

ISA 666 Internet Security Protocols

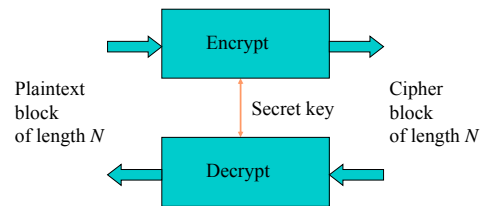
Secret Key Cryptography

Agenda

- Generic Block Cipher
- DES
- Modes of Block Ciphers
- Multiple Encryptions
- Message Authentication through Secret Key Cryptography.

Generic Block Cipher

Generic Block Cipher



Generic Block Encryption (Cont'd)

- Convert one block to another: **one-to-one**
- Block should be long enough to avoid known-plaintext attack, but not too long (performance).
 - 64 bit typical
- Key should be long enough to defeat brute-force attack
- Output should look random
 - No obvious correlation between plaintext and ciphertext
 - Bit spreading

Generic Block Encryption (Cont'd)

- By substitution:
 - Need to know how to substitute each plaintext message.
 - How many bits are needed for specifying random substitution of k -bit blocks: $k \cdot 2^k$ bits
 - Why?
 - There are 2^k possible bit sequences of length k .
 - For each one, need to specify a bit vector of length k .

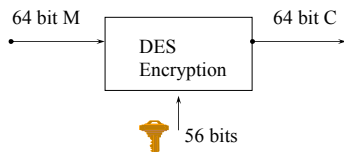
Generic Block Encryption (Cont'd)

- By Permutations:
 - Need to know which position each bit is placed.
 - How many bits are needed for specifying random permutation of k -bit blocks = $k \cdot \log_2 k$
 - How many possible position does each bit has = k (1,..,k)
 - How many bits are needed to encode k positions = $\log_2 k$
- By combining Substitutions and Permutations
 - How about $S \rightarrow P \rightarrow S \rightarrow S \rightarrow P \rightarrow \dots$
 - How about $S \rightarrow P \rightarrow P \rightarrow S \rightarrow \dots$

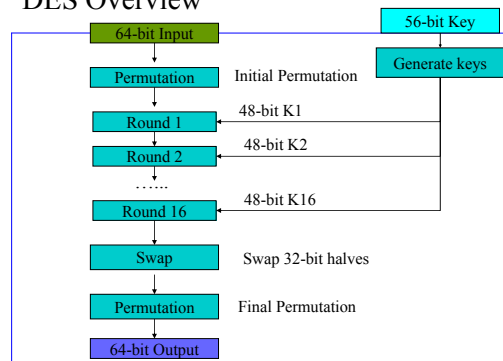
DES (Data Encryption Standard)

DES (Data Encryption Standard)

- Published in 1977, standardized in 1979, expired in 1998.
- Key: 64 bit quantity=8-bit parity+56-bit key
 - Every 8th bit is a parity bit.
- 64 bit input, 64 bit output.



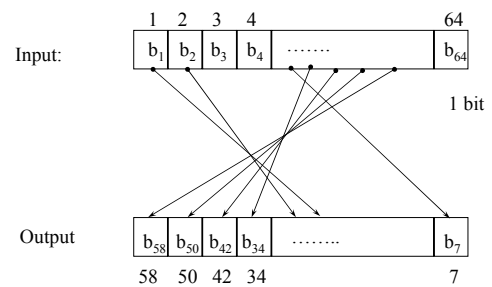
DES Overview



Initial and Final Permutations

- Initial permutation (IP)
- View the input as M : 8×8 bit matrix
- Transform M into $M1$ in two steps
 - Transpose row x into column $(9-x)$, $1 \leq x \leq 8$ (equivalent to 90° clockwise turn of the matrix)
 - Apply permutation on the rows:
 - For even row y , it becomes row $y/2$
 - For odd row y , it becomes row $(5+y/2)$
- Final permutation $FP = IP^{-1}$
 - Why?

