Introduction to Cryptography
Lecture 3

DES
Meet in the middle attack
Differential cryptanalysis
Message authentication

Benny Pinkas

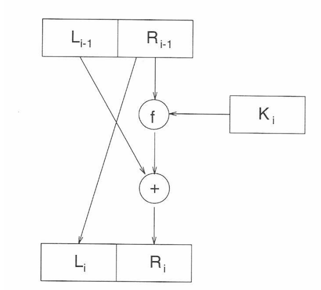
Books

- Open University book in Hebrew (based on Stinson's book)
- Lecture notes from Bar Ilan http://www.cs.biu.ac.il/~lindell/89-656/main-89-656.html

Feistel Networks

- Encryption:
- *Input:* P = $L_{i-1} \mid R_{i-1}$. $|L_{i-1}|=|R_{i-1}|$
  - $L_i = R_{i-1}$
  - $R_i = L_{i-1} \oplus F(K_i, R_{i-1})$
- Decryption?

- No matter which function is used as F, we obtain a permutation (i.e., F is reversible).



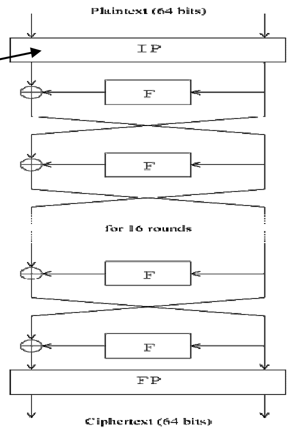
DES (Data Encryption Standard)

- A Feistel network encryption algorithm:
  - How many rounds?
  - How are the round keys generated?
  - What is F?

- DES (Data Encryption Standard)
  - Designed by IBM and the NSA, 1977.
  - 64 bit input and output
  - 56 bit key
  - 16 round Feistel network
  - Each round key is a 48 bit subset of the key
- Throughput ≈ software: 10Mb/sec, hardware: 1Gb/sec (in 1991!).
- Criticized for unpublished design *decisions* (designers did not want to disclose differential cryptanalysis).
- Linear cryptanalysis: about 2^{40} known plaintexts

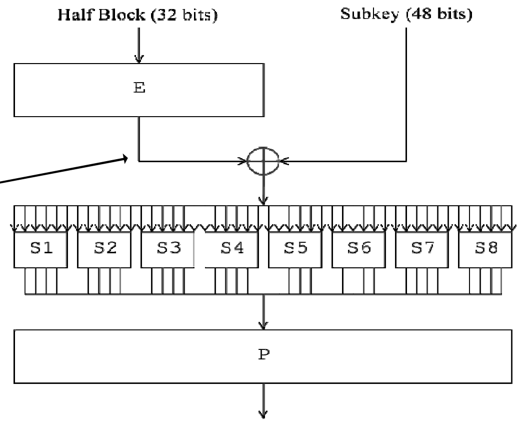
DES diagram (Data Encryption Standard)

Initial permutation:
- not secret
- makes implementations in software less efficient



DES F functions

Expansion to 48 bits

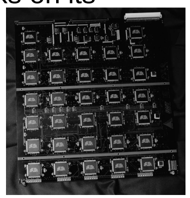


How much effort can be invested in an attack?

- Computation overhead:
  - 2^{56} computation was demonstrated to be feasible.
  - Moore’s Law: computation speed doubles every 1.5 years.
  - Attacker can use a network of machines (over the Internet?)
  - 2^{80} is considered to be the lower end of “infeasible”
  - Brute force attack on DES: 2^{56}
  - Anything more efficient is considered a “break”
- Memory:
  - Terabyte = 2^{43} bits
  - 2^n memory is probably less feasible than 2^n computation

Double DES

- DES is out of date due to brute force attacks on its short key (56 bits)



- Why not apply DES twice with two keys?
  - Double DES: DES $_{k1,k2} = E_{k2}(E_{k1}(m))$
  - Key length: 112 bits
- But, double DES is susceptible to a meet-in-the-middle attack, requiring ≈ 2^{56} operations and storage.
  - Compared to brute a force attack, requiring 2^{112} operations and O(1) storage.

Meet-in-the-middle attack

- Meet-in-the-middle attack
  - $c = E_{k2}(E_{k1}(m))$
  - $D_{k2}(c) = E_{k1}(m)$
- The attack:
  - Input: *(m,c)* for which $c = E_{k2}(E_{k1}(m))$
  - For every possible value of k_1, generate and store $E_{k1}(m)$
  - For every possible value of k_2, check if $D_{k2}(c)$ is in the table
  - Might obtain several options for (k_1,k_2). Check them or repeat the process again with a new *(m,c)* pair.
- The attack is applicable to any iterated cipher

Meet-in-the-middle attack

- The plaintext and the ciphertext are 64 bits long
- The key is 56 bits long
- Suppose that we are given two plaintext-ciphertext pairs (m,c) (m',c')
- The attack looks for k1,k2, such that $D_{k2}(c) = E_{k1}(m)$ and $D_{k2}(c') = E_{k1}(m')$
- The correct value of k1,k2 satisfies both equalities
- There are 2^{112} (actually $2^{112}-1$) other values for k1,k2.
- Each one of these satisfies the equalities with probability 2^{-128}
- The probability that there exists one or more of these other pairs of keys, which satisfy both equalities, is bounded from above by $2^{112-128} = 2^{-16}$.

