7 Cryptanalysis

Cryptanalysis
Attacks such as exhaustive key-search do not exploit any properties of the encryption algorithm or implementation.
Structural attacks exploit structural weaknesses in the algorithm.
Two main types:

- Differential cryptanalysis
- Linear cryptanalysis

Side channel cryptanalysis exploits weaknesses in the implementation.

7.1 Structural Attacks

Structural Attacks
These attacks lead to important design criteria (go back to Shannon):

- Nonlinearity (in plaintext and in key input): 'confusion'
- Spreading of statistics over all bits: 'diffusion'

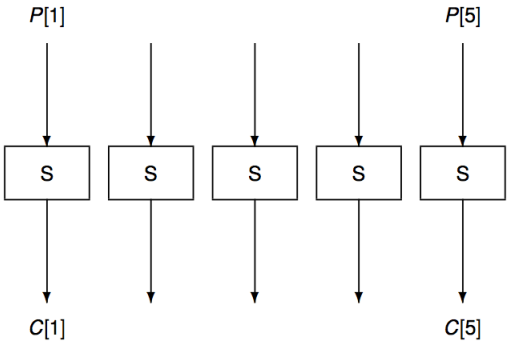
If ciphers are 'too linear', the following known plaintext attacks can succeed:

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

Also dangerous:

- $(P_1 + P_2) \to C_3 \approx (C_1 + C_2)$
- $(P_1 + P_2) \to (C_1 + C_2)$ with large probability.

Structural Attacks



If $C[1]$ is only function of $P[1]$, $K[1]$, then the cryptanalyst can solve separate 'sub-ciphers'.
If $C[1]$ depends only very weakly on $K[i]$, $P[j]$, then the cryptanalyst can solve separate problems.

Structural Attacks
Consider the following example S-Box:

	00	01	10	11
0	10	01	11	00
1	00	10	01	11

For input bits $x_0x_1x_2$, x_0 gives the row, and x_1x_2 gives the column.
For example, Sbox(010) = 11.
Consider the two inputs $X_1 = 110$ and $X_2 = 010$, and key $K = 011$.
$X_1 \oplus K = 101$, $X_2 \oplus K = 001$, so Sbox$(X_1 \oplus K) = 10$, Sbox$(X_2 \oplus K) = 01$.
If K is unknown, but the inputs and corresponding outputs to the S-Box are known then $X_1 \oplus K \in \{000, 101\}$ and $X_2 \oplus K \in \{001, 110\}$.
Since X_1 and X_2 are known, $K \in \{110, 011\} \cap \{011, 100\}$, so $K = 011$.

7.2 Differential Cryptanalysis

Differential Cryptanalysis
Differential cryptanalysis: propagation of differences:

$$\begin{array}{lcl} P_1 & \rightarrow & C_1 \\ P_1 + \delta & \rightarrow & C_1 + \varepsilon \end{array}$$

Look for special values of ε

- Probabilistic attack.
- First proposed by Murphy in 1990 for the cryptanalysis of FEAL-4.
- Further elaborated by Biham and Shamir in a series of papers.
- A chosen plaintext attack.
- Exploits the over-use of the XOR function in many contemporary block ciphers.

Differential Cryptanalysis
Differential cryptanalysis exploits the linearity of operations in block ciphers.
Some commonly used linear operations in block ciphers:

- $a = b \oplus c$
- a = the bits of b in (a known) permuted order

Linear relations can be found by solving a system of linear equations.
In a Feistel structure $R_i = L_{i-1} \oplus F(R_{i-1}, K_i)$ and $L_i = R_{i-1}$.
if F is linear (e.g. $F(R_{i-1}, K_i) = R_{i-1} \oplus K_i$), we can solve and find K_i.
F must therefore be non-linear: this is often achieved through the use of non-linear S-Boxes (e.g. DES).

Differential Cryptanalysis

1. Start with 2 known plaintexts P_0 and P_1 with known XOR difference (called the characteristic) $\Omega_P = P_0 \oplus P_1$, and trace through a probable pattern of differences after each round to obtain a difference for the ciphertext.
   - The assumption is that many pairs (P_0, P_1) with a given difference Ω_P yield the same output difference if the same subkey K_i is used.
   - That is $F(P_0, K_i) \oplus F(P_1, K_i)$ is a function of Ω_P with high probability.
2. Submit P_0 and P_1 for encryption to determine the actual differences under the unknown key.
3. If there is a match between the two values we suspect that all intermediate rounds are correct.
4. Can then work backwards through all the rounds to recover all the subkeys.

Differential Cryptanalysis
Consider the previous S-Box being used in a cipher in which the input to a round is XORed with the round key before being input into the S-Box.
For input X_1, the input to the S-Box is $X_1 \oplus K$ and for input X_2 the input to S-Box is $X_2 \oplus K$, where the key K is unknown.
The S-box input difference is $(X_1 \oplus K) \oplus (X_2 \oplus K) = X_1 \oplus X_2$.
So the input difference is independent of the key K.
This is the fundamental observation that enables differential cryptanalysis to work.

Differential Cryptanalysis
Given any S-Box, we can analyze it for useful input differences.
For each possible input value X, find all X_1, X_2 s.t. $X = X_1 \oplus X_2$.
Compute the corresponding output differences: $\text{Sbox}(X_1) \oplus \text{Sbox}(X_2)$.
By tabulating the results, we can find the most biased input values:

$X_1 \oplus X_2$	00	01	10	11
000	8	0	0	0
001	0	0	4	4
010	0	8	0	0
011	0	0	4	4
100	0	0	4	4
101	4	4	0	0
110	0	0	4	4
111	4	4	0	0

7.3 Linear Cryptanalysis

Linear Cryptanalysis
First described by Matsui.
Based on the idea of finding a linear correlation between the input and output.
A known plaintext attack.
Originally devised to attack DES: key can be found given 2^{47} known plaintexts.
Can also be used to attack FEAL-4.

Linear Cryptanalysis

1. For cipher with n-bit plaintext and ciphertext blocks and m-bit keys: let plaintext $P = p_1 \dots p_n$, ciphertext $C = c_1 \dots c_n$ and key $K = k_1 \dots k_m$.
2. Find fixed locations $1 \leq \alpha_1 < \alpha_2 < \dots \alpha_p \leq n$, $1 \leq \beta_1 < \beta_2 < \dots \beta_c \leq n$, $1 \leq \gamma_1 < \gamma_2 < \dots \gamma_k \leq m$ such that:

   $$(p_{\alpha_1} \oplus \dots \oplus p_{\alpha_p}) \oplus (c_{\beta_1} \oplus \dots \oplus c_{\beta_c}) = (k_{\gamma_1} \oplus \dots \oplus k_{\gamma_k})$$

   holds with probability $\neq \frac{1}{2}$ (the further from $\frac{1}{2}$ the more effective the equation).
3. Compute results for the LHS of the equation for a large number of plaintext/ciphertext pairs. If the result is 0 more than half the time, then guess RHS is 0 (similarly for 1). This gives us a linear equation on key bits. Then try to find more such equations.

