Differential Cryptanalysis

See:

Biham and Shamir,
Differential Cryptanalysis of the Data Encryption Standard, Springer Verlag, 1993.

The Data Encryption Standard - DES

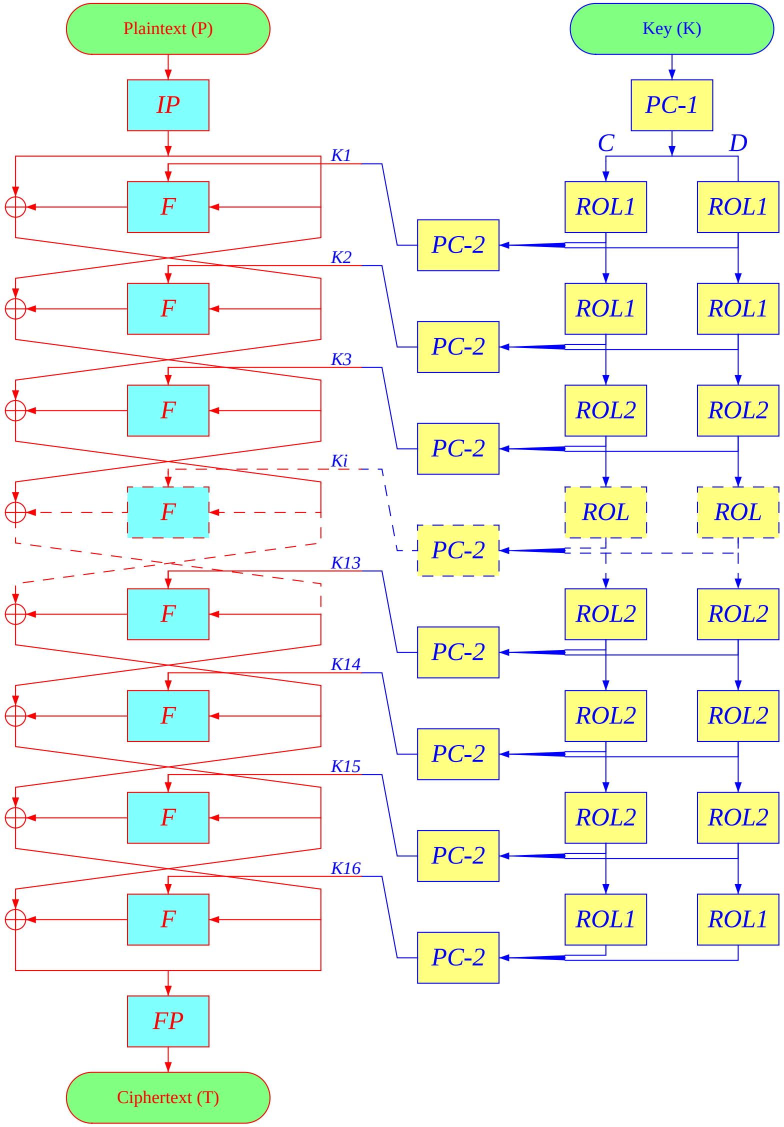
1. The most widely used cipher in civilian applications.
2. Developed by IBM; Evolved from Lucifer.
3. Accepted as an US NBS standard in 1977, and later as an international standard.
4. A block cipher with $\boldsymbol{N = 64}$ **bit blocks**.
5. **56-bit keys** (eight bytes, in each byte seven bits are used; the eighth bit can be used as a parity bit).
6. Exhaustive search requires 2^{56} encryption steps (2^{55} on average).

The Data Encryption Standard - DES (cont.)

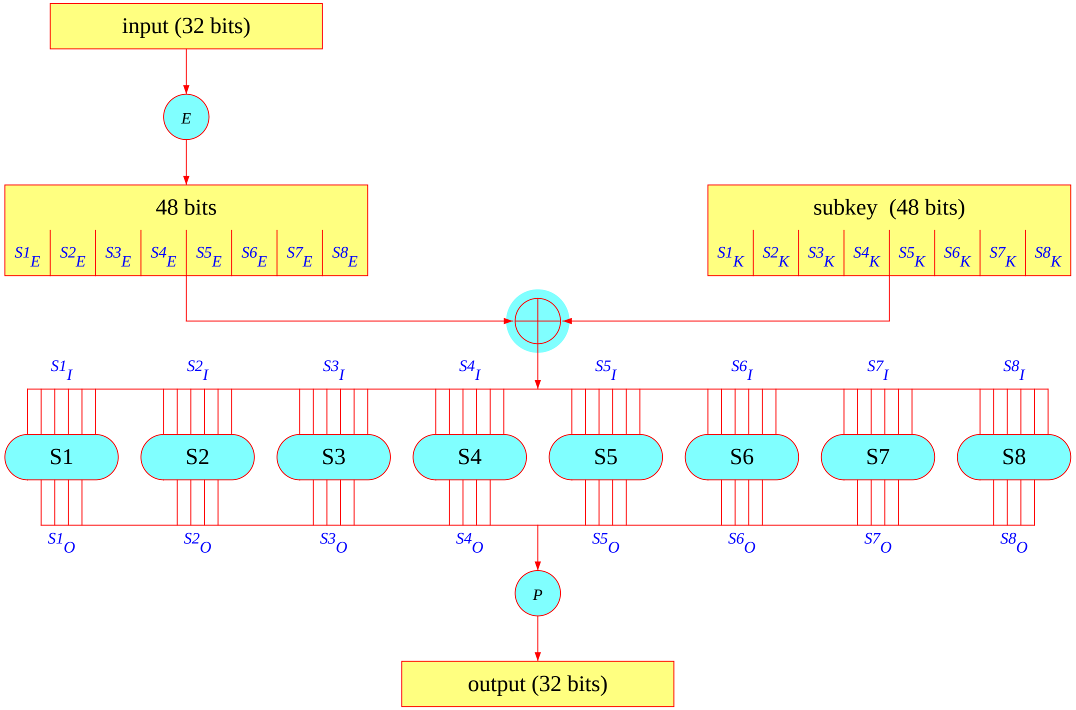
7. Iterates a round-function 16 times in 16 **rounds**. The round-function mixes the data with the key.

8. Each round, the key information entered to the round function is called a **subkey**. The subkeys $K_1, \ldots, K_{16}$ are computed by a **key scheduling algorithm**.

