Topic 3

Symmetrical Cryptography

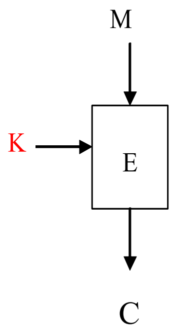
Understand the principles of modern symmetric (conventional) cryptography

Overview

- Block Cipher Design
- Standardised Block Ciphers
  - Data Encryption Standard – DES and 3DES
  - Advanced Encryption Standard – AES
- Use of Block Ciphers in Real World – Modes of Operations
- Block Ciphers vs Stream Ciphers
- Conclusion

source: chapters 4, 6 and 7 of Cryptography and Network Security

Block Cipher Design



- Block Ciphers
  - Plaintext is divided into blocks of fixed length and blocks are encrypted one at a time.
  - Here, we have $C = E(K,M)$, or $C = E_K(M)$, where
    - M is the plaintext block
    - K is the secret key
    - E is the encryption function
    - C is the ciphertext
  - We also have a decryption function, $D_K(\cdot)$, such that $M = D_K(C)$.

Block Cipher Design - Design Criteria

- Completeness
  - Each bit of the output should depend on every bit of the input and every bit of the key.
- Avalanche effect
  - Changing one bit in the input should change many bits in the output.
  - Also, changing one key bit should result in the change of many bits in the output.
- Statistical independence
  - Input and output should appear to be statistically independent.

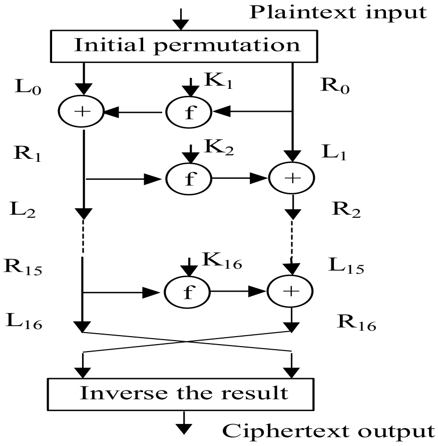
Block Cipher Design

- A complex encryption function can be built out of some simple operations (round function) by repeatedly using them.
- Examples of simple operations:
  - XOR
  - modular multiplication
  - substitutions
  - permutations
- Feistel block cipher is an example implementation of this principle.
- Ciphers that use substitution and permutation are called substitution-permutation (S-P) networks.
- They can be efficiently implemented on both hardware and software platforms.

Block Cipher Design - Feistel Block Cipher

- Operation overview
  - Initial permutation of bits.
  - Split in left and right half.
  - 16 rounds of identical operations, but each round uses a different subkey.
  - Inverse initial permutation.

Block Cipher Design - Feistel Block Cipher

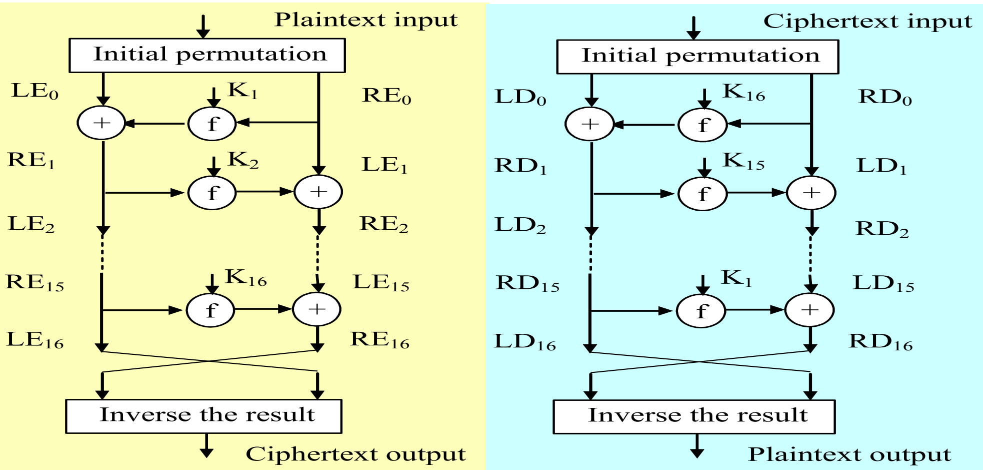


Encryption:
r rounds (for DES, r=16)
Plaintext = (L_0, R_0)

For $1 <= i <= r$
$L_i = R_{i-1}$
$R_i = L_{i-1}$ **xor** $f(R_{i-1}, K_i)$
Subkeys K_i is derived from key K
Ciphertext = (R_r, L_r)

Decryption:
Is the same as the encryption process except that the subkeys are applied in a reverse order.