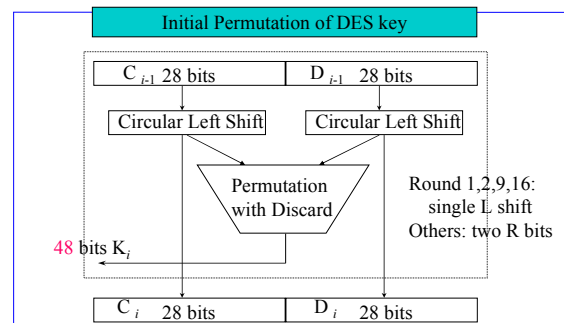
Initial Bit Permutation (1-to-1)



Per Round Key Generation

- Initial Key has 64 bits
- Remove Every 8th bit:
 - Remove 8, 16, 24, 32, 40, 48 (parity check)
- End up with 56 bits:
 - Now do an initial permutation of the 56 bit key:
 - First half (28 bits) = C_0
 - Second half (28 bits) = D_0

Per-Round Key Generation

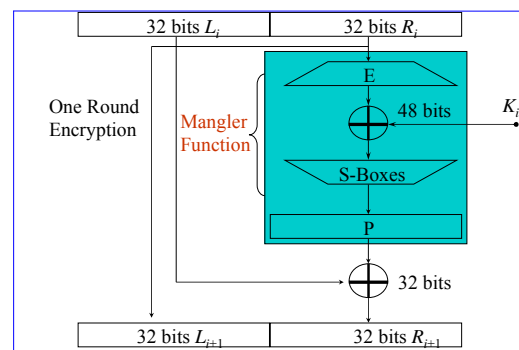


Round (i)

- 64 bit input broken down into two halves
- L_i and R_i (32 bits each)
- Recursively define:
- $L_{i+1} = R_i$
- $R_{n+1} = \text{mangler}(R_n, k_n)(+)L_n$

Pictorially

A DES Round (i)

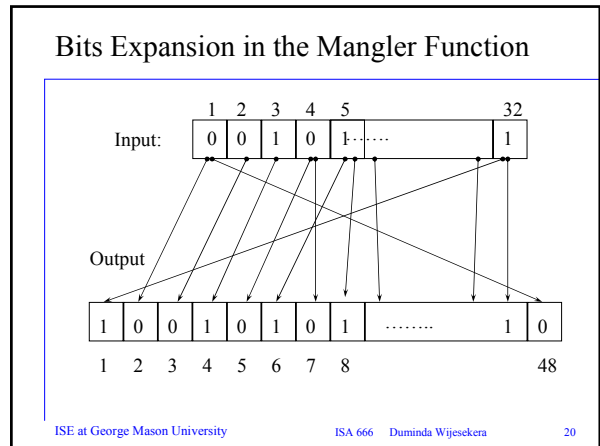
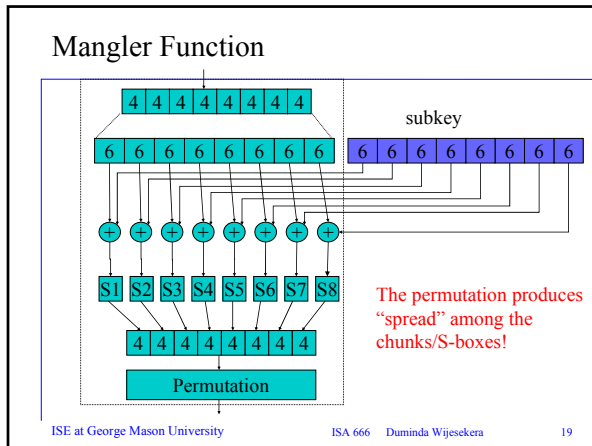


Important Properties of DES Round

- The decryption in a DES round does NOT require the mangler function to be reversible!
- The decryption of 64 bit block in a DES round is equivalent to encryption of the 64 bit block (by swapping the 32 bit halves) with the same key
 - if $E(L_i, R_i, K_i) = (L_{i+1}, R_{i+1})$, then
 - $D(L_{i+1}, R_{i+1}) = E(R_{i+1}, L_{i+1}, K_i) = (R_i, L_i)$
 - Why?

Answer

- Reverse the i^{th} step = compute L_{n+1} and R_{n+1} from L_n and R_n
- Suppose we know L_{n+1} and R_{n+1} , then:
- $R_n = L_{n+1}$
- Know $R_{n+1} = \text{mangler}(R_n, k_n)(+)L_n$
 - Compute $\text{mangler}(R_n, k_n) = \text{mangler}(L_{n+1}, k_n)$
 - Then $L_n = \text{mangler}(R_n, k_n)(+)R_{n+1}$

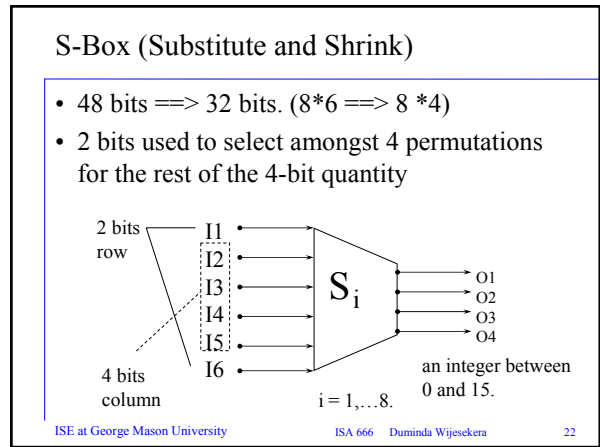


E Box of DES

- How is the E Box defined?