Linear Cryptanalysis
For the previous S-Box, we denote the three input bits as $x_0x_1x_2$ and the two output bits as y_1y_2.
We tabulate the number of values for which each possible linear approximation holds:

	y_0	y_1	$y_0 \oplus y_1$
0	4	4	4
x_0	4	4	4
x_1	4	6	2
x_2	4	4	4
$x_0 \oplus x_1$	4	2	2
$x_0 \oplus x_2$	0	4	4
$x_1 \oplus x_2$	4	6	6
$x_0 \oplus x_1 \oplus x_2$	4	6	2

So $P(y_0 = x_0 \oplus x_2 \oplus 1) = 1$ and $P(y_0 \oplus y_1 = x_1 \oplus x_2) = 3/4$.

7.4 Tiny DES

Tiny DES
Tiny DES (or TDES) is a much simplified version of DES that employs:

- A 16-bit block size
- A 16-bit key size
- Four rounds
- Two S-Boxes, each mapping 6 bits to 4 bits
- A 12-bit subkey in each round

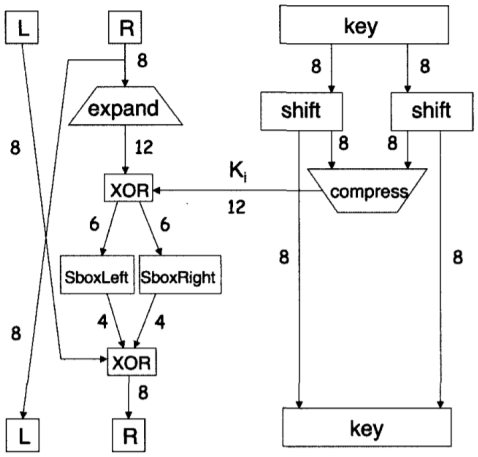
TDES has no P-box, initial or final permutation.

Tiny DES
TDES is a Feistel cipher, where for each round $i = 1, 2, 3, 4$:

$$\begin{aligned} L_i &= R_{i-1} \\ R_i &= L_{i-1} \oplus F(R_{i-1}, K_i) \end{aligned}$$

where the plaintext is (L_0, R_0) and the ciphertext is (L_4, R_4).

Tiny DES
A single round of TDES looks as follows:



Tiny DES
TDES has two S-Boxes denoted by SboxLeft(X) and SboxRight(X).
Both S-Boxes map 6 bits to 4 bits, as in standard DES.
We define the function:

$$F(R, K) = \text{Sboxes}(\text{expand}(R) \oplus K)$$

where:

$$\text{Sboxes}(x_0x_1x_2\ldots x_{11}) = (\text{SboxLeft}(x_0x_1\ldots x_5), \text{SboxRight}(x_6x_7\ldots x_{11}))$$

The expansion permutation is given by:

$$\text{expand}(R) = \text{expand}(r_0r_1r_2r_3r_4r_5r_6r_7) = (r_4r_7r_2r_1r_5r_7r_0r_2r_6r_5r_0r_3)$$

Tiny DES

SboxLeft(X) and SboxRight(X) in hex notation are:

	$x_1x_2x_3x_4$															
x_0x_5	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	6	9	A	3	4	D	7	8	E	1	2	B	5	C	F	0
1	9	E	B	A	4	5	0	7	8	6	3	2	C	D	1	F
2	8	1	C	2	D	3	E	F	0	9	5	A	4	B	6	7
3	9	0	2	5	A	D	6	E	1	8	B	C	3	4	7	F

	$x_1x_2x_3x_4$															
x_0x_5	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	C	5	0	A	E	7	2	8	D	4	3	9	6	F	1	B
1	1	C	9	6	3	E	B	2	F	8	4	5	D	A	0	7
2	F	A	E	6	D	8	2	4	1	7	9	0	3	5	B	C
3	0	A	3	C	8	2	1	E	9	7	F	6	B	5	D	4

Tiny DES

As with DES, each row in a TDES S-Box is a permutation of the hexadecimal digits $\{0,1,2,\ldots,E,F\}$.

The 16-bit key is denoted

$$K = k_0k_1k_2k_3k_4k_5k_6k_7k_8k_9k_{10}k_{11}k_{12}k_{13}k_{14}k_{15}$$

The subkey is generated as follows. Let:

$$\begin{aligned} LK_0 &= k_0k_1\ldots k_7 \\ RK_0 &= k_8k_9\ldots k_{15} \end{aligned}$$

Then for each round $i = 1,2,3,4$:

$$\begin{aligned} LK_i &= \text{rotate } LK_{i-1} \text{ left by 2} \\ RK_i &= \text{rotate } RK_{i-1} \text{ left by 1} \end{aligned}$$

Tiny DES

Subkey K_i is obtained by selecting bits 0, 2, 3, 4, 5, 7, 9, 10, 11, 13, 14, and 15 of (LK_i, RK_i).

The subkeys K_i can be given explicitly as follows:

$$\begin{aligned} K_1 &= k_2k_4k_5k_6k_7k_1k_{10}k_{11}k_{12}k_{14}k_{15}k_8 \\ K_2 &= k_4k_6k_7k_0k_1k_3k_{11}k_{12}k_{13}k_{15}k_8k_9 \\ K_3 &= k_6k_0k_1k_2k_3k_5k_{12}k_{13}k_{14}k_8k_9k_{10} \\ K_4 &= k_0k_2k_3k_4k_5k_7k_{13}k_{14}k_{15}k_9k_{10}k_{11} \end{aligned}$$

Geoff Hamilton

7.5 Differential Cryptanalysis of Tiny DES

Differential Cryptanalysis of Tiny DES

The differential analysis of Tiny DES focuses on the right S-Box.

We tabulate $\text{SboxRight}(X_1) \oplus \text{SboxRight}(X_2)$ for all pairs X_1 and X_2 with $X_1 \oplus X_2 = 001000$.

Some useful properties:

$X_1 \oplus X_2 = 001000$ implies $\text{SboxRight}(X_1) \oplus \text{SboxRight}(X_2) = 0010$ with probability 3/4.