Block Cipher Design - Feistel Block Cipher



Block Cipher Design - Feistel Block Cipher

- Round function *f*:
  - Typically use permutations, substitutions, modular arithmetic.
  - Takes a *n*-bit block to a *n*-bit block.
  - Each use of the round function employs a subkey derived from *K*.
- Block size, *n*
  - larger block sizes mean greater security but make encryption/decryption slower; typically 128-bit or 256-bits.
- Key size, *s*
  - larger key size means greater security but reduced speed; 128-bits size has become a norm.
- Number of rounds, *r* (typically 10+ rounds).

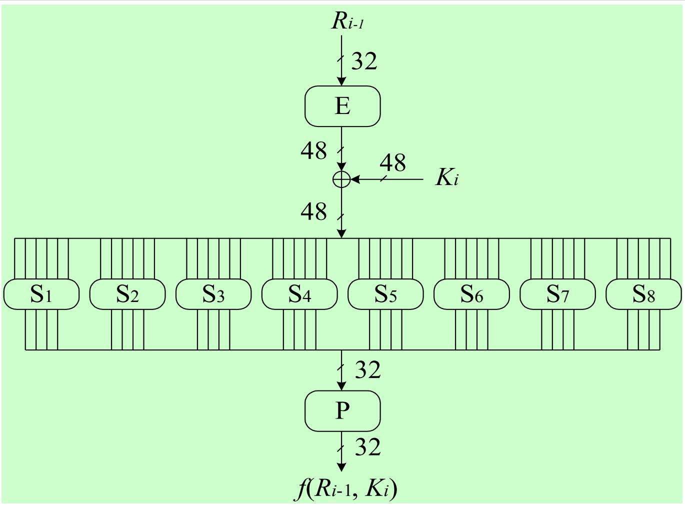
DES (Data Encryption Standard)

- First published in 1977 as a US Federal standard.
- DES is a de facto international standard for banking security.
- DES is **a Feistel block cipher** - block length is 64 bits, key K is 56 bits.
- The subkeys $k_1, k_2, \ldots, k_{16}$ are each 48-bits, generated from key K.
- The DES decryption algorithm is the same as the encryption one; the only difference is that the keys for each round must be used in the reverse order, i.e. k_{16} first and k_1 last.

DES – Round function, f

- Step 1 - Expansion Permutation:
  - Right half (32 bits) is expanded (and permuted) to 48 bits.
  - Diffuses relationship of input bits to output bits.
- Step 2 - Use of Round Key:
  - 48 bits are XOR-ed with the round key (48 bits).
- Step 3 - Splitting:
  - Result is split into **eight** lots of six bits each.
- Step 4 - S-Box: (S = Substitution)
  - Each six bit lot is used as an index to an S-box to produce a four-bit result.
- Step 5 - P-Box: (P = Permutation)
  - 32 bits output from 8 S-Boxes are permuted = the output of *f*.

DES – Round function, f



DES

- S-Box Operation
  - Each of the 8 different S-boxes is a table of 4 rows and 16 columns.
  - The 6 input bits specify which row and column, i.e. a cell, to use.
    - Bits 1 and 6 select the row.
    - Bits 2-5 select the column.
  - The decimal value in the cell is then converted into a 4-bit result, which is the output from the S-box.
- Efficient to encrypt/decrypt, so mainly used for encryption of **message contents - confidentiality**.
- The algorithm public, but the design principles are kept secret. Built-in trapdoors might be placed in secret boxes.