DES Outline


Plaintext (P)
IP
F
F
F
F
F
F
F
F
FP
Ciphertext (T)
K1
K2
K3
Ki
K13
K14
K15
K16
PC-2
Key (K)
PC-1
C
D
ROL1
ROL1
ROL2
ROL
ROL2
ROL2
ROL2
ROL1

The F-Function


input (32 bits)
E
48 bits
S1_E S2_E S3_E S4_E S5_E S6_E S7_E S8_E
subkey (48 bits)
S1_K S2_K S3_K S4_K S5_K S6_K S7_K S8_K
S1_I S2_I S3_I S4_I S5_I S6_I S7_I S8_I
S1 S2 S3 S4 S5 S6 S7 S8
S1_O S2_O S3_O S4_O S5_O S6_O S7_O S8_O
P
output (32 bits)

The Initial Permutation (IP)

The following tables describe for each output bit the number of the input bit whose value enters to the output bit. For example, in IP, the 58'th bit in the input becomes the first bit of the output.

IP:

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

FP=IP^{-1}:

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

The P Permutation and the E Expansion

P Permutes the order of 32 bits. E Expands 32 bits to 48 bits by duplicating 16 bits twice.

P:

16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	3	9
19	13	30	6
22	11	4	25

E:

32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

The S Boxes

S box S1:

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

S box S2:

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

The S Boxes (cont.)

S box S3:

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

S box S4:

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

The S Boxes (cont.)

S box S5:

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

S box S6:

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

The S Boxes (cont.)

S box S7:

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

S box S8:

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

The S Boxes (cont.)

How to interpret the S boxes:

The representation of the S boxes use the first and sixth bits of the input as a line index (between 0 and 3), and the four middle bits as the row index (between 0 and 15).

Thus, the input values which correspond to the standard description of the S boxes are

0	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30
1	3	5	7	9	11	13	15	17	19	21	23	25	27	29	31
32	34	36	38	40	42	44	46	48	50	52	54	56	58	60	62
33	35	37	39	41	43	45	47	49	51	53	55	57	59	61	63

The S Boxes (cont.)

Note that **all the operations are linear, except for the S boxes**. Thus, **the strength of DES crucially depends on the choice of the S boxes**.

If the S boxes would be affine, the cipher becomes affine, and thus easily breakable.

The S boxes were chosen with some criteria to prevent attacks.

The Key Scheduling Algorithm

The key scheduling algorithm generates the 16 48-bit subkeys from the 56-bit key, by duplicating each key bit into about 14 of the subkeys in a particular order.

PC-1:

57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36
63	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4

Number of rotations in the key scheduling algorithm:

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

The Key Scheduling Algorithm (cont.)

PC-2:

14	17	11	24	1	5
3	28	15	6	21	10
23	19	12	4	26	8
16	7	27	20	13	2
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

Decryption

Decryption is done by the **same algorithm** as encryption, except that the order of the subkeys is reversed (i.e., K16 is used instead of K1, K15 instead of K2, . . . , and K1 instead of K16.).

Differential Cryptanalysis

The first method which reduced the complexity of attacking DES below (half of) exhaustive search.

Note: In all the following discussion we ignore the existence of the initial and the final permutations, since they do not affect the analysis.

Motivation:

1. All the operations except for the S boxes are linear.
2. Mixing the key in all the rounds prohibits the attacker from knowing which entries of the S boxes are actually used, and thus he cannot know their output.

Differential Cryptanalysis (cont.)

How can we inhibit the key from hiding the information?

The basic idea of differential cryptanalysis: Study the differences between two encryptions of two different plaintexts: P and P^*.

Notation: For any value $\boldsymbol{X}$ during the encryption of P, and the corresponding value $\boldsymbol{X}^*$ during encryption of P^*, denote the difference by $\boldsymbol{X' = X \oplus X^*}$.

Differential Cryptanalysis (cont.)

Advantages: It is easy to predict the output difference of linear operations given the input difference:

- **Unary operations** (E, P, IP):

$$(P(X))' = P(X) \oplus P(X^*) = P(X')$$

- **Binary operations** (XOR):

$$(X \oplus Y)' = (X \oplus Y) \oplus (X^* \oplus Y^*) = X' \oplus Y'$$

- **Mixing the key**:

$$(X \oplus K)' = (X \oplus K) \oplus (X^* \oplus K) = X'$$

We conclude that the differences are linear in linear operations, and in particular, **the result is key independent**.

Differences and the S Boxes

Assume we have two inputs X and X^* for the same S box, and that **we know only their difference X'**.

Denote $Y = S(X)$.

What do we know about Y'?

The simple case: **when $X' = 0$**: $S(X) = S(X^*)$ for any X, and $\mathbf{Y' = 0}$.

If $X' \neq 0$: we do not know the output difference.

Definition: Lets look on the distribution of the pairs (X', Y') of all the possible inputs X. We call the table containing this information **difference distribution table of the S box**.