$X_1 \oplus X_2 = 000000$ implies $\text{SboxRight}(X_1) \oplus \text{SboxRight}(X_2) = 0000$

Differential Cryptanalysis of Tiny DES

Choose plaintext $P = (L,R)$ and $P' = (L',R')$ such that:

$$P \oplus P' = 0000\ 0000\ 0000\ 0010 = 0x0002$$

$F(R,K) \oplus F(R',K) = \text{Sboxes}(\text{expand}(R) \oplus K) \oplus \text{Sboxes}(\text{expand}(R') \oplus K)$

From the definition of expand:

$$\text{expand}(0000\ 0010) = 000000\ 001000$$

Since expand is linear, if $X_1 \oplus X_2 = 0000\ 0010$ then:

$$\begin{aligned} \text{expand}(X_1) \oplus \text{expand}(X_2) &= \text{expand}(X_1 \oplus X_2) \\ &= \text{expand}(0000\ 0010) \\ &= 000000\ 001000 \end{aligned}$$

Differential Cryptanalysis of Tiny DES

For the chosen plaintext, $R \oplus R' = 0000\ 0010$

$$\begin{aligned} F(R,K) \oplus F(R',K) &= \text{Sboxes}(\text{expand}(R) \oplus K) \oplus \text{Sboxes}(\text{expand}(R') \oplus K) \\ &= (\text{SboxLeft}(A \oplus K), \text{SboxRight}(B \oplus K)) \\ &\quad \oplus (\text{SboxLeft}(A' \oplus K), \text{SboxRight}(B' \oplus K)) \\ &= (\text{SboxLeft}(A \oplus K) \oplus \text{SboxLeft}(A' \oplus K)), \\ &\quad (\text{SboxRight}(B \oplus K) \oplus \text{SboxRight}(B' \oplus K)) \end{aligned}$$

Where $A \oplus A' = 000000$ and $B \oplus B' = 001000$.

So $F(R,K) \oplus F(R',K) = 0000\ 0010$ with probability 3/4.

Differential Cryptanalysis of Tiny DES

$(L_0,R_0) = P$	$(L'_0,R'_0) = P'$	$P \oplus P' = 0x0002$	Probability
$L_1 = R_0$ $R_1 = L_0 \oplus F(R_0,K_1)$	$L'_1 = R'_0$ $R'_1 = L'_0 \oplus F(R'_0,K_1)$	$(L_1,R_1) \oplus (L'_1,R'_1) = 0x0202$	3/4
$L_2 = R_1$ $R_2 = L_1 \oplus F(R_1,K_2)$	$L'_2 = R'_1$ $R'_2 = L'_1 \oplus F(R'_1,K_2)$	$(L_2,R_2) \oplus (L'_2,R'_2) = 0x0200$	$(3/4)^2$
$L_3 = R_2$ $R_3 = L_2 \oplus F(R_2,K_3)$	$L'_3 = R'_2$ $R'_3 = L'_2 \oplus F(R'_2,K_3)$	$(L_3,R_3) \oplus (L'_3,R'_3) = 0x0002$	$(3/4)^2$
$L_4 = R_3$ $R_4 = L_3 \oplus F(R_3,K_4)$	$L'_4 = R'_3$ $R'_4 = L'_3 \oplus F(R'_3,K_4)$	$(L_4,R_4) \oplus (L'_4,R'_4) = 0x0202$	$(3/4)^3$

Differential Cryptanalysis of Tiny DES

$P \oplus P' = 0000\ 0000\ 0000\ 0010$

So:

$$\begin{aligned} R_0 \oplus R'_0 &= 0000\ 0010 \\ L_0 \oplus L'_0 &= 0000\ 0000 \end{aligned}$$

Therefore:

$$\begin{aligned} R_1 \oplus R'_1 &= (L_0 \oplus F(R_0, K_1)) \oplus (L'_0 \oplus F(R'_0, K_1)) \\ &= (L_0 \oplus L'_0) \oplus (F(R_0, K_1) \oplus F(R'_0, K_1)) \\ &= 0000\ 0000 \oplus 0000\ 0010 \text{ with probability } 3/4 \\ &= 0000\ 0010 \text{ with probability } 3/4 \end{aligned}$$

Differential Cryptanalysis of Tiny DES

From this it follows that:

$$\begin{aligned} R_2 \oplus R'_2 &= (L_1 \oplus F(R_1, K_2)) \oplus (L'_1 \oplus F(R'_1, K_2)) \\ &= (L_1 \oplus L'_1) \oplus (F(R_1, K_2) \oplus F(R'_1, K_2)) \\ &= (R_0 \oplus R'_0) \oplus (F(R_1, K_2) \oplus F(R'_1, K_2)) \\ &= 0000\ 0010 \oplus 0000\ 0010 \text{ with probability } (3/4)^2 \\ &= 0000\ 0000 \text{ with probability } (3/4)^2 \end{aligned}$$

$R_3 \oplus R'_3$ and $R_4 \oplus R'_4$ are obtained in a similar manner.

Differential Cryptanalysis of Tiny DES

We now derive an algorithm to recover some of the unknown key bits.

Since Tiny DES is a Feistel cipher:

$$\begin{aligned} R_4 &= L_3 \oplus F(R_3, K_4) \\ R'_4 &= L'_3 \oplus F(R'_3, K_4) \end{aligned}$$

$L_4 = R_3$ and $L'_4 = R'_3$, so:

$$\begin{aligned} R_4 &= L_3 \oplus F(L_4, K_4) \\ R'_4 &= L'_3 \oplus F(L'_4, K_4) \end{aligned}$$

Or equivalently:

$$\begin{aligned} L_3 &= R_4 \oplus F(L_4, K_4) \\ L'_3 &= R'_4 \oplus F(L'_4, K_4) \end{aligned}$$

Differential Cryptanalysis of Tiny DES

If $C \oplus C' = 0x0202$, it is likely that $L_3 \oplus L'_3 = 0000\ 0000$ (i.e. $L_3 = L'_3$).