DES - Strength

- Its weakness is 56-bit key - which is good enough to deter casual DES key browsing, but not for a dedicated adversary.
- Use of a 56-bit key - can be broken on average in 2^{55} (i.e. $3.6 * 10^{16}$) trials.

trials/second	time required
1	10^9 years
10^3	10^6 years
10^6	10^3 years
10^9	1 year
10^{12}	10 hours

  - a DES chip does 1 million encryptions per second.
  - a million chips in parallel do 10^{12} trials per second.
- For today's computing power, key size should be at least 128 bits.
- Improvements: Triple DES (3DES), AES (Rijndael)

Triple DES

- Involves use of two or three DES keys.
- EDE2 (triple DES using two keys)
  - EDE2 uses two DES keys (K_1, K_2), encryption algorithm E, and decryption algorithm D, i.e. $\boldsymbol{C=E_{K1}(D_{K2}(E_{K1}(M)))}$
  - So the key length is 112-bits.
  - The use of D here does not have any security implication; it just makes triple-DES backward compatible with single DES if $K_1=K_2$.
- EDE3 (triple DES using three keys)
  - Liked by some; EDE3 uses three keys, $\boldsymbol{C=E_{K3}(D_{K2}(E_{K1}(M)))}$**;** the key length is 168 bits.

BUT owning to the meet-in-the-middle attack, the effective key lengths for both cases are much shorter.

AES - Background

- In 1997, NIST call for algorithms to replace DES.
  - Algorithm and implementation characteristics – fast & low resource consumption;
  - Cost - fast in both hardware and software;
  - Designers
    - Vincent Rijmen, Joan Daemen → Rijndael.
- Open process: international submissions and public comments.
- In 2001, Rijndael was formally nominated as the Advanced Encryption Standard (AES).
- Website: http://www.nist.gov/aes/

AES – Overview

- Like DES, AES is a symmetric block cipher.
  - The same key is used to encrypt and decrypt the message.
  - The plaintext and the ciphertext have the same size.
- Block size is 128 bits (others are allowed but not recognised by the standard).
- The key lengths are 128, 192, or 256 bits, i.e. the standard comprises **three** block ciphers**, AES-128, AES-192** and **AES-256.**
- It is a **substitution-permutation** cipher involving *r* rounds:
  - for key length=128 bits, r=10;
  - for key length =192 bits, r=12; and
  - for key length =256 bits, r=14.

AES – Overview

- Round transformation consists of:
  - Byte substitution.
  - Shift rows.
  - Mix columns.
  - Round key addition.
- Sequential and light-weight key schedule.

AES – Basic structure

- ❑ AES has a fixed block size of 128 bits (16 bytes) called a *state*,
- ❑ e.g.