The Difference Distribution Table of S1

Input	Output XOR															
XOR	0_x	1_x	2_x	3_x	4_x	5_x	6_x	7_x	8_x	9_x	A_x	B_x	C_x	D_x	E_x	F_x
0_x	64	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1_x	0	0	0	6	0	2	4	4	0	10	12	4	10	6	2	4
2_x	0	0	0	8	0	4	4	4	0	6	8	6	12	6	4	2
3_x	14	4	2	2	10	6	4	2	6	4	4	0	2	2	2	0
4_x	0	0	0	6	0	10	10	6	0	4	6	4	2	8	6	2
5_x	4	8	6	2	2	4	4	2	0	4	4	0	12	2	4	6
6_x	0	4	2	4	8	2	6	2	8	4	4	2	4	2	0	12
7_x	2	4	10	4	0	4	8	4	2	4	8	2	2	2	4	4
8_x	0	0	0	12	0	8	8	4	0	6	2	8	8	2	2	4
9_x	10	2	4	0	2	4	6	0	2	2	8	0	10	0	2	12
A_x	0	8	6	2	2	8	6	0	6	4	6	0	4	0	2	10
B_x	2	4	0	10	2	2	4	0	2	6	2	6	6	4	2	12
C_x	0	0	0	8	0	6	6	0	0	6	6	4	6	6	14	2
D_x	6	6	4	8	4	8	2	6	0	6	4	6	0	2	0	2
E_x	0	4	8	8	6	6	4	0	6	6	4	0	0	4	0	8
F_x	2	0	2	4	4	6	4	2	4	8	2	2	2	6	8	8
10_x	0	0	0	0	0	0	2	14	0	6	6	12	4	6	8	6
									⋮							
27_x	10	4	2	0	2	4	2	0	4	8	0	4	8	8	4	4
28_x	12	2	2	8	2	6	12	0	0	2	6	0	4	0	6	2
29_x	4	2	2	10	0	2	4	0	0	14	10	2	4	6	0	4
$2A_x$	4	2	4	6	0	2	8	2	2	14	2	6	2	6	2	2
$2B_x$	12	2	2	2	4	6	6	2	0	2	6	2	6	0	8	4
$2C_x$	4	2	2	4	0	2	10	4	2	2	4	8	8	4	2	6
$2D_x$	6	2	6	2	8	4	4	4	2	4	6	0	8	2	0	6
$2E_x$	6	6	2	2	0	2	4	6	4	0	6	2	12	2	6	4
$2F_x$	2	2	2	2	2	6	8	8	2	4	4	6	8	2	4	2
30_x	0	4	6	0	12	6	2	2	8	2	4	4	6	2	2	4
31_x	4	8	2	10	2	2	2	2	6	0	0	2	2	4	10	8
32_x	4	2	6	4	4	2	2	4	6	6	4	8	2	2	8	0
33_x	4	4	6	2	10	8	4	2	4	0	2	2	4	6	2	4
34_x	0	8	16	6	2	0	0	12	6	0	0	0	0	8	0	6
35_x	2	2	4	0	8	0	0	0	14	4	6	8	0	2	14	0
36_x	2	6	2	2	8	0	2	2	4	2	6	8	6	4	10	0
37_x	2	2	12	4	2	4	4	10	4	4	2	6	0	2	2	4
38_x	0	6	2	2	2	0	2	2	4	6	4	4	4	6	10	10
39_x	6	2	2	4	12	6	4	8	4	0	2	4	2	4	4	0
$3A_x$	6	4	6	4	6	8	0	6	2	2	6	2	2	6	4	0
$3B_x$	2	6	4	0	0	2	4	6	4	6	8	6	4	4	6	2
$3C_x$	0	10	4	0	12	0	4	2	6	0	4	12	4	4	2	0
$3D_x$	0	8	6	2	2	6	0	8	4	4	0	4	0	12	4	4
$3E_x$	4	8	2	2	2	4	4	14	4	2	0	2	0	8	4	4
$3F_x$	4	8	4	2	4	0	2	4	4	2	4	8	8	6	2	2

The Difference Distribution Table of S1 (cont.)

Observe that:

- In the first line $X' = 0$ and thus all the 64 pairs satisfy $Y' = 0$. $Y' \neq 0$ is impossible.
- In the rest of the lines: The average value is 4, the sum in each line is 64. The values are all even in the range 0–16.

  The entries with value 16 mean that for a quarter of the pairs with this input difference X', the output difference is the particular Y'.

  The entries with value 0 mean that there are no pairs with the corresponding input difference X' and the corresponding output difference Y'.

Differences and the S Boxes (cont.)

Definition: If the entry of the input difference X' and the output difference Y' is greater than zero, we say that **X' may cause Y' by the S box**, and denote $X' \to Y'$.

Definition: **The probability of $X' \to Y'$** is the probability that for a pair with the input difference X', the output difference is Y', among all the possible pairs. In DES, the probability is the corresponding value in the difference distribution table divided by 64.

Similarly we define **$X' \to Y'$ by the F-function**, and define the probability as the product of the probabilities by the eight S boxes.

Differences and the S Boxes (cont.)

Differential cryptanalysis uses the entries with large values, and in particular the $0 \rightarrow 0$ entry and the entries with value 16, and other large values.

Observation

Given an input and output differences of an S box, it is possible to list all the pairs with these differences.

Example: For the entry $09_x \rightarrow 1_x$ the 2 pairs are:

1. 33_x, $3A_x$
2. $3A_x$, 33_x

For the entry $01_x \rightarrow F_x$ the 4 pairs are:

1. $1E_x$, $1F_x$
2. $1F_x$, $1E_x$
3. $2A_x$, $2B_x$
4. $2B_x$, $2A_x$

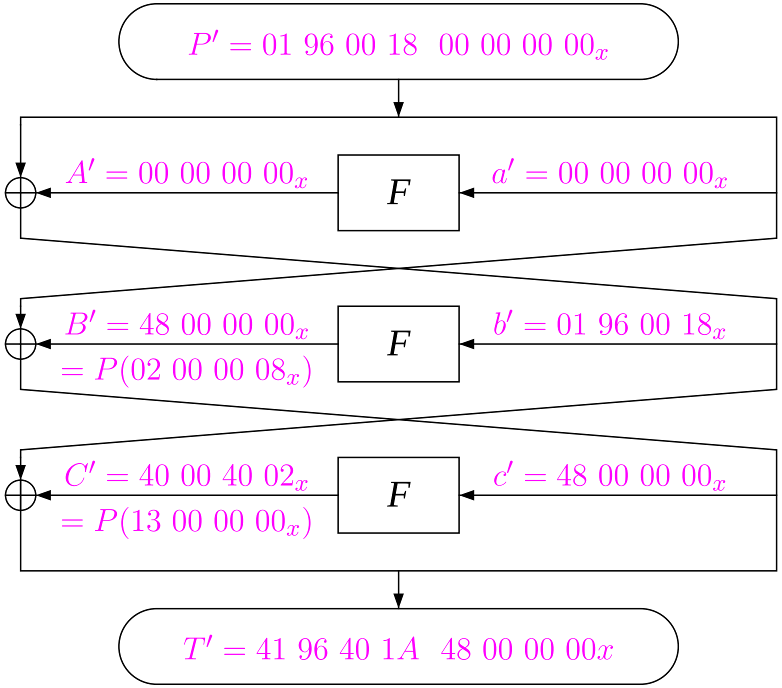
The lists of pairs of all the differences can easily be computed in advance.

Example of a Simple Attack

Assume a 3-round DES, in which for some pair of plaintexts
$P' = 01\ 96\ 00\ 18\ \ 00\ 00\ 00\ 00_x$, and $T' = 41\ 96\ 40\ 1A\ \ 48\ 00\ 00\ 00_x$.
We also assume that $T = 00\ 00\ 00\ 00\ \ 08\ 00\ 00\ 00_x$ and
$T^* = 41\ 96\ 40\ 1A\ \ 40\ 00\ 00\ 00_x$.
(We use the notation T for the ciphertexts, as we use C for the third round intermediate values.)