So:

$$R_4 \oplus F(L_4, K_4) = R'_4 \oplus F(L'_4, K_4)$$

Or equivalently:

$$R_4 \oplus R'_4 = F(L_4, K_4) \oplus F(L'_4, K_4)$$

Also, since $C \oplus C' = (L_4, R_4) \oplus (L'_4, R'_4) = 0x0202$:

$$\begin{aligned} R_4 \oplus R'_4 &= 0000\ 0010 \\ L_4 \oplus L'_4 &= 0000\ 0010 \end{aligned}$$

Geoff Hamilton

Differential Cryptanalysis of Tiny DES

Let $L_4 = l_0 l_1 l_2 l_3 l_4 l_5 l_6 l_7$ and $L'_4 = l'_0 l'_1 l'_2 l'_3 l'_4 l'_5 l'_6 l'_7$.
$l_i = l'_i$ for i = 0,1,2,3,4,5,7 and $l_6 \neq l'_6$.
So:

$$\begin{aligned} 0000\ 0010 \quad = \quad & (\text{SboxLeft}(l_4 l_7 l_2 l_1 l_5 l_7 \oplus k_0 k_2 k_3 k_4 k_5 k_7), \\ & \text{SboxRight}(l_0 l_2 l_6 l_5 l_0 l_3 \oplus k_{13} k_{14} k_{15} k_9 k_{10} k_{11})) \\ & \oplus \\ & (\text{SboxLeft}(l'_4 l'_7 l'_2 l'_1 l'_5 l'_7 \oplus k_0 k_2 k_3 k_4 k_5 k_7), \\ & \text{SboxRight}(l'_0 l'_2 l'_6 l'_5 l'_0 l'_3 \oplus k_{13} k_{14} k_{15} k_9 k_{10} k_{11})) \end{aligned}$$

We gain no information about the subkey K_4 from the left S-Box since $l_i = l'_i$ for all $i \neq 6$.

Differential Cryptanalysis of Tiny DES

We use the following algorithm for recovering the bits of subkey K_4 from the right S-Box:

```
for i = 0 to 63
   count[i] = 0
next i
for i = 1 to iterations
   Choose P and P' with P ⊕ P' = 0x0002
   Obtain corresponding C = c0c1...c15 and C' = c'0c'1...c'15
   if C ⊕ C' == 0x0202 then
      for K = 0 to 63
         if (SboxRight(c0c2c6c5c0c3 ⊕ K)
           ⊕ SboxRight(c'0c'2c'6c'5c'0c'3 ⊕ K)) == 0010 then
            increment count[K]
         end if
      next K
   end if
next i
```

Differential Cryptanalysis of Tiny DES

If, for example, 100 pairs of plaintexts P and P' that satisfy $P \oplus P' = 0x0002$ were generated and 47 of the resulting ciphertext pairs satisfied $C \oplus C' = 0x0202$.
The algorithm can then be run for each of these 47 ciphertext pairs.
Say we found that each of the four subkeys 000001, 001001, 110000, and 000111 had the maximum count of 47, while no other had a count greater than 39.
We can conclude that subkey K_4 must be one of these four values:

$$k_{13} k_{14} k_{15} k_9 k_{10} k_{11} \in \{000001, 001001, 110000, 111000\}$$

Or equivalently:

$$k_{13} k_{14} k_9 k_{10} k_{11} \in \{00001, 11000\}$$

Differential Cryptanalysis of Tiny DES
We can then exhaustively search over the remaining 2^{11} unknown key bits, and for each of these try both of these possibilities, giving a total of 2^{12} possible keys.
We expect to try about half of the possibilities (about 2^{11} keys) before finding the correct key.
The total expected work to recover the entire key by this method is about 2^{11} encryptions, plus the work required for the differential attack, which is insignificant in comparison.
We can recover the entire 16-bit key with a work factor of about 2^{11} encryptions, which is much better than an exhaustive key search, which has an expected work of 2^{15} encryptions.
This shows that a shortcut attack exists, and as a result Tiny DES is insecure.

7.6 Linear Cryptanalysis of Tiny DES

Linear Cryptanalysis of Tiny DES
The linear cryptanalysis of Tiny DES focuses on the left S-Box.
If $y_0y_1y_2y_3 = \text{SboxLeft}(x_0x_1x_2x_3x_4x_5)$ then $y_1 = x_2$ and $y_2 = x_3$ with probability 3/4.
If the plaintext $P = (L_0, R_0)$ and $R_0 = r_0r_1r_2r_3r_4r_5r_6r_7$ then the expansion permutation is given by:

$$\text{expand}(R_0) = \text{expand}(r_0r_1r_2r_3r_4r_5r_6r_7) = (r_4r_7r_2r_1r_5r_7r_0r_2r_6r_5r_0r_3)$$

The input to the left S-Box in the first round is given by:

$$\text{expand}(R_0) \oplus K_1 = r_4r_7r_2r_1r_5r_7 \oplus k_2k_4k_5k_6k_7k_1$$

For the left S-Box, $y_1 = r_2 \oplus k_5$ and $y_2 = r_1 \oplus k_6$ with probability 3/4.

Linear Cryptanalysis of Tiny DES
If $L_0 = l_0l_1l_2l_3l_4l_5l_6l_7$ and $R_1 = r'_0r'_1r'_2r'_3r'_4r'_5r'_6r'_7$
The output of the left S-Box from the first round is XORed with $l_0l_1l_2l_3$ to give $r'_0r'_1r'_2r'_3$.
So $r'_1 = r_2 \oplus k_5 \oplus l_1$ and $r'_2 = r_1 \oplus k_6 \oplus l_2$ with probability 3/4.
Similar results hold for subsequent rounds, where the specific key bits depend on the subkey K_i, so we can chain the linear approximation through multiple rounds.
Since $L4 = c_0c_1c_2c_3c_4c_5c_6c_7$, we can determine that $k_0 \oplus k_1 = c_1 \oplus p_{10}$ and $k_7 \oplus k_2 = c_2 \oplus p_9$ with probability $(3/4)^3$.
Since c_1, c_2, p_9, and p_{10} are all known, we have obtained some information about the key bits k_0, k_1, k_2, and k_7.

Linear Cryptanalysis of Tiny DES

$(L_0, R_0) = (p_0 \dots p_7, p_8 \dots p_{15})$	Bits 1 and 2	Probability
$L_1 = R_0$	p_9, p_{10}	1
$R_1 = L_0 \oplus F(R_0, K_1)$	$p_1 \oplus p_{10} \oplus k_5$, $p_2 \oplus p_9 \oplus k_6$	3/4
$L_2 = R_1$	$p_1 \oplus p_{10} \oplus k_5$, $p_2 \oplus p_9 \oplus k_6$	3/4
$R_2 = L_1 \oplus F(R_1, K_2)$	$p_2 \oplus k_6 \oplus k_7$, $p_1 \oplus k_5 \oplus k_0$	$(3/4)^2$
$L_3 = R_2$	$p_2 \oplus k_6 \oplus k_7$, $p_1 \oplus k_5 \oplus k_0$	$(3/4)^2$
$R_3 = L_2 \oplus F(R_3, K_4)$	$p_{10} \oplus k_0 \oplus k_1$, $p_9 \oplus k_7 \oplus k_2$	$(3/4)^3$
$L_4 = R_3$	$p_{10} \oplus k_0 \oplus k_1$, $p_9 \oplus k_7 \oplus k_2$	$(3/4)^3$
$R_4 = L_3 \oplus F(R_3, K_4)$		

