MasterMath Cryptology 2015 - 2/2 - Cryptanalysis

Wednesday, 8 April, 2015 10:38

9. Differential cryptanalysis (v2)

9.1. Differential cryptanalysis

In differential analysis we simultaneously consider two encryptions $C= E_K(P)$ and $C' = E_K(P')$ and look at all differences between their computation.
For any variable X related to (P,C), we denote by X' the corresponding variable related to (P',C') and denote $\Delta X = X \oplus X'$.
A key property we will be using is that key additions cancel out:

Let $C = P \oplus K, C' = P' \oplus K$, then $\Delta C = C' \oplus C = P \oplus K \oplus P' \oplus K = P \oplus P' = \Delta P$

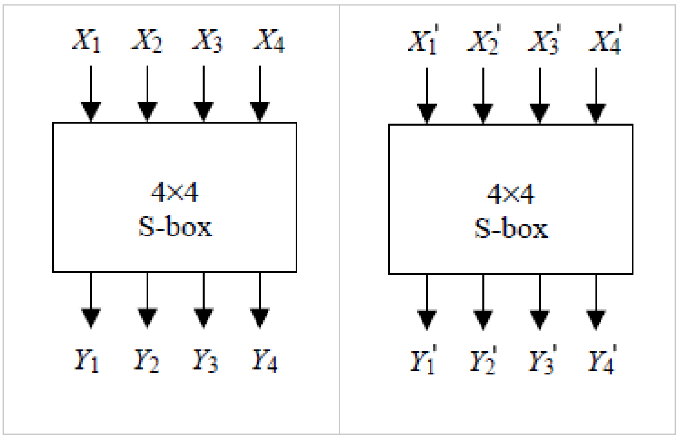
Consider input difference $\Delta X = (\Delta X_1, \dots, \Delta X_n)$ and output difference $\Delta Y = (\Delta Y_1, \dots, \Delta Y_n)$ then we call the pair $(\Delta\mathrm{X}, \Delta\mathrm{Y})$ a *differential.*
For an ideally randomizing cipher, the probability that ΔY occurs given that ΔX holds is 2^{-n}.
For differential cryptanalysis instead of the probability bias we will use the probability directly:

$$p_{\Delta X, \Delta Y} = \Pr[\Delta Y | \Delta X]$$

Differential cryptanalysis tries to exploit high probability occurrances of certain input and output differences of the Sboxes.
Very similar to linear cryptanalysis we try to construct a differential relating plaintext differences to differences in the output of the second-last round that has a sufficiently high probability.
Similar to linear cryptanalysis, we can then again try final round subkey values and check for which fraction of (plaintext,ciphertext,plaintext',ciphertext')-pairs the differential hold to distinguish the correct final round subkey value.
The attack is a chosen-plaintext attack as we need to be able to query encryptions of P and $P' \oplus \Delta P$.

9.2. S-BOX Difference Distribution Table (DDT)

Let X_1, X_2, X_3, X_4 be random variables for the input bits assumed to be independent and uniformly random and let Y_1, Y_2, Y_3, Y_4 be random variables for the output bits. Similarly, X_1', X_2', X_3', X_4' and Y_1', Y_2', Y_3', Y_4' for the second instance.



We are interested in differentials for the SBOX of the form

$(\Delta X_1 \Delta X_2 \Delta X_3 \Delta X_4, \Delta Y_1 \Delta Y_2 \Delta Y_3 \Delta Y_4)$

There are $2^4 = 16$ different values for $X_1X_2X_3X_4$ all equally likely (by assumption).
Hence the probability that such a differential holds can be determined by counting the number of $X_1X_2X_3X_4$ values such that $(X_1X_2X_3X_4, Y_1Y_2Y_3Y_4) \oplus (X_1'X_2'X_3'X_4', Y_1'Y_2'Y_3'Y_4')$ satisfies the differential, divided by 2^{16}.
Here the Y_i's are determined through the Sbox with input $X_1X_2X_3X_4$, as well as the Y_i''s are determined through the Sbox with input $X_1'X_2'X_3'X_4' = X_1X_2X_3X_4 \oplus \Delta X_1 \Delta X_2 \Delta X_3 \Delta X_4$.