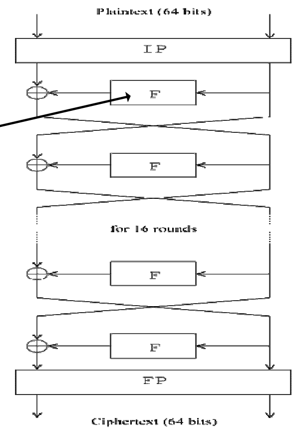
Triple DES

- $3DES_{k1,k2} = E_{k1}(D_{k2}(E_{k1}(m)))$
- Why use Enc(Dec(Enc( ))) ?
  - Backward compatibility: setting $k_1=k_2$ is compatible with single key DES
- Only two keys
  - Effective key length is 112 bits
  - Why not use three keys? There is a meet-in-the-middle attack with 2^{112} operations
- 3DES provides good security. Widely used. Less efficient.

Differential Cryptanalysis of DES

DES diagram:

S-boxes



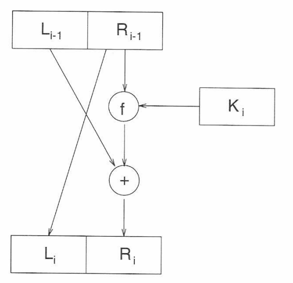
Differential Cryptanalysis [Biham-Shamir 1990]

- The first attack to reduce the overhead of breaking DES to below exhaustive search
- Very powerful when applied to other encryption algorithms

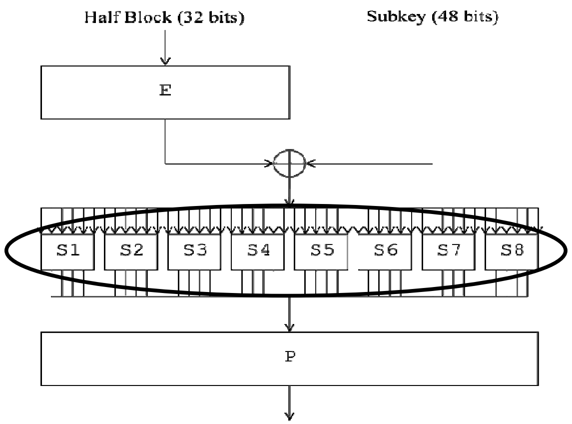
- Depends on the structure of the encryption algorithm
- Observation: all operations except for the s-boxes are linear
- Linear operations:
  - *a* = *b* ⊕ *c*
  - *a* = the bits of *b* in (known) permuted order
- Linear relations can be exposed by solving a system of linear equations

A Linear F in a Feistel Network?

- Suppose $F(R_{i-1},K_i) = R_{i-1} \oplus K_i$
  - Namely, that F is linear
- Then $R_i = L_{i-1} \oplus R_{i-1} \oplus K_i$
  $L_i = R_{i-1}$
- Write L_{16}, R_{16} as linear functions of L_0, R_0 and K.
  - Given L_0R_0 and $L_{16}R_{16}$ Solve and find K.
- F must therefore be non-linear.
- F is the only source of non-linearity in DES.



DES F functions



Source of non-linearity

Differential Cryptanalysis

- The S-boxes are non-linear
- We study the differences between two encryptions of two different plaintexts

- Notation:
  - The plaintexts are P and P*
  - Their difference is dP = P ⊕ P*
  - Let X and X* be two intermediate values, for P and P*, respectively, in the encryption process.
  - Their difference is dX = X ⊕ X*

The advantage of looking at XORs

- It's easy to predict the difference of the results of linear operations
- Unary operations, (e.g. P is a permutation of the bits of X)
  - $dP(x) = P(x) \oplus P(x^*) = P(x \oplus x^*) = P(dx)$
- XOR
  - $d(x \oplus y) = (x \oplus y) \oplus (x^* \oplus y^*) = (x \oplus x^*) \oplus (y \oplus y^*) = dx \oplus dy$
- Mixing the key
  - $d(x \oplus k) = (x \oplus k) \oplus (x^* \oplus k) = x \oplus x^* = dx$
  - The result here is key independent (the key disappears)

Differences and S-boxes

- S-box: a function (table) from 6 bit inputs to 4 bit output
- X and X* are inputs to the same S-box, and we know their difference $dX = X \oplus X^*$.
- $Y = S(X)$
- When dX=0, X=X*, and therefore $Y=S(X)=S(X^*)=Y^*$, and dY=0.
- When $dX \neq 0$, $X \neq X^*$ and we don't know dY for sure, but we can investigate its distribution.
- For example,