Linear Cryptanalysis of Tiny DES
Given known plaintexts $P = p_0 p_1 p_2 \dots p_{15}$ along with corresponding ciphertexts $C = c_0 c_1 c_2 \dots c_{15}$, we increment counters depending on whether:

1. $c_1 \oplus p_{10} = 0$ or $c_1 \oplus p_{10} = 1$
2. $c_2 \oplus p_9 = 0$ or $c_2 \oplus p_9 = 1$

If, for example, with 100 known plaintexts the following results were obtained:

- $c_1 \oplus p_{10} = 0$ occurred 38 times
- $c_1 \oplus p_{10} = 1$ occurred 62 times
- $c_2 \oplus p_9 = 0$ occurred 62 times
- $c_2 \oplus p_9 = 1$ occurred 38 times

We conclude that it is likely that $k_0 \oplus k_1 = 1$ and $k_7 \oplus k_2 = 0$.

Linear Cryptanalysis of Tiny DES
We have only recovered the equivalent of two bits of information.
We can then exhaustively search over the remaining 2^{14} unknown key bits.
We expect to try about half of the possibilities (about 2^{13} keys) before finding the correct key.
The total expected work to recover the entire key by this method is about 2^{13} encryptions, plus the work required for the linear attack, which is insignificant in comparison.
We can recover the entire 16-bit key with a work factor of about 2^{13} encryptions, which is better than an exhaustive key search, which has an expected work of 2^{15} encryptions.
This shows that another shortcut attack exists, and as a result Tiny DES is insecure.

7.7 FEAL-4

FEAL
Fast data Encryption ALgorithm.
Designed as replacement for DES.
Designed to be fast and efficient with modest security.
Original version (FEAL-4) found to be weak.

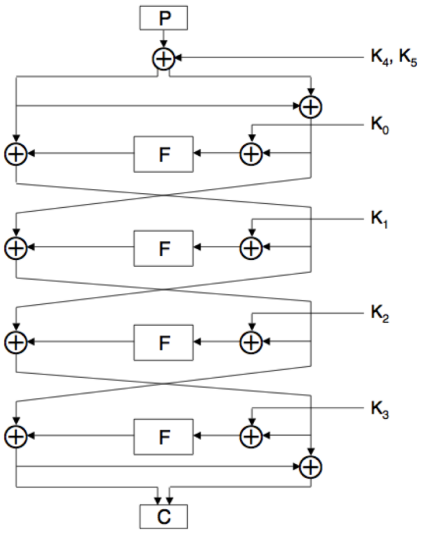
- Many "improved" versions followed
- All are flawed to some degree

Important in history of cryptanalysis.
Differential cryptanalysis developed for FEAL.
Good example to illustrate both linear and differential attacks.

FEAL-4
We will look more closely at FEAL-4 to analyse its weaknesses.

- 4-round Feistel cipher with a 64-bit block and 64-bit key.
- A key scheduler expands a 64-bit key into 12 16-bit subkeys or round keys.
- Some of these subkeys are used for key whitening, in which they are combined with portions of the data using XOR before the first and after the last round.
- The cipher can be simplified to an equivalent cipher in which some of the whitening has been removed and 6 32-bit subkeys $K_0, \ldots, K_5$ are used.
- Round function F maps 32 bits to 32 bits.

FEAL-4



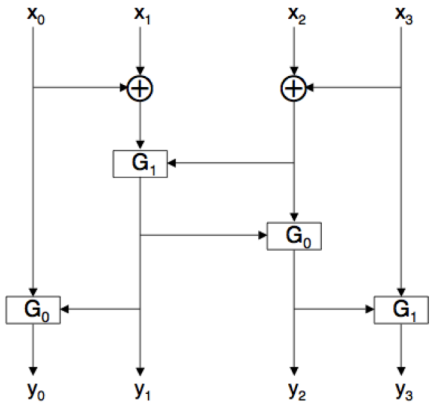
FEAL-4
To define the round function F, we first define:

- $G_0(a,b) = (a+b \pmod{256}) <<< 2$
- $G_1(a,b) = (a+b+1 \pmod{256}) <<< 2$

Where $<<<$ is left cyclic shift (rotation)
Then $F(x_0,x_1,x_2,x_3) = (y_0,y_1,y_2,y_3)$ where:

- $y_0 = G_0(x_0, y_1)$
- $y_1 = G_1(x_0 \oplus x_1, x_2 \oplus x_3)$
- $y_2 = G_0(y_1, x_2 \oplus x_3)$
- $y_3 = G_1(y_2, x_3)$

FEAL-4
The round function F can be viewed schematically as follows:



7.8 Differential Cryptanalysis of FEAL-4

Differential Cryptanalysis of FEAL-4
Some useful properties:

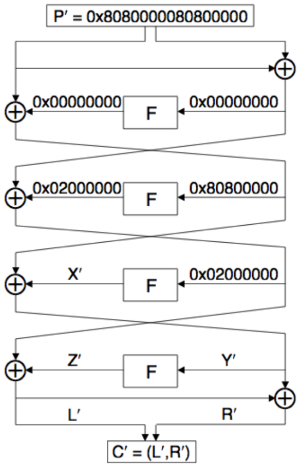
- $A_0 \oplus A_1 = 0$ implies $F(A_0) = F(A_1)$
- $A_0 \oplus A_1 = 0x80800000$ implies $F(A_0) \oplus F(A_1) = 0x02000000$

Differential attack is based on this:

- Choose plaintext P_0 and P_1 such that $P_0 \oplus P_1 = 0x8080000080800000$

- Determine ciphertexts C_0 and C_1 corresponding to P_0 and P_1 respectively.
- Let $P' = P_0 \oplus P_1$ and $C' = C_0 \oplus C_1$
- Consider what happens to P' as it passes through the cipher.

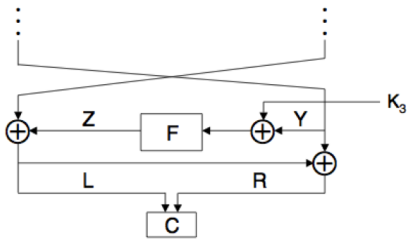
Differential Cryptanalysis of FEAL-4



Differential Cryptanalysis of FEAL-4
Characteristic for P' gets us half way through.
We can then work backwards from C' and try to meet in the middle.
$Y' = L' \oplus R'$, which gives us Y' (L' and R' are known).
$X' = 0x80800000 \oplus Y'$, which gives us X'
$Z' = 0x02000000 \oplus L'$, which gives us Z' (L' is known).
Now that we have calculated the differentials, we attack the final subkey K_3.