E.g., consider the output differences for input differences $\Delta X_1 \Delta X_2 \Delta X_3 \Delta X_4 = 1011$, $\Delta X_1 \Delta X_2 \Delta X_3 \Delta X_4 = 1000$ and $\Delta X_1 \Delta X_2 \Delta X_3 \Delta X_4 = 0100$:

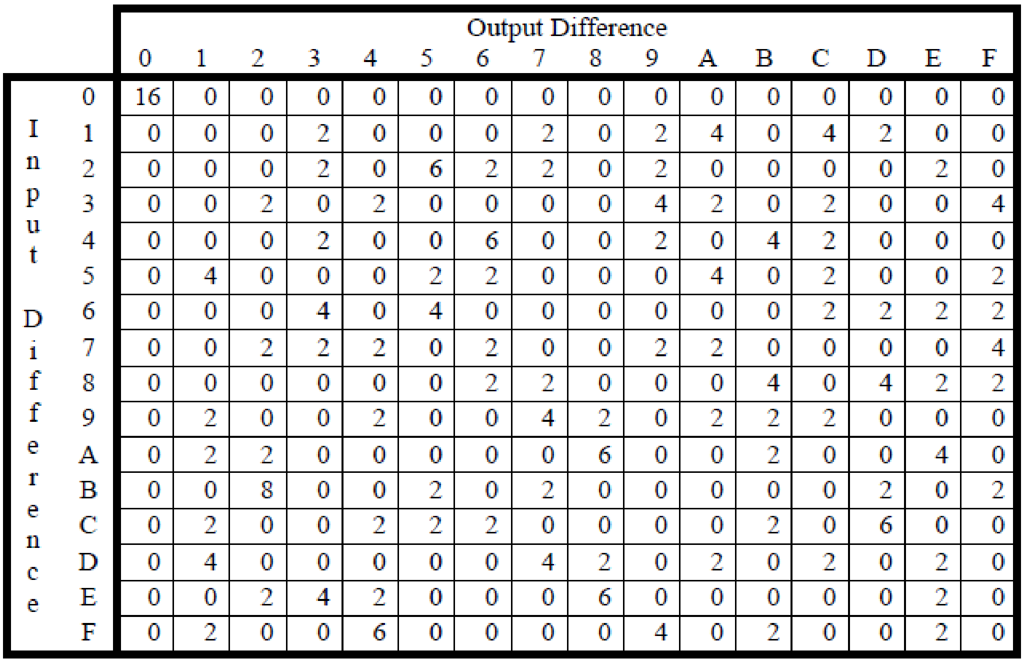
$X_1X_2X_3X_4$	$Y_1Y_2Y_3Y_4$	$\Delta Y \mid \Delta X = 1011$	$\Delta Y \mid \Delta X = 1000$	$\Delta Y \mid \Delta X = 0100$
0000	1110	0010	1101	1100
0001	0100	0010	1110	1011
0010	1101	0111	0101	0110

0011	0001	0010	1011	1001
0100	0010	0101	0111	1100
0101	1111	1111	0110	1011
0110	1011	0010	1011	0110
0111	1000	1101	1111	1001
1000	0011	0010	1101	0110
1001	1010	0111	1110	0011
1010	0110	0010	0101	0110
1011	1100	0010	1011	1011
1100	0101	1101	0111	0110
1101	1001	0010	0110	0011
1110	0000	1111	1011	0110
1111	0111	0101	1111	1011

Note that $p_{1011,0010} = \frac{8}{16}$, $p_{1000,1011} = \frac{4}{16}$ and $p_{0100,0110} = \frac{6}{16}$.
An ideal Sbox would have probability $\frac{1}{16}$ for every differential which is impossible, so a good Sbox comes as close as possible.

We can describe the probability for all possible differentials in the *Difference Distribution Table (DDT)*,
which is a table whose rows (respectively columns) describe the possible input (respectively output) difference.
The cell at row I and column O contains the number of matches between the input difference and output difference:

$$SBox(X) \oplus Sbox(X \oplus \Delta X) = \Delta Y$$



		Output Difference															
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Input Difference	0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	1	0	0	0	2	0	0	0	2	0	2	4	0	4	2	0	0
	2	0	0	0	2	0	6	2	2	0	2	0	0	0	0	2	0
	3	0	0	2	0	2	0	0	0	0	4	2	0	2	0	0	4
	4	0	0	0	2	0	0	6	0	0	2	0	4	2	0	0	0
	5	0	4	0	0	0	2	2	0	0	0	4	0	2	0	0	2
	6	0	0	0	4	0	4	0	0	0	0	0	0	2	2	2	2
	7	0	0	2	2	2	0	2	0	0	2	2	0	0	0	0	4
	8	0	0	0	0	0	0	2	2	0	0	0	4	0	4	2	2
	9	0	2	0	0	2	0	0	4	2	0	2	2	2	0	0	0
	A	0	2	2	0	0	0	0	0	6	0	0	2	0	0	4	0
	B	0	0	8	0	0	2	0	2	0	0	0	0	0	2	0	2
	C	0	2	0	0	2	2	2	0	0	0	0	2	0	6	0	0
	D	0	4	0	0	0	0	0	4	2	0	2	0	2	0	2	0
	E	0	0	2	4	2	0	0	0	6	0	0	0	0	0	2	0
	F	0	2	0	0	6	0	0	0	0	4	0	2	0	0	2	0