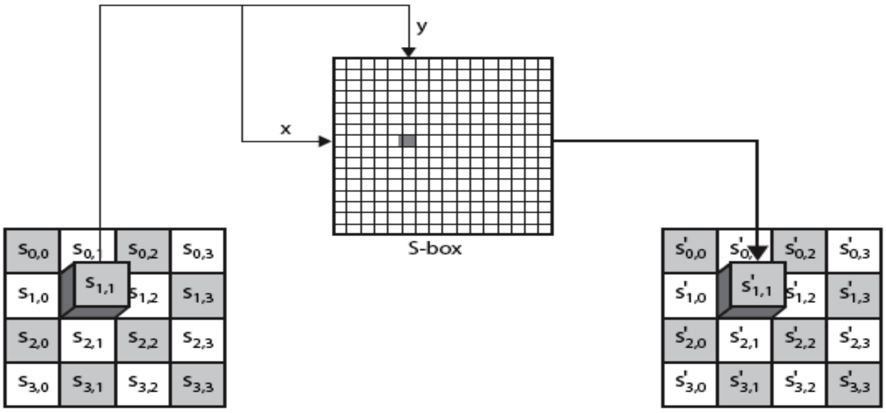
ABCDEFGHIJKLMNOP

A	E	I	M		41	45	49	4D
B	F	J	N	ASCII →	42	46	4A	4E
C	G	K	O		43	47	4B	4F
D	H	L	P		44	48	4C	50

AES – Substitute Bytes

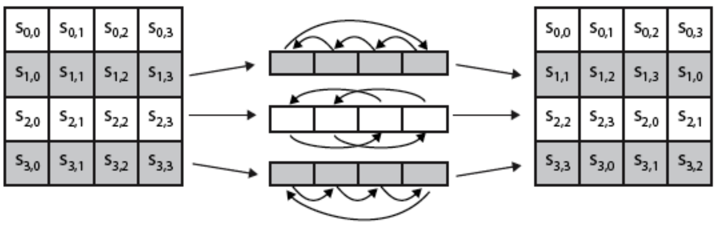
- ❑ This SubBytes transformation is a simple table lookup.
- ❑ Only one S-Box for the whole cipher, a 16 × 16 matrix of byte values, that contains a permutation of all possible 256 8-bit values.
- ❑ Each individual byte of **State** is mapped into a new byte in this way:
  - ❍ Leftmost 4 bits of the byte are used as a row value; rightmost 4-bits used as a column value;
  - ❍ these row and column values serve as indexes into the S-box to select a unique 8-bit output value.

AES – Substitute Bytes



AES – Shift Rows

- This ShiftRows transformation is a simple permutation (circular byte shift) process:
  - 1st row: not altered;
  - 2nd row: a 1-byte circular left shift;
  - 3rd row: a 2-byte circular left shift;
  - 4th row: a 3-byte circular left shift.
- Decryption uses shifts to right.
- This step permutes bytes between the columns.



AES – Mix Columns

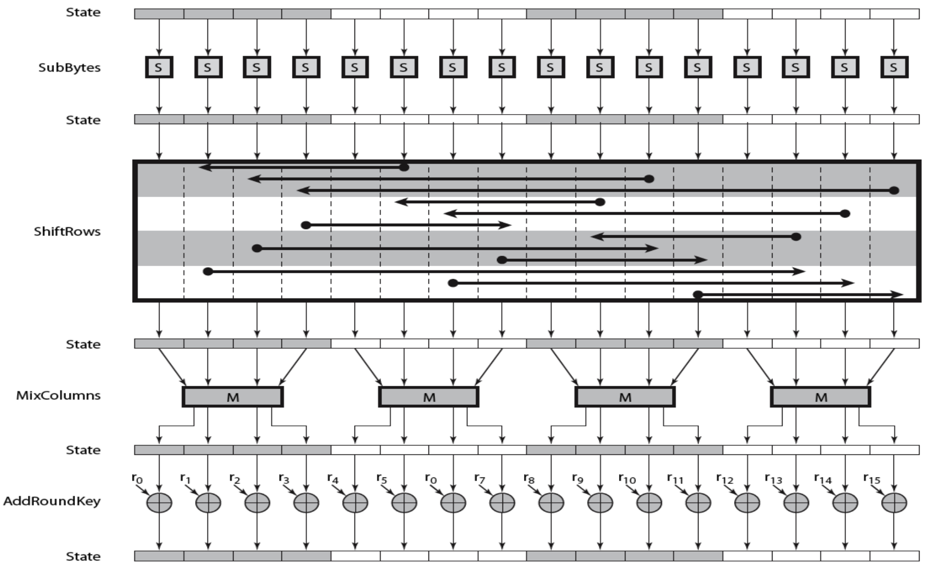
- The MixColumns transformation operates on each column individually.
- Each byte of a column is mapped into a new value that is a function of all four bytes in the column; the transformation is performed in GF(2^8).
- **This with shiftRows** transformation provides **diffusion**.

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \end{bmatrix} = \begin{bmatrix} s'_{0,0} & s'_{0,1} & s'_{0,2} & s'_{0,3} \\ s'_{1,0} & s'_{1,1} & s'_{1,2} & s'_{1,3} \\ s'_{2,0} & s'_{2,1} & s'_{2,2} & s'_{2,3} \\ s'_{3,0} & s'_{3,1} & s'_{3,2} & s'_{3,3} \end{bmatrix}$$

AES – Add Round Key

- In this AddRoundKey transformation, each byte of the state is combined with the round key using XOR, i.e. the 128 bits of state are bitwise XORed with the 128 bits of the round key.
- The round key is derived from the cipher key using a key schedule.

AES – One Round Operation



AES – Pseudo code

- **AES-128 (Encryption):**
  AddRoundKey(S,K[0]); K[0] is the cipher key, K, and other round keys are expanded from K.

  for (i = 1; i <= 9; i++)
  {
  SubBytes(S);
  ShiftRows(S);
  MixColumns(S);
  AddRoundKey(S,K[i]);
  }

  SubBytes(S);
  ShiftRows(S);
  AddRoundKey(S,K[10]).