Differential Cryptanalysis of FEAL-4
To determine K_3, we need to know the actual text inputs into the last round function (not the differentials).
For ciphertext $C_0 = (L_0, R_0)$ we compute $Y_0 = L_0 \oplus R_0$
and for ciphertext $C_1 = (L_1, R_1)$ we compute $Y_1 = L_1 \oplus R_1$
We guess K_3 and compute $Z_0 = F(Y_0 \oplus K_3)$ and $Z_1 = F(Y_1 \oplus K_3)$
We then compare the true value of Z' to the guessed value $Z_0 \oplus Z_1$ and the key value

with the most matches survives.



Differential Cryptanalysis of FEAL-4
Using 4 chosen plaintext pairs:

- Work is of order 2^{32}.
- Expect one K_3 to survive.

Good divide and conquer strategy

- Divides large search space for overall key into smaller search spaces for subkeys.

It is possible to do better.

- Can reduce work to about 2^{17}

Differential Cryptanalysis of FEAL-4
For 32-bit word $W = (b_0, b_1, b_2, b_3)$, define $M(W) = (0, b_0 \oplus b_1, b_2 \oplus b_3, 0)$
For all possible $A = (0, a_0, a_1, 0)$, compute $Q_0 = F(M(Y_0) \oplus A)$ and $Q_1 = F(M(Y_1) \oplus A)$
Can be used to find 16 bits of K_3
When $A = M(K_3)$ by definition of F, we have $\langle Q_0 \oplus Q_1 \rangle_{8\ldots23} = \langle Z' \rangle_{8\ldots23}$ where $\langle X \rangle_{i\ldots j}$ is bits i to j of X
Can recover K_3 with about 2^{17} work.
Once K_3 is known, can successively recover K_2, K_1, K_0 and finally K_4, K_5.

- Attack is similar in each case
- Some require different characteristics
- There are a few subtle points

Differential Cryptanalysis of FEAL-4
Primary algorithm for recovering K_3:

```
// Characteristic is 0x8080000080800000
P_0 = random 64-bit value
P_1 = P_0 ⊕ 0x8080000080800000
// Given corresponding ciphertexts
// C_0 = (L_0, R_0) and C_1 = (L_1, R_1)
Y_0 = L_0 ⊕ R_0
Y_1 = L_1 ⊕ R_1
L' = L_0 ⊕ L_1
Z' = L' ⊕ 0x02000000
for (a_0, a_1) = (0x00, 0x00) to (0xff, 0xff)
    Q_0 = F(M(Y_0) ⊕ (0x00, a_0, a_1, 0x00))
    Q_1 = F(M(Y_1) ⊕ (0x00, a_0, a_1, 0x00))
    if ⟨Q_0 ⊕ Q_1⟩_8...23 == ⟨Z'⟩_8...23 then
        Save (a_0, a_1)
    end if
next (a_0, a_1)
```

Differential Cryptanalysis of FEAL-4
Secondary algorithm for recovering K_3:

```
// P_0, P_1, C_0, C_1, Y_0, Y_1, Z' as in primary
// Given list of saved (a_0, a_1) from primary
for each primary survivor (a_0, a_1)
    for (c_0, c_1) = (0x00, 0x00) to (0xff, 0xff)
        D = (c_0, a_0 ⊕ c_0, a_1 ⊕ c_1, c_1)
        Z̃_0 = F(Y_0 ⊕ D)
        Z̃_1 = F(Y_1 ⊕ D)
        if Z̃_0 ⊕ Z̃_1 == Z' then
            Save D // candidate subkey K_3
        end if
    next (c_0, c_1)
next (a_0, a_1)
```

Differential Cryptanalysis of FEAL-4
In FEAL-4, the differential for K_3 holds with probability 1
In most differential attacks, probability is small:

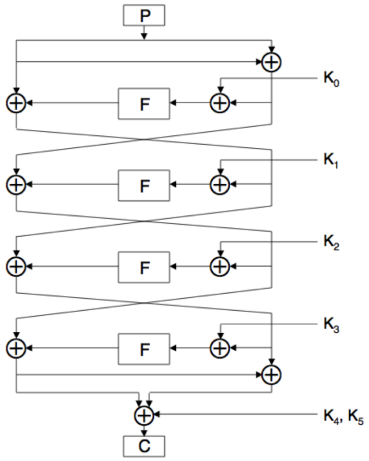
- Increases chosen plaintext requirement
- Increases work factor

Geoff Hamilton

Differential cryptanalysis seldom practical.
Usually only a theoretical tool.

7.9 Linear Cryptanalysis of FEAL-4

Linear Cryptanalysis of FEAL-4
Equivalent form for FEAL-4:

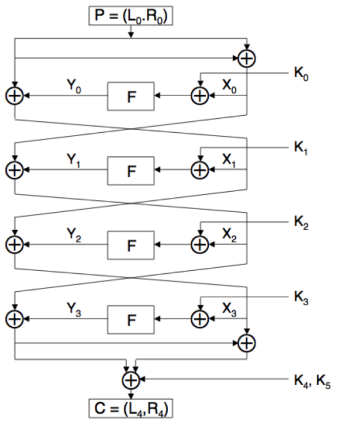


Linear Cryptanalysis of FEAL-4
Let X be 32-bit word, $X = x_0 \dots x_{31}$
Define $S_{i,j}(X) = x_i \oplus x_j$ and $S_i(X) = x_i$
Attack uses fact that for bytes a and b: $S_7(a \oplus b) = S_7(a + b \pmod{256})$
Recall $G_0(a,b) = (a + b \pmod{256}) <<< 2$, so $S_5(G_0(a,b)) = S_7(a \oplus b)$
Also, $S_5(G_1(a,b)) = S_7(a \oplus b) \oplus 1$
Let $Y = F(X)$, where X, Y are 32-bit words
Then it can be shown that:

- $S_{13}(Y) = S_{7,15}(X) \oplus S_{23,31}(X) \oplus 1$
- $S_5(Y) = S_{15}(Y) \oplus S_7(X)$
- $S_{15}(Y) = S_{21}(Y) \oplus S_{23,31}(X)$
- $S_{23}(Y) = S_{29}(Y) \oplus S_{31}(X) \oplus 1$