Difference distribution table: input and output differences are described in hexidecimal, e.g.,

$\Delta X_1 \Delta X_2 \Delta X_3 \Delta X_4 = 0110 \rightarrow 0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 6,$

$\Delta Y_1 \Delta Y_2 \Delta Y_3 \Delta Y_4 = 1011 \rightarrow 11 = 0xB.$

Each table cell contains the count of inputs for which the output difference occurs given the input difference.
All table cells are even as differences are symmetric: $X' = X \oplus \Delta X$ and $X = X' \oplus \Delta X$.
Note that the sum of every row and column is 16.

Such a difference distribution table can be easily computed using SAGE:

```
sage: S=mq.SBox(14,4,13,1,2,15,11,8,3,10,6,12,5,9,0,7);
sage: S.difference_distribution_matrix()
[16  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
[ 0  0  0  2  0  0  0  2  0  2  4  0  4  2  0  0]
[ 0  0  0  2  0  6  2  2  0  2  0  0  0  0  2  0]
[ 0  0  2  0  2  0  0  0  0  4  2  0  2  0  0  4]
[ 0  0  0  2  0  0  6  0  0  2  0  4  2  0  0  0]
[ 0  4  0  0  0  2  2  0  0  0  4  0  2  0  0  2]
[ 0  0  0  4  0  4  0  0  0  0  0  0  2  2  2  2]
[ 0  0  2  2  2  0  2  0  0  2  2  0  0  0  0  4]
[ 0  0  0  0  0  0  2  2  0  0  0  4  0  4  2  2]
[ 0  2  0  0  2  0  0  4  2  0  2  2  2  0  0  0]
[ 0  2  2  0  0  0  0  0  6  0  0  2  0  0  4  0]
[ 0  0  8  0  0  2  0  2  0  0  0  0  0  2  0  2]
[ 0  2  0  0  2  2  2  0  0  0  0  2  0  6  0  0]
[ 0  4  0  0  0  0  0  4  2  0  2  0  2  0  2  0]
[ 0  0  2  4  2  0  0  0  6  0  0  0  0  0  2  0]
[ 0  2  0  0  6  0  0  0  0  4  0  2  0  0  2  0]
```

9.3. Piling-Up Differentials

Piling up differentials is very similar to piling-up linear relations, but instead of combining probability biases we can multiply probabilities directly.
However, differentials cannot be arbitrarily combined as they can contradict each other, whereas linear relations can simply be added to each other.
So to simplify notation we will construct 1-round differentials over the entire state from differentials over Sboxes in the same round, and we will construct (r+s)-round differentials by combining a r-round differential and a s-round differential.
One can also take a similar view for linear cryptanalysis.

9.3.1. Constructing 1-round differentials

Let $(\Delta X_{1,2,3,4}, \Delta Y_{1,2,3,4})$, $(\Delta X_{5,6,7,8}, \Delta Y_{5,6,7,8})$, $(\Delta X_{9,10,11,12}, \Delta Y_{9,10,11,12})$ and $(\Delta X_{13,14,15,16}, \Delta Y_{13,14,15,16})$ be differentials for Sboxes $S_{r1}, S_{r2}, S_{r3}, S_{r4}$ respectively for some round r with probabilities $p_{r1}, p_{r2}, p_{r3}, p_{r4}$ respectively.
We combine these to obtain a differential over the entire substitution step:

$$(\Delta X_1 \ldots \Delta X_{16}, \Delta Y_1 \ldots, \Delta Y_{16}) \text{ with probability } p_{subst} = p_{r1}p_{r2}p_{r3}p_{r4}$$

Let I_r and O_r be the input and output state of round r

then $X_i = I_{r,i} \oplus K_{r,i}$ thus $\Delta I_{r,i} = \Delta X_i$ for $i = 1, \ldots, 16$

and $O_{r,\pi_P(i)} = Y_i$ or $O_{r,j} = Y_{\pi_P^{-1}(j)}$ for $i = 1, \ldots, 16, j = 1, \ldots, 16$