AES-128 (Decryption) (first apply InvMixColumns to the round key)
AddRoundKey(S,K[10]);
for (i = 9; i >= 1; i--)
{
InvSubBytes(S);
InvShiftRows(S);
InvMixColumns(S);
AddRoundKey(S,K[i]);
}

InvSubBytes(S);
InvShiftRows(S);
AddRoundKey(S,K[0]).

DES versus AES

- DES:
  - Substitution-Permutation, iterated cipher, Feistel structure.
  - 64-bit block size, 56-bit key size.
  - 8 different S-boxes.
  - design optimised for hardware implementations.
  - closed (secret) design process.

- AES:
  - Substitution-Permutation, iterated cipher.
  - 128-bit block size, 128-bit (192, 256) key sizes.
  - 1 S-box.
  - design optimised for byte-orientated implementations.
  - open design and evaluation process.

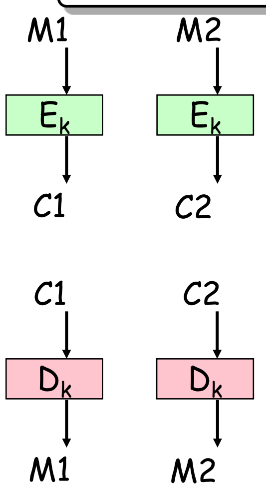
Other Symmetrical Cryptosystems

Algorithms	**Mode (block length in bits)**	**Key length (bits)**
DES	Block cipher (64)	56
Triple DES	Block cipher (64)	168 (=3*56) (112 effective)
Rijndael	Block cipher (128, 192, or 256)	128, 192, or 256
Blowfish	Block cipher (64)	Variable up to 448
IDEA	Block cipher (64)	128
RC5	Block cipher (32, 64, 128)	Variable up to 2040

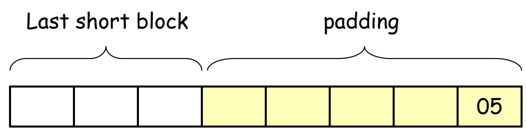
Modes of Operation – encrypting large messages

- If message is longer than block size, block cipher can be used in a variety of ways to encrypt the plaintext.
- There are a number of modes of operations; we here cover **three** of them:
  - ECB – Electronic Code Book mode
  - CBC – Cipher Block Chaining mode
  - CTR – Counter mode
- These modes of operation have been standardised internationally and are applicable to any block ciphers.

Modes of Operation - ECB mode



- $C_n = E_k(M_n)$ (or $E(K, M_n)$);
- $M_n = D_k(C_n)$; $n = \{1, 2, \ldots\}$.
- Each block is encrypted independently using the same key. The last block should be padded if necessary.
- Usually the last byte indicates the number of padding bytes added; this allows the receiver to remove the padding.



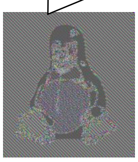
Modes of Operation - ECB mode

- Blocks are encrypted independently of other blocks
  - Reordering ciphertext blocks result in correspondingly reordered plaintext blocks.
  - Ciphertext blocks can be cut from one message and pasted in another, so block replay or block insertion (or deletion) attacks may go undetected.
- The same block of plaintext always produces the same ciphertext (with the same key)
  - patterns in plaintext show up in ciphertext.
- Error propagation: one bit error in a ciphertext block affects only the corresponding plaintext block.
- Not recommended for messages longer than one block, or if keys are reused for more than one block.