Example of a Simple Attack (cont.)

Then, the differences in the various rounds are



Example of a Simple Attack (cont.)

We identify that S1 in the third round accepts difference 09_x in the input and outputs difference 1_x in the output. Looking at the difference distribution table, we find only two possible pairs for this combination $((33_x, 3A_x)$ and $(3A_x, 33_x))$.

Thus, we get the following equations:

$$\begin{aligned} S1_E \oplus S1_K &= 33_x \text{ or } 3A_x \\ S1^*_E \oplus S1_K &= 3A_x \text{ or } 33_x. \end{aligned}$$

From the known ciphertexts we know that

$$\begin{aligned} S1_E &= 01_x \\ S1^*_E &= 08_x. \end{aligned}$$

Therefore, we can find two possible values for $S1_K$

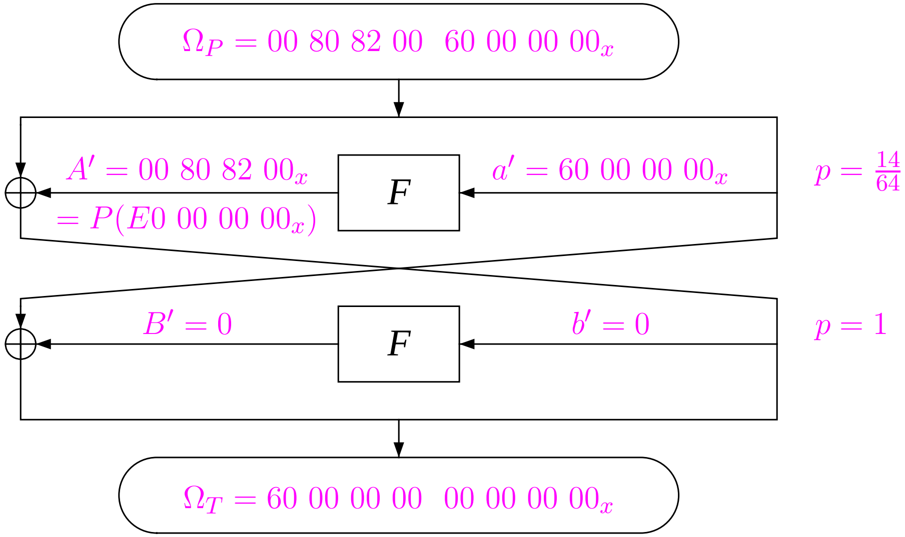
$$S1_K = 32_x \text{ or } 3B_x.$$

(Notice that the difference between these two values is always the input difference, 09_x in this case.)

Characteristics

In differential cryptanalysis we wish to know some statistical information on the differences in intermediate rounds during encryption, given only the plaintext difference.

Example: A **two-round characteristic** with probability $\frac{14}{64}$ (In S1, $0C_x \rightarrow E_x$ with probability $\frac{14}{64}$):



Characteristics (cont.)

Informal Definition: Associated with any pair of encryptions are the XOR value of its two plaintexts, the XOR of its ciphertexts, the XORs of the inputs of each round in the two executions and the XORs of the outputs of each round in the two executions. These XOR values form an ***n*-round characteristic**. A characteristic has a probability, which is the probability that a random pair with the chosen plaintext XOR has the round and ciphertext XORs specified in the characteristic. We denote the plaintext XOR of a characteristic by $\boldsymbol{\Omega_P}$ and its ciphertext XOR by $\boldsymbol{\Omega_T}$.

Characteristics (cont.)

Definition: An ***n*-round characteristic** is a tuple $\boldsymbol{\Omega = (\Omega_P, \Omega_\Lambda, \Omega_T)}$ where Ω_P and Ω_T are m-bit numbers and Ω_Λ is a list of n elements $\Omega_\Lambda = (\Lambda_1, \Lambda_2, \ldots, \Lambda_n)$, each is a pair of the form $\Lambda_i = (\lambda_I^i, \lambda_O^i)$ where λ_I^i and λ_O^i are $m/2$ bit numbers and m is the block size of the cryptosystem. A characteristic satisfies the following requirements:

$$\begin{aligned}
\lambda_I^1 &= \text{the right half of } \Omega_P \\
\lambda_I^2 &= \text{the left half of } \Omega_P \oplus \lambda_O^1 \\
\lambda_I^n &= \text{the right half of } \Omega_T \\
\lambda_I^{n-1} &= \text{the left half of } \Omega_T \oplus \lambda_O^n
\end{aligned}$$

and for every i such that $2 \leq i \leq n-1$:

$$\lambda_O^i = \lambda_I^{i-1} \oplus \lambda_I^{i+1}.$$

Characteristics (cont.)

Definition: **Characteristics can be concatenated** if $\text{swap}(\Omega_T^1) = \Omega_P^2$. The resultant characteristic is

$$\Omega = (\Omega_P^1, \Omega_\Lambda^1 || \Omega_\Lambda^2, \Omega_T^2).$$

Definition: A **right pair** with respect to a characteristic Ω and a key K is a pair P, P^*, which satisfies $P' = \Omega_P$, and all whose differences in the rounds $1, \ldots, n$ are as predicted by the characteristic.

Characteristics (cont.)

Definition: An **independent key** is a list of subkeys which is not necessarily derivable from some key via the key scheduling algorithm.

Probability of a Characteristic

Definition: The **probability** of a characteristic is the probability that a random pair P, P^* which satisfies $P' = \Omega_P$ is a right pair with respect to a random independent key.

Note: The probability of a characteristic is the product of all the probabilities of the S boxes in the characteristic.