Therefore:

$$\left(\Delta X_1 \ldots \Delta X_{16}, \Delta Y_{\pi_P^{-1}(1)} \ldots \Delta Y_{\pi_P^{-1}(16)}\right) = (\Delta X_1 \ldots \Delta X_{16}, \quad \Delta Y_1 \Delta Y_5 \Delta Y_9 \Delta Y_{13} \ldots \Delta \mathrm{Y}_{12} \Delta Y_{16})$$

is a 1-round differential with probability p_{subst}

9.3.2. Concatenating a r-round differential and a s-round differential

Let $(\Delta I_1, \Delta O_1)$ be a r-round differential with probability p_1.
Let $(\Delta I_2, \Delta O_2)$ be a s-round differential with probability p_2.
If $\Delta O_1 = \Delta I_2$ then

$(\Delta I_1, \Delta O_2)$ is a $(r+s)$-round differential with probability $p = p_1 p_2$.

9.4. Constructing a 3-round differential for the toy cipher

Let I_r and O_r be the input and output state of round r, and X_r and Y_r the state before and after the substitution step of round r.
We start with a differential over Sbox S_{12}: $\left(\Delta X_{1,\{5,6,7,8\}}, \Delta Y_{1,\{5,6,7,8\}}\right) = (1011{,}0010)$ which has probability 8/16 (lookup row B=1011 column 2=0010 in DDT).
With zero differences for the other Sboxes in round 1, this translates to a differential

$$(\Delta P, \Delta Y_1) = (\Delta X_1, \Delta Y_1) = (0000\ 1011\ 0000\ 0000, 0000\ 0010\ 0000\ 0000) \quad p = 1/2$$

and we get the round 1 differential by apply the permutation on ΔY_1:

$$(\Delta P, \Delta O_1) = (0000\ 1011\ 0000\ 0000, 0000\ 0000\ 0100\ 0000) \quad p = 1/2$$

For round 2 we only have a non-zero difference for Sbox S_{23} and we use differential

$$\left(\Delta X_{2,\{9,10,11,12\}}, \Delta Y_{2,\{9,10,11,12\}}\right) = (0100, 0110) \quad \text{with prob. } 6/16 \text{ (DDT row 4 col 6)}$$

This leads to

$$(\Delta X_2, \Delta Y_2) = (0000\ 0000\ 0100\ 0000, 0000\ 0000\ 0110\ 0000) \quad p = 3/8$$

and round-2 differential

$$(\Delta I_2, \Delta O_2) = (0000\ 0000\ 0100\ 0000, 0000\ 0010\ 0010\ 0000) \quad p = 3/8$$

Combining the round-1 differential and round-2 differential we get

$$(\Delta P, \Delta O_2) = (0000\ 1011\ 0000\ 0000, 0000\ 0010\ 0010\ 0000) \quad p = 3/16$$

For round 3 we have a non-zero difference for Sboxes S_{32} and S_{33}, we'll use the same differential for both with probability 6/16:

$\left(\Delta X_{3,\{5,6,7,8\}}, \Delta Y_{3,\{5,6,7,8\}}\right) = (0010{,}0101) \quad p = 3/8$

$\left(\Delta X_{3,\{9,10,11,12\}}, \Delta Y_{3,\{9,10,11,12\}}\right) = (0010{,}0101) \quad p = 3/8$

This leads to

$(\Delta X_3, \Delta Y_3) = (0000\ 0010\ 0010\ 0000, 0000\ 0101\ 0101\ 0000) \quad p = 9/64$