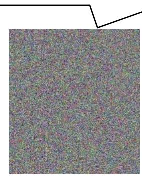
Modes of Operation - ECB mode

Plaintext

ECB encrypted cipher text

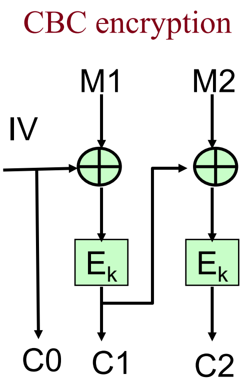


Encrypted using CBC mode



- Source: http://en.wikipedia.org/wiki/Block_cipher_modes_of_operation

Modes of Operation - CBC mode

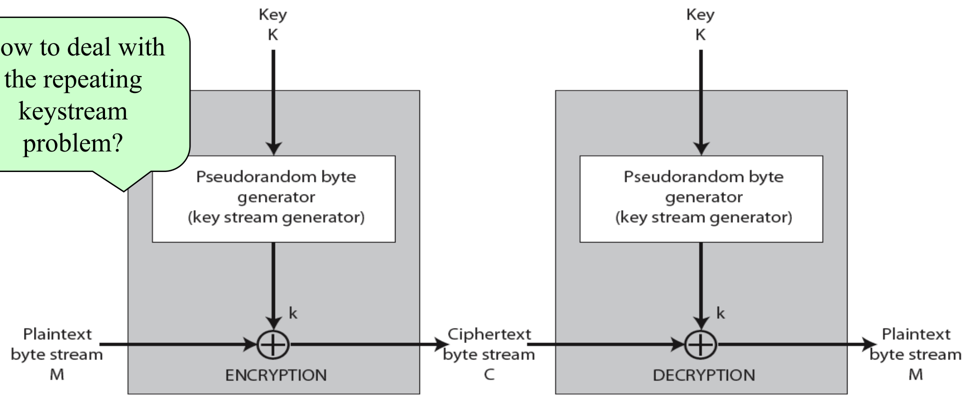


- Plaintext is divided into blocks, and the last block is padded if necessary.
- $C_n = E_K(M_n \textbf{ xor } C_{n-1})$, where C_0 = IV (Initialization vector).
- **In this example, the plaintext is *M1M2*, and the ciphertext is *C0C1C2*.**
- Ciphertext block Cj depends on Mj and all preceding plaintext blocks.
  - any repeated patterns in the plaintext are concealed by the feedback.
- Using different *IV*s in different encryption operations will result in: the same plaintext produces different ciphertexts.

Recall this slide from Topic 2: stream cipher

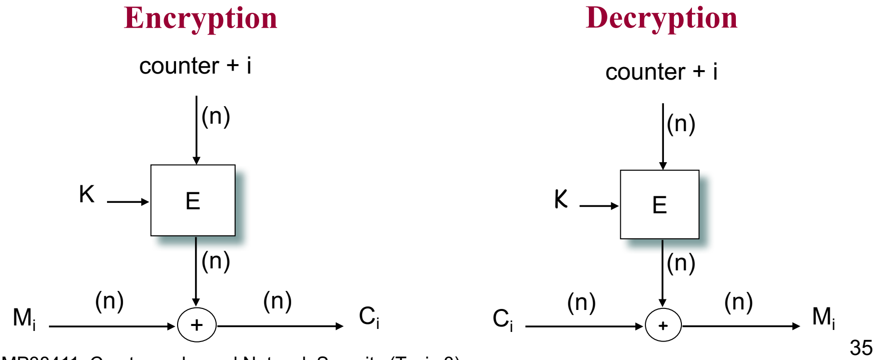
Generate a keystream from a short key that initializes the generator.

How to deal with the repeating keystream problem?



More Modes of Operation – CTR mode

- A counter, equal to the plaintext block size, is used.
- The counter value must be different for each plaintext block.
- Typically the counter is initialised to some value, and then incremented by 1 for each subsequent block (modulo 2^n, where n is the block length).



More Modes of Operation – CTR mode

- Each block can be decrypted independently of the others
  - Parallelizable.
  - Support random access.
  - The values to be XORed with the plaintext can be pre-computed.
- The counter needs to be synchronised
  - If a block is inserted into or deleted from the ciphertext stream then synchronization is lost and the plaintext cannot be recovered.
- No error propagation
  - a ciphertext block that is modified during transmission affects only the decryption of that block.

Why in CTR mode, only the encryption function of a block cipher is used (decryption is not needed)?

Block Ciphers vs Stream Ciphers

- While block ciphers encrypt blocks of characters, stream ciphers encrypt individual characters or bit streams.

- Stream ciphers
  - are usually faster than block ciphers in hardware; mostly used for continuous communications and/or real-time applications.
  - requires less memory space, so cheaper for resource restrained devices such as embedded sensors.
  - have limited or no error propagation, so advantageous when transmission errors are probable.
  - can be built out of block ciphers, e.g. by using CTR modes.