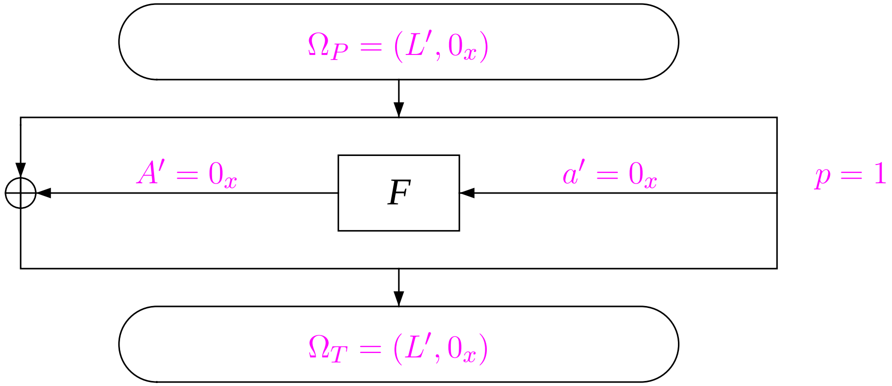
Probability of a Characteristic (cont.)

Note: The probability of characteristics of DES is the probability that any specific pair P, P^* $(P' = \Omega_P)$ is a right pair among all random keys. We are more interested in the probability that for a specific (unknown) key, a random pair P, P^* $(P' = \Omega_P)$ is a right pair. In practice, the first probability is a good approximation of the second probability.

Examples of One-Round Characteristics

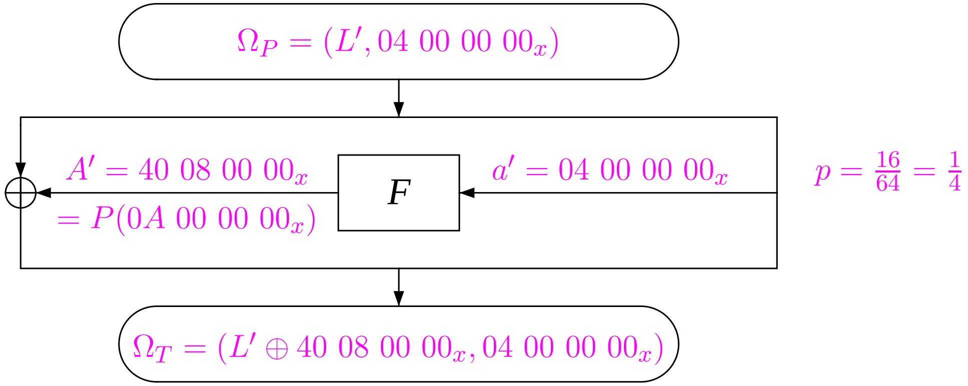
Choose the inputs of the S boxes by the best entries in the difference distribution tables.

Example: An one-round characteristic with probability 1 is (for any L'):



Examples of One-Round Characteristics (cont.)

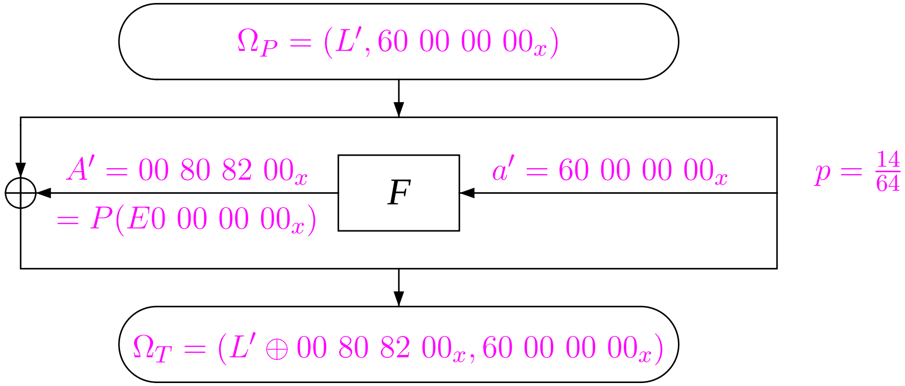
The second best one-round characteristic has probability 1/4, using only one active S box (S2):


$\Omega_P = (L', 04\ 00\ 00\ 00_x)$
$A' = 40\ 08\ 00\ 00_x$
$= P(0A\ 00\ 00\ 00_x)$
F
$a' = 04\ 00\ 00\ 00_x$
$p = \frac{16}{64} = \frac{1}{4}$
$\Omega_T = (L' \oplus 40\ 08\ 00\ 00_x, 04\ 00\ 00\ 00_x)$

There is a similar characteristic using S6.

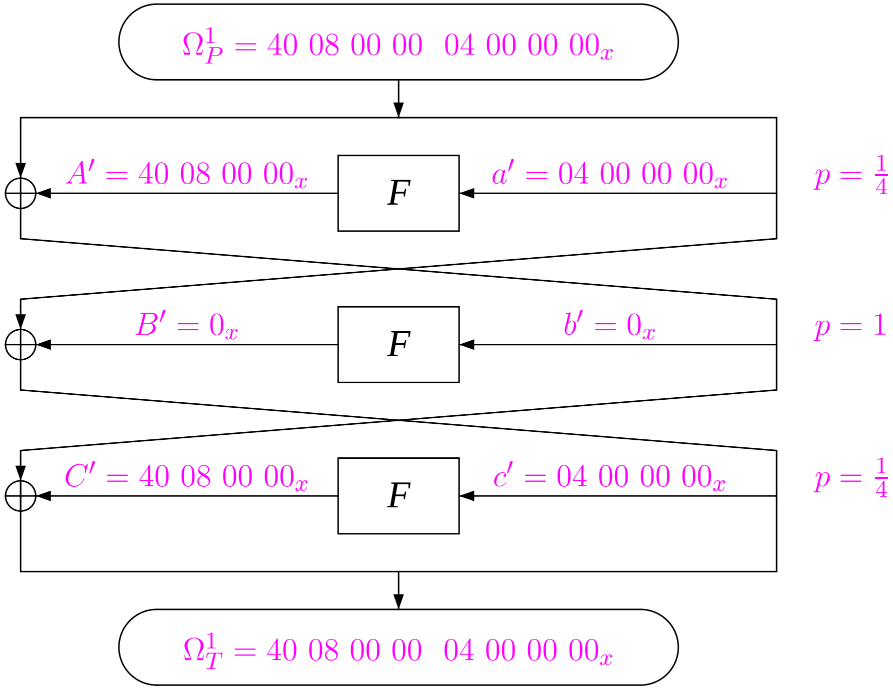
Examples of One-Round Characteristics (cont.)