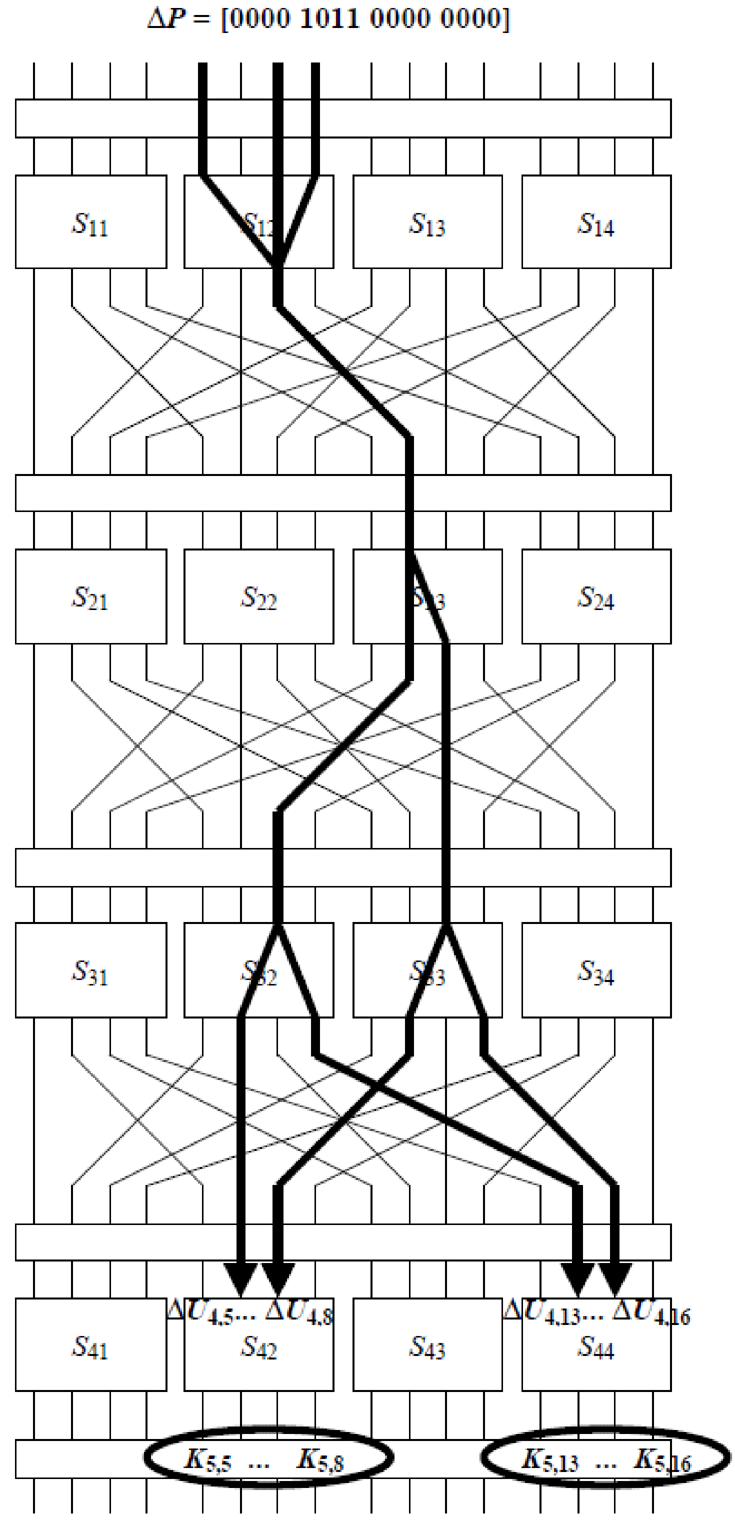
and round-3 differential

$(\Delta I_3, \Delta O_3) = (0000\ 0010\ 0010\ 0000, 0000\ 0110\ 0000\ 0110) \quad p = 9/64$

Combining the round-{1,2} differential and round-3 differential we get

$(\Delta P, \Delta O_3) = (0000\ 1011\ 0000\ 0000,\ \ 0000\ 0110\ 0000\ 0110) \quad p = 27/1024$

which is a 3-round differential.



Graphical representation of the 3-round differential

9.5. Extracting final round key bits

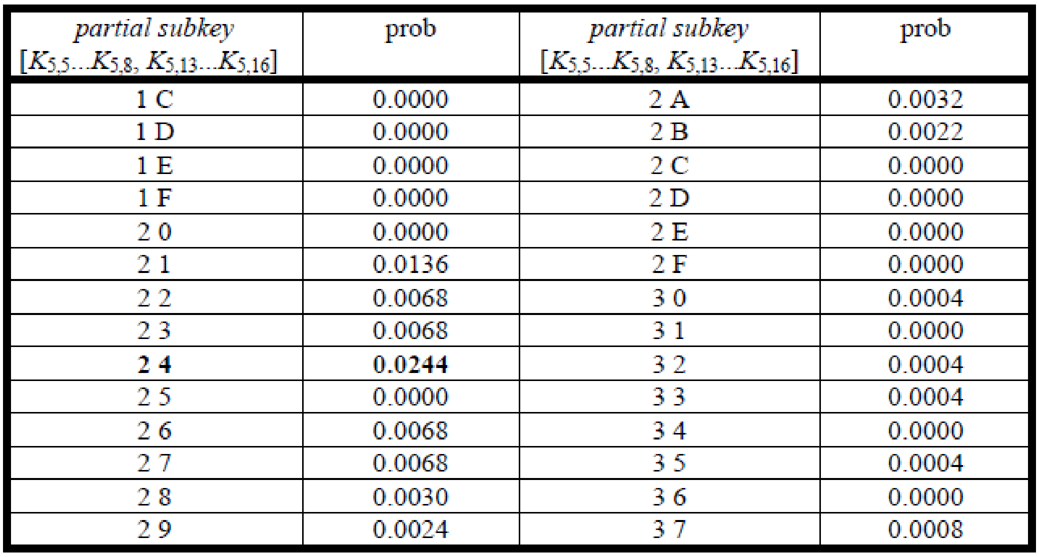
The attack procedure is very similar to that for linear cryptanalysis.
Once we have a differential over all but the last round for a given cipher that has a suitably large enough probability, we can try to exploit it to recover some bits of the final round key bits from the final key-mixing.
The differential we obtained above