Linear Cryptanalysis of FEAL-4

We label the FEAL-4 intermediate steps:



Linear Cryptanalysis of FEAL-4

$$\begin{aligned} S_{23,29}(Y_3) &= S_{31}(X_3 \oplus K_3) \oplus 1 \\ &= S_{31}(L_4 \oplus K_4 \oplus R_4 \oplus K_5 \oplus K_3) \oplus 1 \\ &= S_{31}(L_4 \oplus R_4) \oplus S_{31}(K_4 \oplus K_5 \oplus K_3) \oplus 1 \\ S_{23,29}(Y_1) &= S_{23,29}F(X_1 \oplus K_1) \\ &= S_{23,29}F(K_1 \oplus Y_0 \oplus L_0) \\ &= S_{31}(K_1) \oplus S_{31}(Y_0) \oplus S_{31}(L_0) \oplus 1 \\ S_{31}(Y_0) &= S_{31}F(X_0 \oplus K_0) \\ &= S_{31}F(L_0 \oplus R_0 \oplus K_0) \end{aligned}$$

Combining all of these results and rearranging terms, we obtain:

$$a = S_{23,29}(L_0 \oplus R_0 \oplus L_4) \oplus S_{31}(L_0 \oplus L_4 \oplus R_4) \oplus S_{31}(F(L_0 \oplus R_0 \oplus K_0)) \quad (4.37)$$

Where $a = S_{31}(K_1 \oplus K_3 \oplus K_4 \oplus K_5) \oplus S_{23,29}(K_4)$
Treat a as unknown, but constant.

Linear Cryptanalysis of FEAL-4

Linear attack to find K_0:

```
// Given (plaintext,ciphertext) pairs (P_i, C_i), i = 0, 1, 2, ..., n − 1
for K = 0 to 2^32 − 1 // putative K_0
    count[0] = count[1] = 0
    for i = 0 to n − 1
        j = bit computed in right-hand-side of (4.37)
        count[j] = count[j] + 1
    next i
    if count[0] == n or count[1] == n then
        Save K // candidate for K_0
    end if
next K
```

Linear Cryptanalysis of FEAL-4
Perform exhaustive search over all choices for K_0
Test all known plaintext/ciphertext pairs.
If a is not constant, then guess for K_0 is incorrect
Possible to improve on linear attack:

- Exhaust for 12 bits of K_0 first
- Work is much less than 2^{32}

Can extend this attack to recover other subkeys.

Linear Cryptanalysis of FEAL-4
We define: $\tilde{K_0} = M(K_0) = (0, \langle K_0 \rangle_{0...7} \oplus \langle K_0 \rangle_{8...15}, \langle K_0 \rangle_{16...23} \oplus \langle K_0 \rangle_{24...31}, 0)$
We then calculate the following fixed, but unknown, constant:

$$S_{5,13}(L_0 \oplus R_0 \oplus L_4) \oplus S_{21}(L_0 \oplus R_0 \oplus L_4) \oplus S_{15}(L_0 \oplus L_4 \oplus R_4) \oplus S_{15}(F(L_0 \oplus R_0 \oplus \tilde{K_0}))(*)$$

$S_{15}(F(L_0 \oplus R_0 \oplus \tilde{K_0}))$ depends only on the bits of $\langle \tilde{K_0} \rangle_{9...15,17...23}$
Also, bits 9 and 17 of $\tilde{K_0}$ are XORed in $(*)$, so these bits can also be treated as constant (but unknown) values.
We are left with an expression that depends only on the 12 unknown key bits $\langle \tilde{K_0} \rangle_{10...15,18...23}$
This allows for an exhaustive search for 12 bits of $\tilde{K_0}$.
The remaining bits can also be found by deriving similar expressions and performing an exhaustive search.

7.10 Side Channel Cryptanalysis

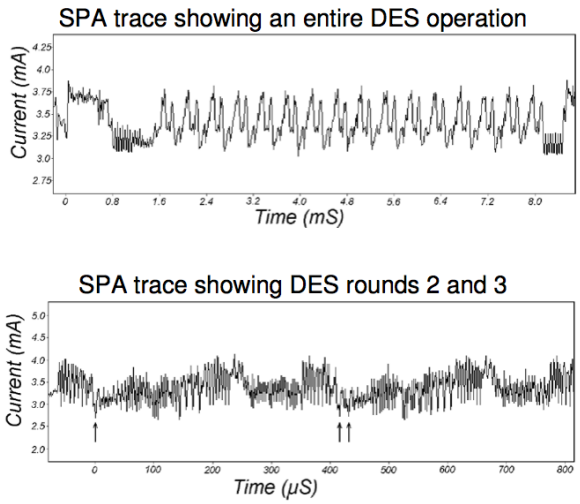
Side Channel Cryptanalysis
A correct implementation of a strong protocol is not necessarily secure.
Failures can be caused by

- Defective computation
- Information leaked during secret key operations
- Timing information
- Invasive measuring techniques
- Electromagnetic emanations

Simple Power Analysis (SPA)
Focus on the use of visual inspection techniques to identify relevant power fluctuations during cryptographic operations.
Interpretation of power traces:

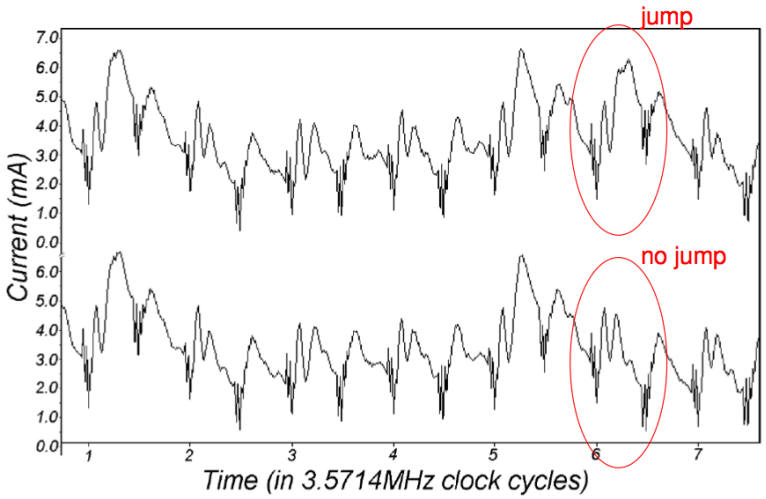
- Power consumption measurements taken across a cryptographic operation
- Typically current used by a device over time

Simple Power Analysis of DES



Simple Power Analysis of DES

SPA DES trace showing differences in power consumption of different microprocessor instructions:



Simple Power Analysis of DES
Simple Power analysis can reveal sequence of instructions executed.
It can be use to break cryptographic implementations in which the execution path depends on the data being processed:

- DES key schedule
- DES permutations
- Comparisons
- Multipliers
- Exponentiators

Simple Power Analysis
In general, techniques to prevent Simple Power Analysis are fairly simple:

- Avoid procedures that use secret intermediates or keys for conditional branching operations
- Hard-wired implementations of symmetric cryptography algorithms

Differential Power Analysis
Use of statistical analysis and error correction techniques to extract information correlated to secret keys.
Based on the effects correlated to data values being manipulated.
More powerful than Simple Power Analysis and is much more difficult to prevent.