Topic 3 – A Quick Question

You have been given the equation and a block diagram for CBC encryption operation. Can you derive the equation and draw a block diagram for CBC decryption operation?

Exercise 3 (1/3)

Exercise 3 (a): The diagram on the next page illustrates an early version of the ATM (Automatic Teller Machine) solution. From the diagram, it can be seen that:

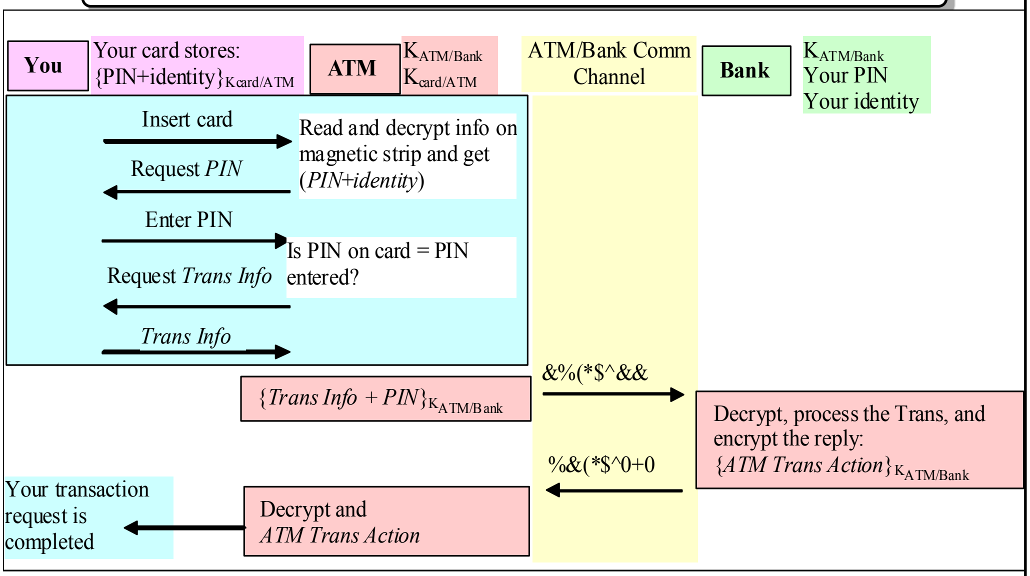
- Cash card stores the ciphertext of user's Identity and PIN that are encrypted using a symmetric key, $K_{card/ATM}$.
- The communication between ATM and bank backend office is secured using another symmetric key, $K_{ATM/Bank}$.

Answer the following questions:

(i) Identify any vulnerability in this solution, and propose a solution to address any vulnerability that you have identified.

(ii) Are there any other issues that you could identify from this application of symmetric ciphers?

Exercise 3 (2/3)



Exercise 3 (3/3)

- **Exercise 3 (b):** Use *DES(ECB)* and *DES(CBC)* modules in CrypTool, respectively, to encrypt the two messages, Msg1=‘*abcdefgh*’, and Msg2=‘*abcdefghabcdefgh*’. The encryption key is: 11 22 33 44 55 66 77 88. Compare the four ciphertexts generated. What observation(s) can you make?

- Notes: These encryption modules can be found via the menu “Encrypt/Decrypt -> symmetric (modern)”. There are comprehensive online help supplied with the tool. Simply go to the Help menu and start from the Start Page, you will learn how to use the CrypTool.

Please do explore the capability of CrypTool.

Conclusions

- Modern symmetric ciphers come in two variants: block ciphers and stream ciphers.
- The mostly used block ciphers are DES/3DES/AES; and the most recent block cipher standard is the AES - Rijndael.
- Both DES and AES obtain their security by repeated application of simple rounds consisting of substitution, permutation, shift and key addition.
- To use a block cipher one needs to also specify a mode of operation:
  - the simplest mode is ECB mode, but has problems associated with it.
  - hence a more advanced mode such as CBC mode is the default mode to use (in most commercial applications that encrypt more than one blocks).
  - CTR modes can help you to convert a block cipher into stream cipher.
- Symmetrical ciphers have a key exchange problem and do not support non-repudiation.