$$(\Delta P, \Delta O_3) = (0000\ 1011\ 0000\ 0000,\ \ 0000\ 0110\ 0000\ 0110) \quad p = 27/1024$$

can be trivially extended with the key-mixing with $K4$:

$$(\Delta P, \Delta X_4) = (0000\ 1011\ 0000\ 0000,\ \ 0000\ 0110\ 0000\ 0110) \quad p = 27/1024$$

We will assume that we have access to a lot of tuples (plaintext P, plaintext $P' = P \oplus \Delta P$, ciphertext $C = E_k(P)$, ciphertext $C' = E_k(P')$) that were obtained in a chosen-plaintext attack scenario.
To be able to check whether the differential holds we need to guess the key bits corresponding to the output bits of the two active Sboxes S_{42} and S_{44} in round 4: $K_{5,5}, K_{5,6}, K_{5,7}, K_{5,8}, K_{5,13}, K_{5,14}, K_{5,15}, K_{5,16}$, we call these bits the final round *subkey*.
The idea is to guess the final round subkey bits and that we hope that for the correct guess the measured differential probability is what's expected, whereas for incorrect guesses we hope to see a measured differential probability very close to 2^{-16}.



partial subkey $[K_{5,5} \ldots K_{5,8}, K_{5,13} \ldots K_{5,16}]$	prob	*partial subkey* $[K_{5,5} \ldots K_{5,8}, K_{5,13} \ldots K_{5,16}]$	prob
1 C	0.0000	2 A	0.0032
1 D	0.0000	2 B	0.0022
1 E	0.0000	2 C	0.0000
1 F	0.0000	2 D	0.0000
2 0	0.0000	2 E	0.0000
2 1	0.0136	2 F	0.0000
2 2	0.0068	3 0	0.0004
2 3	0.0068	3 1	0.0000
2 4	**0.0244**	3 2	0.0004
2 5	0.0000	3 3	0.0004
2 6	0.0068	3 4	0.0000
2 7	0.0068	3 5	0.0004
2 8	0.0030	3 6	0.0000
2 9	0.0024	3 7	0.0008

Experimental results for a differential attack with 5000 (plaintext,ciphertext,plaintext',ciphertext')-tuples.
It shows partial subkey guesses and the determined $prob = count/5000$,
subkeys are written down in hexidecimal.
The correct subkey 24 is listed in bold and has the highest bias magnitude not only in the showed listing, but among all guesses for the subkey.

For differential cryptanalysis the number of required (plaintext,ciphertext,plaintext',ciphertext')-tuples for the attack is proportional to p^{-1}, where p is the probability of the differential over $R-1$ rounds.
In practice, it is generally reasonable to use a small multiple of p^{-1} tuples.

Once we have determined the correct value for the final round subkey bits (or at least a short list of highly-likely values that we can go through), we can continue in the following manner identical as for linear cryptanalysis:

1. Try to exploit other differentials in order to obtain all final round key bits,
   guess all remaining unknown final round key bits.
2. Strip the last round of all known (plaintext,ciphertext)-pairs,
   i.e., revert the last cipheroperations till the second-last key-mixing.
3. One can view the cipher as having $R-1$ rounds, continue to attack the cipher using a differential over $R-2$ rounds.
4. Iterate 1-3 until all round keys have been broken.

9.6. Assumption of independence and using multiple differentials

For differential cryptanalysis we make the same kind of assumption of independence between the Sbox differentials as between the linear relations in linear cryptanalysis.
Hence, for differential cryptanalysis we hope that the predicted differential probability is a very good approximation for the real bias.
Also, there may exist several ways to construct the same 3-round differential whose probabilities add up as they are so-called *disjoint probability events*.