Differential Power Analysis
Data collection:

- Capture power traces $T_1 \dots T_m[1 \dots k]$ containing k samples each
- Record the ciphertexts $C_1 \dots C_m$
- Knowledge of plaintext is not required

Data analysis:

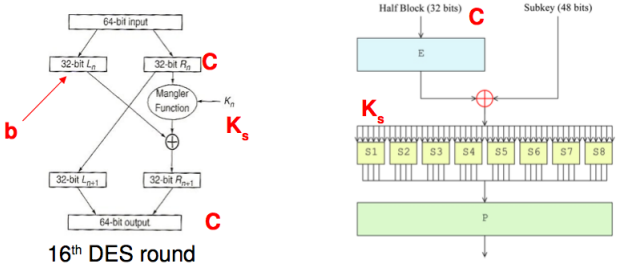
- Differential Power Analysis selection function $D(C,b,K_s) \rightarrow \{0,1\}$
- Compute k-sample differential trace $\Delta D[1 \dots k]$, where:

$$\begin{aligned} \Delta_D[j] &= \frac{\sum_{i=1}^{m} D(C_i,b,K_s)\mathbf{T}_i[j]}{\sum_{i=1}^{m} D(C_i,b,K_s)} - \frac{\sum_{i=1}^{m} (1-D(C_i,b,K_s))\,\mathbf{T}_i[j]}{\sum_{i=1}^{m} (1-D(C_i,b,K_s))} \\ &\approx 2\left(\frac{\sum_{i=1}^{m} D(C_i,b,K_s)\mathbf{T}_i[j]}{\sum_{i=1}^{m} D(C_i,b,K_s)} - \frac{\sum_{i=1}^{m} \mathbf{T}_i[j]}{m}\right). \end{aligned}$$

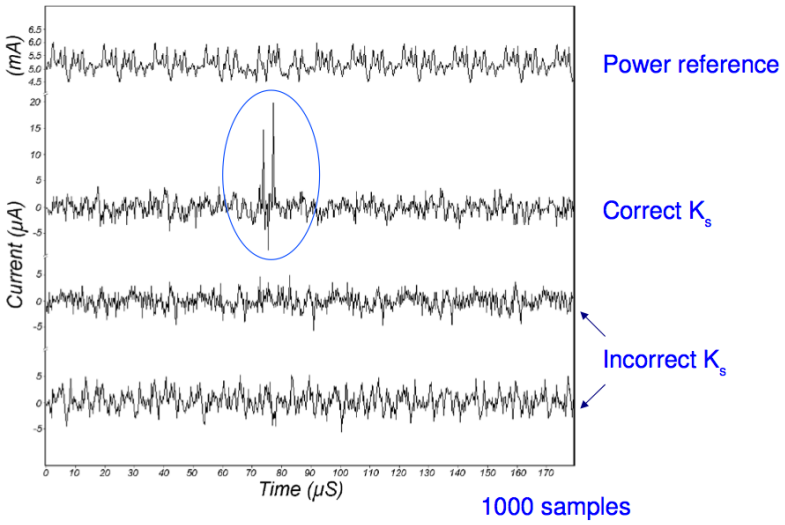
Differential Power Analysis
Differential Power Analysis selection function $D(C,b,K_s)$ is defined as:

- Returning the value of bit b of the DES intermediate L at the beginning of the 16^{th} round $(0 \leq b < 32)$
- C is the corresponding ciphertext
- K_s is the 6 key bits entering the S-Box corresponding to bit b $(0 \leq K_s < 2^6)$

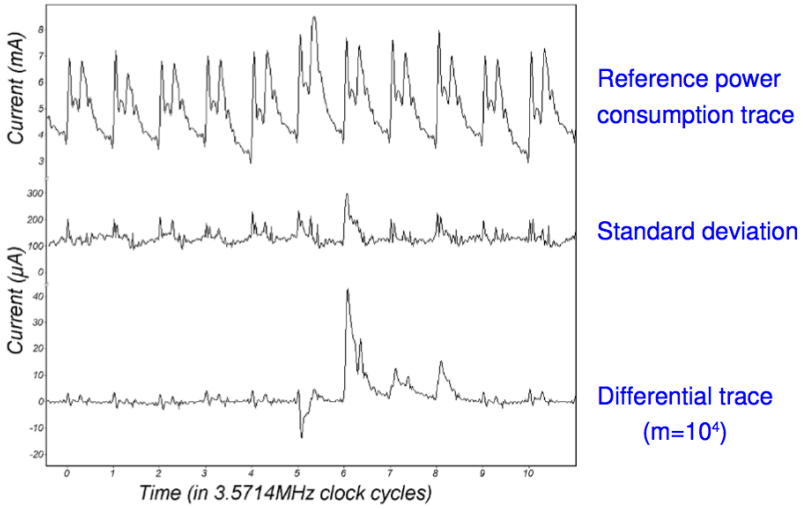
Repeat procedure to find all K_s values (8) to get the entire subkey



Differential Power Analysis
Differential Power Analysis traces for DES:



Differential Power Analysis
Quantitative Differential Power Analysis measurements:



Differential Power Analysis
Noise can be a problem:

- Electronic radiation and thermal noise
- Quantization errors
- Uncorrected temporal misalignment

Differential Power Analysis variations:

- Automated template Differential Power Analysis
- High-order Differential Power Analysis

In general, Differential Power Analysis can be used to break any symmetric or asymmetric algorithm

- Asymmetric operations tend to produce stronger signals leaking than symmetric ones.

Differential Power Analysis
Preventing Differential Power Analysis:

- Reduce signals size
- Introducing noise into power consumption measurements
- Designing cryptosystems with realistic assumptions about the underlying hardware.
  - Balanced hardware and software (i.e. leak tolerant design)
  - Incorporating randomness
  - Algorithm and protocol-level countermeasures