The next best characteristic has probability $\frac{14}{64}$:


$\Omega_P = (L', 60\ 00\ 00\ 00_x)$
$A' = 00\ 80\ 82\ 00_x$
$= P(E0\ 00\ 00\ 00_x)$
F
$a' = 60\ 00\ 00\ 00_x$
$p = \frac{14}{64}$
$\Omega_T = (L' \oplus 00\ 80\ 82\ 00_x, 60\ 00\ 00\ 00_x)$

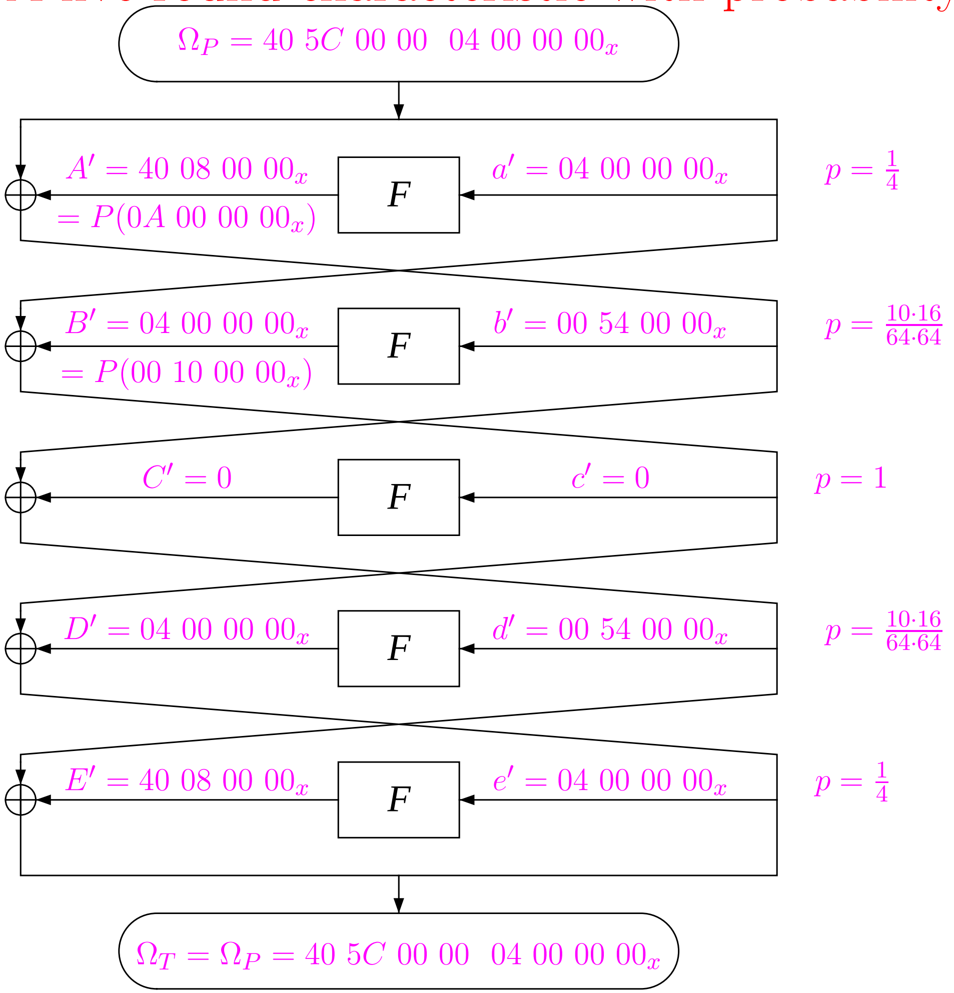
A Three-Round Characteristic

A three-round characteristic with probability 1/16:



A Five-Round Characteristic

A five-round characteristic with probability about 1/10486:



Differential Attacks

The simplest differential attack (0R-attack) breaks ciphers with the same number of rounds as the characteristic. Using 3-round characteristics we can find key bits of 3-round DES, and using 5-round characteristics we can find key bits of 5-round DES.

Differential Attacks (cont.)

The basic algorithm:

1. Choose some $m = 2p^{-1}$ random pairs P, P^* such that $P' = \Omega_P$, and request the corresponding ciphertexts T and T^* under the unknown key K.

2. Choose only the pairs satisfying $T' = \Omega_T$, and discard the others. About $m(p + 2^{-64})$ pairs remain (from the m pairs): mp right pairs and $2^{-64}m$ wrong pairs. If $p \gg 2^{-64}$ we can assume that all the remaining pairs are right pairs.

Differential Attacks (cont.)

3. Each remaining right pair satisfies the difference predictions of the characteristics and its values of T and T^* are known. The differences of the inputs and the outputs of the S boxes of the last round are known from $T' = T \oplus T^*$ (and from the characteristic).

   If the input difference is non-zero, not all the inputs are possible, and only a minority of the inputs satisfy the input and output differences: in each pair only about 0–16 possible values for the 6 input bits of the S box are possible. Each value suggests one value for the 6 corresponding key bits.

   The right value of the 6 key bits must be suggested by all the right pairs, while other values are suggested arbitrarily by only a few of the pairs. By cutting the sets of keys suggested by all the pairs, we receive two possible values for each 6 key bits; in total we receive $2^8 = 256$ possible values for 48 key bits (if all the eight S boxes are active).

   If a wrong pair still remains, still the keys suggested by the largest number of pairs are likely to include the right key.

Success Rate Analysis

Why 2^{-64} of the remaining pairs are wrong?:
Because if the cipher is a random permutation, given any pair of ciphertexts, the probability that their difference is a given value is 2^{-64} (actually $1/2^{64}-1$) independent of the value.

What is the success rate?:
Let the number of active S-boxes in the last round be s. Each right pair suggests 2^s keys for sure (two options for each active S-box). Each active S-box has actually four possible solutions on average. Hence, each right pair suggests 4^s solutions. Moreover, $m \cdot 2^{-64}$ pairs suggest completely random values in the active S-boxes (again 4^s values on average). But if $p \gg 2^{-64}$, we can discard this option.

Probabilities Versus Number of Rounds

The probabilities of the characteristics reduces very fast with the number of rounds:

Number of rounds	Probability
1	1
2	1/4
3	1/16
4	$\approx 1/800$
5	$\approx 1/10000$
6	$\approx 1/1000000$

Probabilities Versus Number of Rounds (cont.)

As the number of rounds is increased, the reduction rate grows. By the table, we may expect that at 9–10 rounds, the probabilities are smaller than 2^{-56} or 2^{-64}.

We are interested in longer characteristics with higher probabilities.