E.g., consider just two sbox differentials with the same input difference $(\Delta X, \Delta Y)$ and $(\Delta X, \Delta Z)$ with probabilities p_1 and p_2, now depending on the bits $X_1 X_2 X_3 X_4$ we'll see output difference either ΔY or ΔZ or some other difference, but you can't have two different output differences simultaneously.

9.7. Truncated differential cryptanalysis

The fact that probabilities of differentials with the same input difference can be added, is used in so-called *truncated differential cryptanalysis* where a characteristic is defined over a set of input differences and a set of output differences.
E.g., the following Sbox truncated differential $(\{3,7,E\},\{2,4\})$ has probability 4/16: given input difference 3, 7 or E, one obtains output difference in the set {2,4} with probability 4/16.
This is because differentials 32, 34, 72, 74, E2, E4 all have probability 2/16, so, e.g., for input difference E the probability of an output difference in the set {2,4} is 2/16+2/16=4/16.
As you can see using truncated differential cryptanalysis we can obtain significantly higher differential probabilities.

9.8. Impossible differential cryptanalysis

Another advanced form of differential cryptanalysis is impossible differential cryptanalysis (http://link.springer.com/chapter/10.1007/3-540-48910-X_2)
In impossible differential cryptanalysis one tries to exploit a differential $(\Delta P, \Delta O_3)$ that has probability zero, i.e., a 3-round differential for which there exist no construction that leads to a non-zero predicted probability.
Thus it does not suffice to consider only one non-zero probability construction for $(\Delta P, \Delta O_3)$, but all constructions that lead to $(\Delta P, \Delta O_3)$ must have predicted probability zero.
An example impossible differential is

$$(\Delta P, \Delta O_3) = (1000\ 0000\ 0000\ 0000, 1000\ 0000\ 0000\ 0000)$$

One way to construct is to simply concatenate the following 1-round differential 3 times:

$$(\Delta I, \Delta O) = (1000\ 0000\ 0000\ 0000, 1000\ 0000\ 0000\ 0000)$$

This 1-round differential has probability 0 as for the only active Sbox S_{r1} table cell 88 of the DDT is 0.

But we can actually prove there exist no construction with non-zero probability that leads to the above 3-round differential.
First off, we will exclude all constructions that use differentials of the form x0 or 0x (either input or output difference zero) with non-zero x as they have probability zero anyway.
Then note that in round 1 the only active Sbox is Sbox S_{11} and that its output bits are all mapped to the first input bit of round 2 Sboxes $S_{21}, S_{22}, S_{23}, S_{24}$.
This implies that the output mask determines which round 2 Sboxes are active and that the active round 2 Sboxes have input difference 8 (difference only in the first bit).
E.g., consider Sbox differential 8B=(1000,1011) for S_{11}, then output difference B (=1011) implies that S_{21}, S_{23}, S_{24} are active and have input difference 8 (=1000).
On the other hand, in round 3 the only active Sbox is Sbox S_{31} and its input bits are all mapped to the first output bit of the round 2 Sboxes $S_{21}, S_{22}, S_{23}, S_{24}$.
This implies that in order to obtain $\Delta O_3 = 1000\ 0000\ 0000\ 0000$, the output bits of sboxes $S_{21}, S_{22}, S_{23}, S_{24}$ that map to the other Sboxes S_{32}, S_{33}, S_{34} should all have difference zero.
In other words, the active round 2 sboxes should have output difference 8=(1000).
To conclude, the active round 2 sboxes must have input difference 8 and output difference 8, but sbox characteristic 88 has probability 0, so this will always lead to a zero-probability characteristic.

To exploit an impossible differential we can use a procedure very similar to the one from section 9.5 with the following modification:

- We guess the entire key K_5
- For every key guess K_5, we go over all (plaintext,plaintext',ciphertext,ciphertext')-tuples with plaintext difference
  $\Delta P = 1000\ 0000\ 0000\ 0000$
  we partially decrypt the ciphertexts with K_5 and determine ΔO_3
  as soon as we find a tuple for which $\Delta O_3 = 1000\ 0000\ 0000\ 0000$ we can cross off this key guess.

Note that even for incorrect K_5 guesses the probility of running into the target ΔO_3 is 2^{-16}, so we can expect many key guesses left that pass this filter.
So in order to have only the correct key guess remaining, we'll need many more impossible differentials and use each one to filter out bad key guesses.