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

ISE at George Mason University ISA 666 Duminda Wijesekera 21



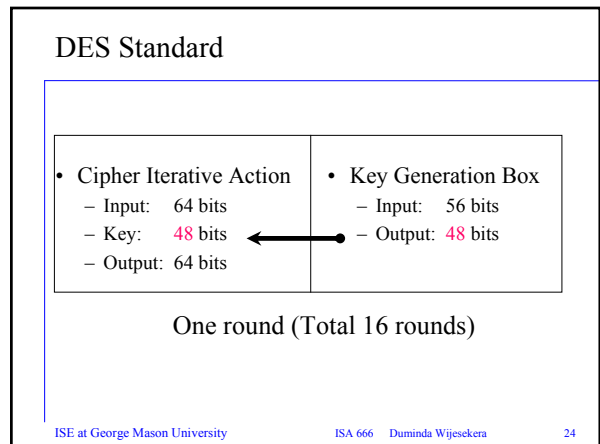
S-Box (Cont'd)

Each row and column contain different numbers.

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

Example: input: 100110 output: ???

ISE at George Mason University ISA 666 Duminda Wijesekera 23



Avalanche Effect

- A small change in either the plaintext or the key should produce a significant change in the ciphertext.
- DES has a strong avalanche effect.
- Example
 - Plaintexts: 0X0000000000000000 and 0X8000000000000000
 - Same key: 0X016B24621C181C32
 - 34 bits difference in cipher-texts
 - Similar result with same plaintext and slightly different keys

Concerns About DES

- Key space problem: 56 bit key (2^{56})
 - DESCHALL recovered RSA challenge I key on June 17, 1997 (6 month into the contest)
 - \$.25m (total cost), July 15, 1998, RSA DES challenge II key recovered in 56 hours
- Cryptanalysis
 - Sixteen Weak and semi-weak keys:
 - Differential cryptanalysis require less tries using chosen plaintext/ciphertext [Biham, 1993]
 - Effective up to 15 rounds
 - DES is well designed to defeat differential analysis
 - Linear cryptanalysis requires only known plaintext/ciphertext [Matsui, 1993]

DES Summary

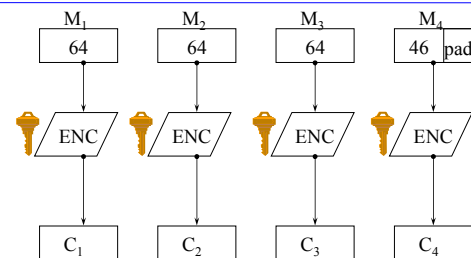
- Simple, easy to implement:
 - Hardware/gigabits/second, software/megabits/second
- 56-bit key DES maybe acceptable for non-critical applications but triple DES (3-DES) should be secure for most applications today
- Supports several operation modes: ECB CBC, OFB, CFB

Modes of Block Cipher Operations

Encrypting a Large Message

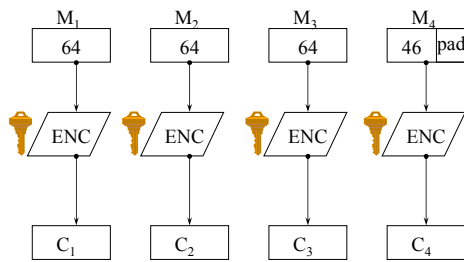
- Modes of block cipher operations
 - ECB (Electronic Code Book)
 - CBC (Cipher Block Chaining Mode)
 - OFB (Output Feedback Mode)
 - CFB (Cipher Feedback Mode)

Electronic Code Book (ECB)



Divide and conquer !

