cdot) = ARK \circ SR \circ \textit{super-Sbox} \cdot M' \circ \textit{super-Sbox} \cdot SR \cdot ARK(\cdot)$$

where

$$M'(\cdot) = SR \circ ARK \circ MC \circ SR(\cdot),$$

and swap *(1st)* S-Box$(\cdot)$ and $SR(\cdot)$ and *(2nd)* ARK and $SR(\cdot)$.

Minimum $\mathcal{B}^2 = 25$ active S-Boxes for 4 rounds since

- each active super-Sbox contains at least 5 active S-Box;
- M' is MDS (remember that MC is MDS).

Proof via the Super-Sbox (2/2)

Using the *super-Sbox notation*

$$R^4(\cdot) = ARK \circ SR \circ \textit{super-Sbox} \cdot M' \circ \textit{super-Sbox} \cdot SR \cdot ARK(\cdot)$$

where

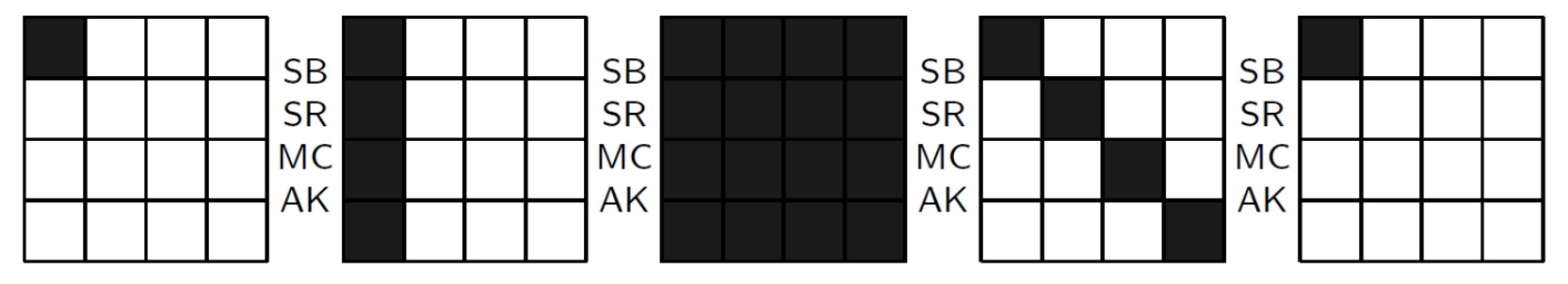
$$M'(\cdot) = SR \circ ARK \circ MC \circ SR(\cdot),$$

and swap *(1st)* S-Box$(\cdot)$ and $SR(\cdot)$ and *(2nd)* ARK and $SR(\cdot)$.

Minimum $\mathcal{B}^2 = 25$ active S-Boxes for 4 rounds since

- each active super-Sbox contains at least 5 active S-Box;
- M' is MDS (remember that MC is MDS).

Bounds in AES (2/2)



- Diffusion in AES: *lower bound* for number of active S-Boxes per 4 rounds: 25.
- AES S-Box:
  - differential probability (DP) $\leq 4/256 = 2^{-6}$, that is $DP_{max} = 2^{-6}$.
- Provable bound:
  - *probability of 4-round characteristic* $\leq (2^{-6})^{25} = 2^{-150}$ (w.r.t. of 2^{-128} for the case of a random permutation).

Conclusion

Security against Differential Attacks

- Differential attack is a very powerful attack against symmetric-key primitive.
- DES broken via differential attacks, while AES is secure.
- How to provide security?
  - Design ciphers with provable low probabilities.
  - Design ciphers with easily computable bounds.