Differentials

Usually differential cryptanalysis use only the Ω_P and Ω_T of the characteristics, but not the intermediate values.

Definition: A **Differential** is a set of all the characteristics with the same Ω_P and Ω_T.

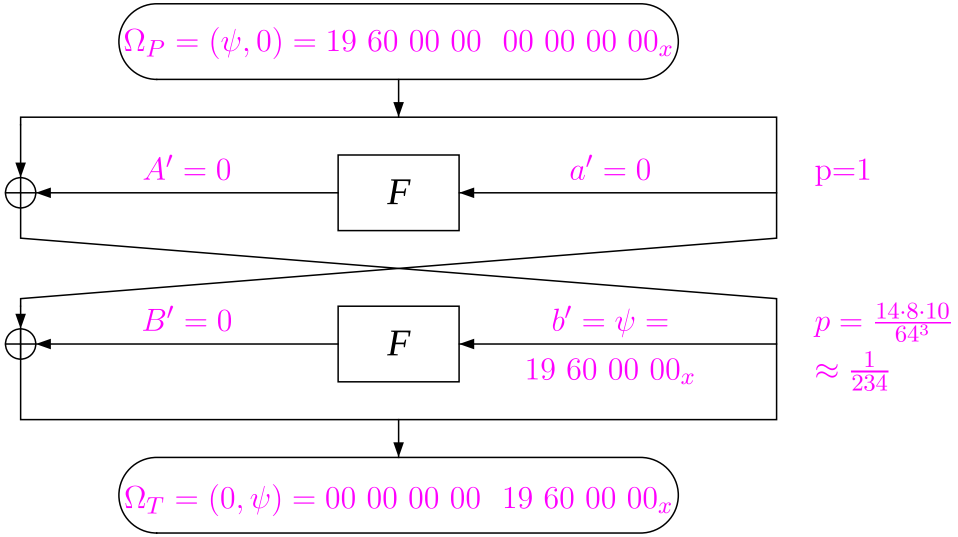
The probability of the differential is the sum of the probabilities of the various characteristics.

In most differential attacks we actually use differentials, rather than characteristics. The probabilities of the characteristics serve as **lower bounds** for the probabilities of the differentials.

Iterative Characteristics

Characteristics which can be concatenated to themselves are called **iterative characteristics**.

The best iterative characteristic of DES is:


$\Omega_P = (\psi, 0) = 19\ 60\ 00\ 00\ \ 00\ 00\ 00\ 00_x$
$A' = 0$
F
$a' = 0$
p=1
$B' = 0$
F
$b' = \psi =$
$19\ 60\ 00\ 00_x$
$p = \frac{14\cdot 8\cdot 10}{64^3}$
$\approx \frac{1}{234}$
$\Omega_T = (0, \psi) = 00\ 00\ 00\ 00\ \ 19\ 60\ 00\ 00_x$

where $\psi = 19\ 60\ 00\ 00_x$. Due to the importance of this iterative characteristic, we call it **the iterative characteristic**.

There is another value $\psi^\dagger = 1B\ 60\ 00\ 00_x$ for which the iterative characteristic has the same probability.

Iterative Characteristics (cont.)

These two characteristics are the best when iterated to seven or more rounds.

Note: In DES, in order to receive the same output of the F-function, two different inputs must differ in the input of at least three S boxes.

Probabilities Versus Number of Rounds

The probability of the iterative characteristic versus the number of rounds:

Number of rounds	Probability
3	$2^{-7.9} \approx 1/234$
5	$2^{-15.7} \approx 1/55000$
7	$2^{-23.6}$
9	$2^{-31.5}$
11	$2^{-39.4}$
13	$2^{-47.2}$
15	$2^{-55.1}$
16	2^{-62}
17	2^{-63}

XOR-Differences in the Presence of Additions

Consider the operation $Z = X + Y$. If $X' = Y' = 0$, then necessarily $Z' = 0$. But when $X' = 1_x, Y' = 0$, there are several possible XOR-differences of Z'. $X' = 1_x$ means that $X = X^* + 1$ or vice versa (we shall continue under the assumption that $X = X^* + 1$). Both are added with $Y = Y^*$, to obtain Z.
If the least significant bit of $Y = Y^*$ is zero, then there is the difference in Z is going to be only in the least significant bit.
When the least significant bit of $Y = Y^*$ is one, there there is going to be carry in $X + Y$ but no carry in $X^* + Y^*$. This means, that the same process is repeated (i.e., if the second least significant bit of $Y = Y^*$ is 0, the *carry chain* ends here, otherwise, there is difference in the carry).

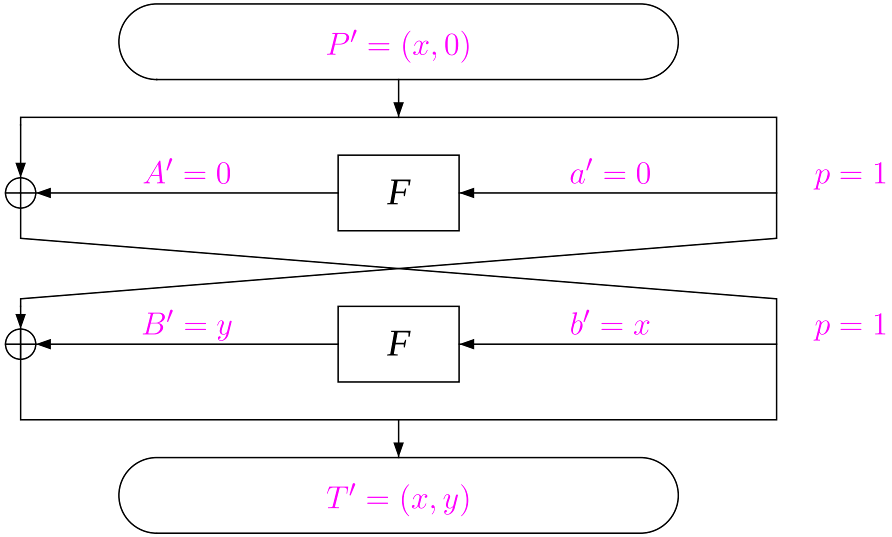
XOR-Differences in the Presence of Additions (cont.)