ECB Properties

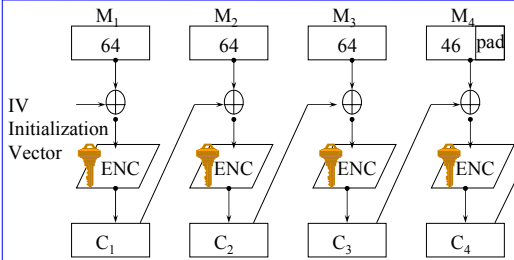


$(M_1 == M_3) \Rightarrow (C_1 == C_3)$ giving some information to eavesdroppers

ECB Properties (Cont'd)

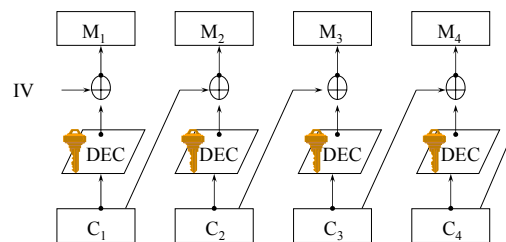
- Cipher block substitution and rearrangement attacks
 - fabrication of specific information
- Advantage: No error propagation.
- **Two serious flaws, one advantage!**

Cipher Block Chaining (CBC)



$(M_1 == M_3)$ very unlikely leads to $(C_1 == C_3)$

CBC Decryption

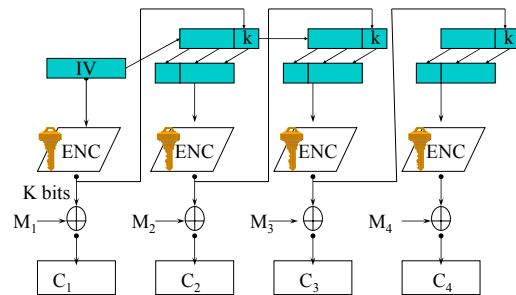


CBC Properties

- Chaining dependency
 - Each ciphertext block depends on all preceding plaintext blocks
- Error propagation
 - Each error in c_j affects decipherment of c_j and c_{j+1} .
 - Predictable bit change in m_{j+1} by alert corresponding bits of c_j .
- Error recovery
 - An error in c_j doesn't propagate beyond c_{j+1} .
 - Can recover from loss of cipher text blocks.

Output Feedback Mode (OFB)

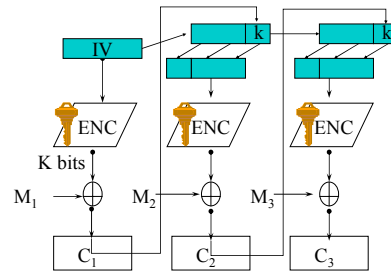
Like a Random Number Generator...



OFB Properties

- Chaining dependencies
 - Key stream is plaintext-independent
 - Allow pre-computing of pseudo-random stream (One-Time Pad); XOR can be implemented very efficiently
- No error propagation problem as in CBC
- Error recovery
 - Can recover from bit error
 - But not from block loss.
- **If the attacker knows the plaintext, he can change the ciphertext by XORing it with the plaintext and then XORing with whatever he wants to transmit.**

Cipher Feedback Mode (CFB)



CFB Properties

- Chaining dependencies
 - Ciphertext block c_j depends on all preceding plaintext blocks.
- Error propagation
 - Bit error in one ciphertext block affects the next several blocks
- Error recovery
 - Can recover from bit errors after several blocks
 - Can resynchronize after loss of blocks.
- **Secure against known plaintext attack (plaintext substitution)**
- Less vulnerable to tampering with ciphertext - cipher C_i 's impact on m_{i+1} is subtle (through encryption function) and thus less predictable

Multiple Encryption

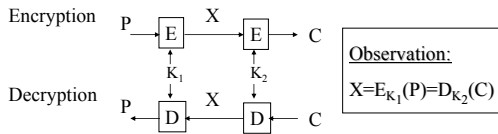
Triple DES

- Major limitation of DES
 - Key length is too short (56 bits).
- Question: Can we apply DES multiple times to increase the strength of encryption?
 - Advantage: preserve the existing investment in software and equipment.

Triple DES (Cont'd)

- Double DES
 - Encrypt the plaintext twice with two different DES keys
 - Key length increases to 112 bits
- Two concerns
 - Is DES a group?
 - $E_{k_2}(E_{k_1}(P)) = E_{k_3}(P)$
 - Implication?
 - Meet-in-the-middle attack

Meet-in-the-middle attack

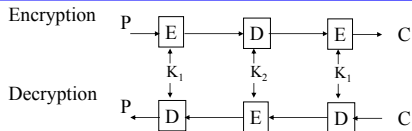


- For a known pair (P,C)
 - Encrypt P for all 2^{56} values for K_1
 - Store the results in a table sorted by the value of X
 - Decrypt C for all 2^{56} values for K_2 , and for each result check the table
 - A match reveals a possible combination of key

Meet-in-the-middle attack (Cont'd)