Distribution of Y' for S1

- dX=110100
- $2^6=64$ input pairs, { (000000,110100), (000001,110101),…}
- For each pair compute xor of outputs of S1
- E.g., S1(000000)=1110, S1(110100)=1001. dY=0111.
- Table of frequencies of each dY:

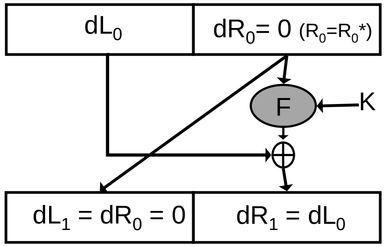
0000	0001	0010	0011	0100	0101	0110	0111
0	8	16	6	2	0	0	12
1000	1001	1010	1011	1100	1101	1110	1111
6	0	0	0	0	8	0	6

Differential Probabilities

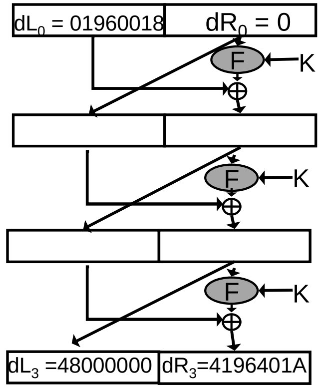
- The probability of $dX \Rightarrow dY$ is the probability that a pair of difference dX results in a pair of difference dY (for a given S-box).
- Namely, the entries in the table divided by 64.
- Differential cryptanalysis uses entries with large values
  - $dX=0 \Rightarrow dY=0$
  - Entries with value 16/64.

Warmup

Inputs: L_0R_0, $L_0{*}R_0{*}$, s.t. $R_0=R_0{*}$.
Namely, inputs whose xor is dL_0 0

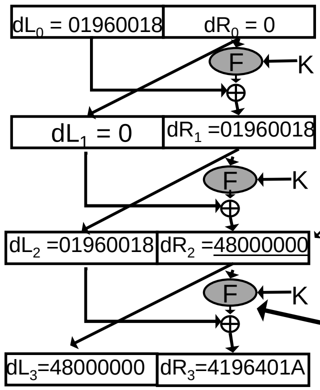


3 Round DES



The attacker knows the two plaintext/ciphertext pairs, and therefore also their differences

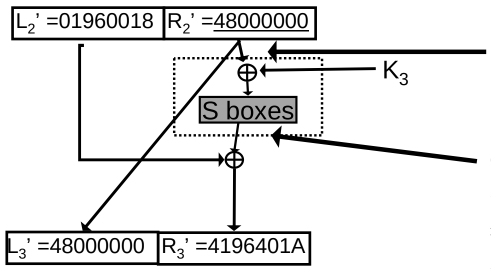
Intermediate differences equal to plaintext/ciphertext differences



Note that here the adversary also knows the actual two values

```
dF = 4196401A
⊕     01960018
=     40004002
```


Finding K



The actual two inputs to F are known

Output xor of F (i.e., S boxes) is 40004002

⇒Table enumerates options for the pairs of inputs to S box

Find which K_3 maps the inputs to an s-box input pair that results in the output pair!

DES with more than 3 rounds

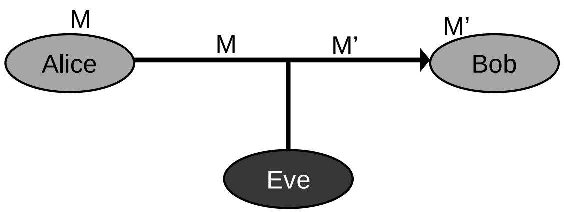
- Carefully choose pairs of plaintexts with specific xor, and determine xor of pairs of intermediate values at various rounds.
- E.g., if dL_0=40080000_x, dR_0=04000000_x
  Then, with probability ¼, dL_3=04000000_x, dR_3=4008000_x
- 8 round DES is broken given 2^{14} chosen plaintexts.
- 16 round DES is broken given 2^{47} chosen plaintexts...

AES (Advanced Encryption Standard)

- Design initiated in 1997 by NIST
  - Goals: improve security and software efficiency of DES
  - 15 submissions, several rounds of public analysis
  - The winning algorithm: Rijndael
- Input block length: 128 bits
- Key length: 128, 192 or 256 bits
- Multiple rounds (10, 12 or 14), but does not use a Feistel network

Data Integrity, Message Authentication

- Challenge: an *active* adversary might change messages exchanged between Alice and Bob



- Authentication is orthogonal to secrecy. A relevant challenge regardless of whether encryption is applied.

One Time Pad

- OTP is a perfect cipher, yet provides no authentication
  - Plaintext $x_1x_2\ldots x_n$
  - Key $k_{1k2}\ldots k_n$
  - Ciphertext $c_1=x_1\oplus k_1$, $c_2=x_2\oplus k_2,\ldots,c_n=x_n\oplus k_n$
- Adversary changes, e.g., c_2 to $1\oplus c_2$
- User decrypts $1\oplus x_2$
- Error-detection codes are insufficient. (For example, linear codes can be changed by the adversary, even if encrypted.)