There is a special bit which cause a very short carry chain. A difference in the most significant bit, *does not* generate a carry chain, as the modular reduction cancels the difference. Hence, when we are dealing with the most significant bit there is no probability associated with it.
If X' has two active bits, the carry chain from the lower bit, can cancel the difference in the more higher order bit. The probability of each carry/no carry decision is $1/2$ (where of course, after no-carry decision, there is no more carries).

If $Y' \neq 0$ as well, one can repeat the previous analysis. Each active bit (either in X' or in Y') may cause a carry (or not cause a carry) with probability $1/2$.

Truncated Differentials

Truncated differential are an extension of differential cryptanalysis where the difference is not fully specified. For example, consider the following 2-round truncated differential:



Truncated Differentials (cont.)

Using truncated differentials in differential attacks is similar to the use of regular differentials. There are two small differences:

1. The probability that a wrong pair looks as if it is a correct one is $S \cdot 2^{-64}$, where S is the number of possible differences (in the example above, x and y can be any value, and thus, $S = 2^{64}$).

2. In differential cryptanalysis, the probability of the differential is independent of the direction (encryption/decryption). In the case of truncated differentials, this is not the case. For example, inverting the order of the rounds in the above example yields a truncated differential with probability 2^{-32}.

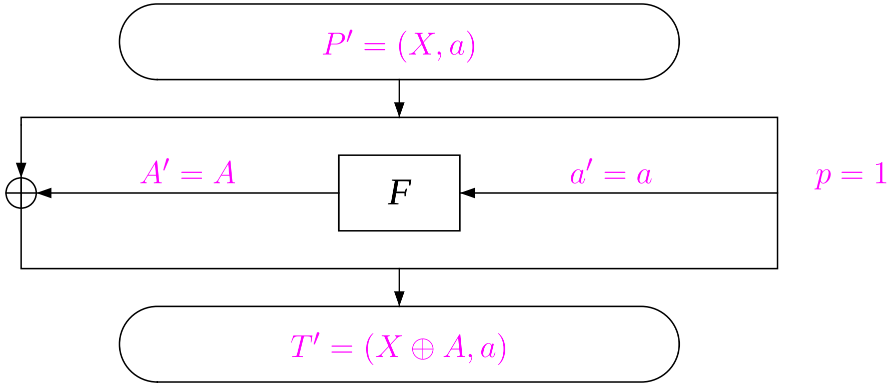
Some Caveats

Usually truncated differentials are really useful to handle. The reason for that is that transitions of the form $a \rightarrow b$ can be approximated with the probability 2^{-w} (for $a \neq 0$, $|b| = w$), independent of a and b.

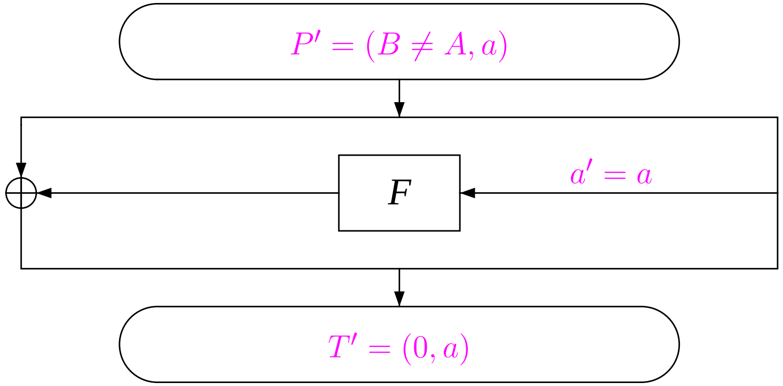
Of course, this is under the assumption that a may cause difference b. If the round function is bijective, and $a \neq 0$ then b cannot be 0.

Some Caveats (cont.)

Let us assume that for a specific a, A:

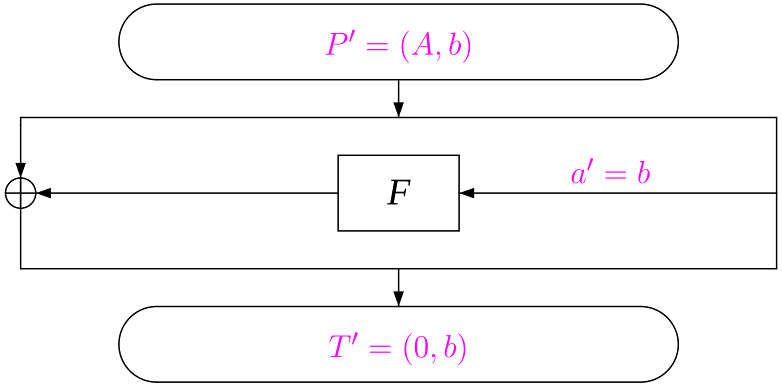


Then, the probability of following truncated differential is 0:



Some Caveats (cont.)

The following truncate differential may also have probability 0:



Motivation

Consider a collision in a hash function. There are two messages M_1, M_2 for which $h(M_1) = h(M_2)$. If $h(\cdot)$ is a Merkle-Damgård hash function, then this collision also exists somewhere in the compression function, i.e., there are two sets of inputs to the compression function, cv_1, m_1 and cv_2, m_2, for which $F(cv_1, m_1) = F(cv_2, m_2)$.

In other words, for F, we can define the differential $(\Delta cv, \Delta m) \rightarrow 0$.

Differential Characteristics of Compression Functions

To differentially attack compression functions, one needs to first find a suitable differential characteristic. For example, in collision producing attacks, one aims to find a characteristics that predicts a zero difference after the feed forward operation.

If we consider a case where we have one block collision, then the differential characteristic is of the form $(\Delta cv = 0, \Delta m) \rightarrow 0$.

As most compression functions are block ciphers in disguise, it seems to be the same process for finding and suggesting differential characteristics. However, in most of these compression functions, the message blocks are used as the key of the block cipher (Davies-Meyer construction). This means that we are restricted to a very special class of "related-key" differential characteristics.

Differential Characteristics of Compression Functions (cont.)

There is a huge advantage in attacking compression functions over attacks directly targeted at the block cipher. The adversary has a much greater knowledge about what is going on inside the primitive. This means that the adversary can very quickly know if the characteristic is followed, and to what extent.

On the other hand, when discussing a collision-finding characteristics, we are restricted by a lower probability bound. While an attack on a block cipher with n-bit blocks can use differentials with probability of $c \cdot 2^{-n}$, for $c > 1$, attacks on hash functions with digest size of n bits cannot exploit differentials with probability lower than $2^{-n/2}$.