- Analysis
 - With one pair (P_1, C_1) , # of $\langle K_1, K_2 \rangle$ that satisfies $E_{K_1}(P_1) = D_{K_2}(C_1)$ is $2^{112}/2^{64} = 2^{48}$.
 - Given a pair of keys $\langle K_1, K_2 \rangle$ and another pair $\langle P_2, C_2 \rangle$, the probability that it satisfies $E_{K_1}(P_2) = D_{K_2}(C_2)$ is $2^{48}/2^{64} = 2^{-16}$.
 - The probability that pair of keys $\langle K_1, K_2 \rangle$ satisfies $E_{K_1}(P_3) = D_{K_2}(C_3)$ is $2^{-16}/2^{64} = 2^{-80}$.
 - The false positive probability after matching 3 pairs of known plaintext is only 2^{-80} !
- Goal of double DES
 - Increase the difficulty of exhaustive key search (2^{112} keys)
 - In effect, the effort is on the order of 2^{56} .

Triple DES (Cont'd)



- Apply DES encryption/decryption three times.
 - With two keys or three keys
- Why E-D-E?
 - It's not clear if DES is a group when this was proposed.
 - If one key is used, it's equivalent to doing DES once.

Triple DES Is Not Ideal...

- Efficiency demands schemes with longer keys to begin with!
- Triple DES runs one third as fast as DES on the same platform
- New candidates are numerous - RC5, IDEA, two-fish, CAST, etc
- New AES

Message Authentication through Secret Key Algorithms

Message Authentication

- Message authentication is the process to verify that received messages come from the alleged source and have not been altered.
- The goals of message authentication is to prevent
 - Masquerade: insertion of messages from a fraudulent source.
 - Content modification: change of messages
 - Sequence modification: insertion, deletion and reordering of messages.
 - Timing modification: delay or replay of messages.

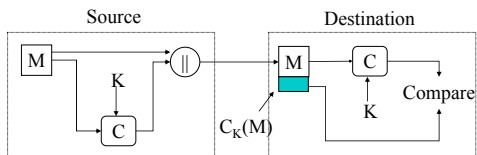
Message Authentication Functions

- Message encryption
 - Use the structure or pattern in the plaintext
 - Accept the decrypted plaintext if it is in an intelligible form.
 - No guarantee!
- Message Authentication Code (MAC)
- Hash function

Message Authentication Code (MAC)

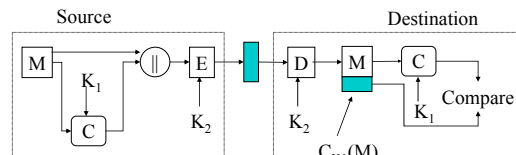
- MAC
 - Also known as cryptographic checksum, Message Integrity Code (MIC).
 - **Assumption: the sender and the receiver share a common secret key.**
 - A small fixed-size block generated from the message with secret key cryptography.
 - Usually appended to the original message.

MAC (Cont'd)



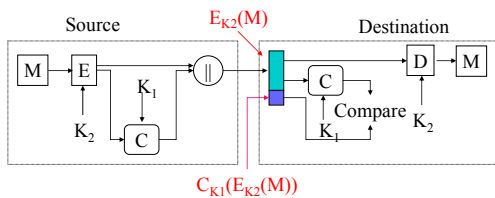
- Mode I
 - Message authentication
 - No confidentiality

MAC (Cont'd)



- Mode II
 - Message authentication and confidentiality
 - Authentication tied to plaintext

MAC (Cont'd)



- Mode III
 - Message authentication and confidentiality
 - Authentication tied to ciphertext

Requirements for MAC

- For M and $C_K(M)$, it's computationally infeasible to construct a message M' such that $C_K(M') = C_K(M)$.
- $C_K(M)$ should be uniformly distributed in terms of M
 - For any two messages M and M' , $\Pr[C_K(M) = C_K(M')] = 2^{-n}$, where n is the number of bits in the MAC.
 - Intuition: prevent chosen plaintext attack.
- If M' is equal to some known transformation on M , then $\Pr[C_K(M) = C_K(M')] = 2^{-n}$.
 - Intuition: no weak spot with respect to certain bits of the message.

MAC Based on DES CBC Residue

- Known as Data Authentication Algorithm
- DES CBC mode with IV being zero.
- A message is padded with zeroes to form 64-bit blocks.
- The data authentication code (DAC, i.e., the MAC) consists of either the entire last ciphertext block or the left M bits with $16 \leq M \leq 64$.

MAC Based on DES CBC Residue (Cont'd)