9.9. Boomerang distinguishers

This attack by David Wagner (http://link.springer.com/chapter/10.1007%2F3-540-48519-8_12) effectively doubles the range of differentials and was designed as a distinguisher attack (remember: an attack that distinguishes a cipher with a random key from a random permutation).
The idea is to split the cipher into 2 parts (e.g., rounds 1&2 and rounds 3&4) and determine a high probability differential over

each part independently and then to find a quartet of plaintext-ciphertext pairs that satisfies these differentials.
Thus we'll have a differential over say rounds 1&2: $(\Delta P, \Delta O_2)$ with probability p_1 and a differential over the remaining rounds: $(\Delta I_3, \Delta C)$ with probability p_2 where $\Delta C = \Delta O_4$ is the ciphertext difference.

For the toy cipher we can use almost the same differential for both parts, where the only difference is caused due to the fact that there is no permutation step in the final round.
We use differential $B2$ for Sbox S_{13} and differential 25 for Sbox S_{23}:

$(\Delta P, \Delta O_1) = (0000\ 0000\ 1011\ 0000, 0000\ 0000\ 0010\ 0000)$ with probability 1/2
$(\Delta I_2, \Delta O_2) = (0000\ 0000\ 0010\ 0000, 0000\ 0010\ 0000\ 0010)$ with probability 3/8

This combines to:

$(\Delta P, \Delta O_2) = (0000\ 0000\ 1011\ 0000, 0000\ 0010\ 0000\ 0010)$ with probability 3/16

Also, we use differential $B2$ for Sbox S_{33} and differential 25 for Sbox S_{43}:

$(\Delta I_3, \Delta O_3) = (0000\ 0000\ 1011\ 0000, 0000\ 0000\ 0010\ 0000)$ with probability 1/2
$(\Delta I_4, \Delta \mathrm{C}) = (0000\ 0000\ 0010\ 0000, 0000\ 0000\ 0101\ 0000)$ with probability 3/8

This combines to:

$(\Delta I_3, \Delta C) = (0000\ 0000\ 1011\ 0000, 0000\ 0000\ 0101\ 0000)$ with probability 3/16

Given the two seperate differentials we try to find a quartet of plaintext-ciphertext pairs $\big((P_1, C_1), (P_2, C_2), (P_3, C_3), (P_4, C_4)\big)$ that satisfies the following properties:

- $P_1 \oplus P_2 = \Delta P$
- $P_3 \oplus P_4 = \Delta P$
- $C_1 \oplus C_3 = \Delta C$
- $C_2 \oplus C_4 = \Delta C$

This can be done in the following manner:

1. Select P_1 at random, determine $P_2 = P_1 \oplus \Delta P$
2. Ask to encrypt P_1 and P_2: $C_1 = E_K(P_1), C_2 = E_K(P_2)$
3. Determine $C_3 = C_1 \oplus \Delta C, C_4 = C_2 \oplus \Delta C$
4. Ask to decrypt C_3 and C_4: $P_3 = E_K^{-1}(C_3), P_4 = E_K^{-1}(C_4)$
5. If $P_3 \oplus P_4 \neq \Delta P$ then go back to step 1.
6. return $\big((P_1, C_1), (P_2, C_2), (P_3, C_3), (P_4, C_4)\big)$

The predicted success probability for each iteration is $p_1^2 p_2^2$, hence the expected number of tries required $p_1^{-2} p_2^{-2}$, which is the example is $\left(\frac{3}{16}\right)^{-4} \approx 809$.
However the success probability may be amplified by other differentials that only differ in ΔO_2 or ΔI_3.
Given a set $\mathcal{R}_1$ of round-1,2 differentials with identical ΔP and a set $\mathcal{R}_2$ of round-3,4 differentials with identical ΔC, the total predicted success probability is

$$p_{success} = \left(\sum_{\Delta P, \Delta O_2 \in \mathcal{R}_1} \Pr[(\Delta P, \Delta O_2)]^2 \right) \left(\sum_{\Delta I_3, \Delta C \in \mathcal{R}_2} \Pr[(\Delta I_3, \Delta C)]^2 \right)$$

In theory also sets of characteristics over round 1 and round 2-4 increase the total success probability.
But note that a quartet may satisfy both 1-2/3-4-round split differentials and 1/2-4-round split differentials, so these events are not disjoint and therefore we may not simply add success probabilities of differential-pairs with different round splits.
The measured expected number of tries for the above $\Delta P, \Delta C$ is actually about 100, about 8 times less than expected..

Finding such a quartet against a uniformly selected random permutation instead of a block cipher has a success probability of 2^{-N}, where N is the state size in bits.
So when $p_1^{-2} p_2^{-2} \ll 2^N$, we can distinguish the cipher from a uniformly selected random permutation by trying to find such a quartet in a number of tries significantly smaller than 2^N.