MANCHESTER
1824

The University of Manchester

Topic 3: Symmetric Cryptography

Understand symmetric-key (conventional) ciphers and their applications

Source: Main textbook: Chapter 2 (more detailed coverage is in Chapter 20)

Overview

- ❑ What is Cryptography and its Applications
- ❑ Symmetric-key Ciphers (cryptosystems, primitives, algorithms)
  - ❍ Block Ciphers (DES-Data Encryption Standard, 3DES, AES)
  - ❍ Stream Ciphers
  - ❍ Block Ciphers vs Stream Ciphers
- ❑ Use of Block Ciphers in Real World – Modes of Encryptions
  - ❍ ECB (Electronic Code Book) mode
  - ❍ CBC (Cipher Block Chaining) mode
  - ❍ CTR (Counter) mode
- ❑ Conclusion

What is Cryptography?

- **Cryptography** is "the art of keeping messages secure" by Schneier.

Applications of Cryptography

- Cryptography is a crucial part of cybersecurity toolbox
- Can be applied to provide the following properties/services:
  - Confidentiality (secrecy, privacy) of data in transmission & in storage.
  - Integrity of data (data authentication/authenticity) in transit & in storage.
  - Authentication of an identity (entity authentication).
  - Credential systems (a proof of qualification or competence of a person).
  - Digital signatures.
  - Electronic money (e.g. cryptocurrency, bit coins).

Applications of Cryptography

- Threshold cryptosystems (a decryption key, or a signature signing key, is shared among a group of entities and a subset of these entities (more than some threshold number) have to collaborate to perform the decryption or signature signing).
- Secure multi-party computations (e.g. multiple parties compute a function jointly, the input is from the multiple parties, but no party should learn anything rather than its own input and the final result of the computation).
- Digital right management (e.g. activation of a software license by authorized users).
- Electronic voting.
- …

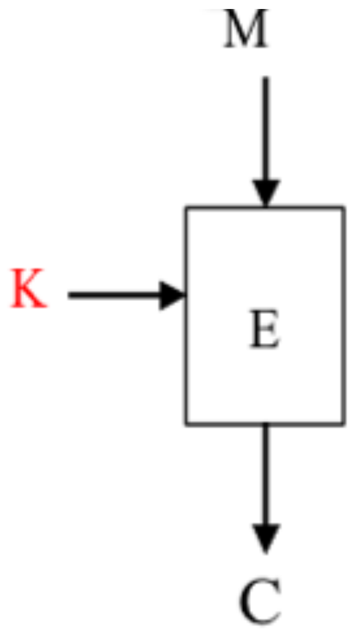
MANCHESTER
1824

The University of Manchester

Applications of Cryptography: how ciphers are used

- Ciphers or cryptographic primitives are used as building blocks to construct security methods or protocols.
- **Cryptographic algorithms/building blocks**
  - Symmetric ciphers: block ciphers and stream ciphers, e.g. DES, AES, one-time pad; same key is used.
  - Asymmetric ciphers: RSA, DSA and DH; different keys are used.
  - Hash and MACing functions: e.g. SHA256, AES-CBC.
- **Cryptographic methods/protocols**
  - Modes of encryptions, e.g. CBC (Cipher Block Chaining) mode, CTR (Counter) mode.
  - HMAC message authentication code.
  - Key distribution/agreement protocols.
  - IPSec protocol, SSL, …

Block Ciphers



- Plaintext is divided into blocks of fixed length and blocks are encrypted one at a time.
- In addition to **a key generation function**, a block cipher has two functions, **an encryption function**, $E_K(.)$, and **a decryption function**, $D_K(\cdot)$, such that

$$C = E_K(M) \text{ (or } C = E(K, M))$$
$$M = D_K(C) \text{ (or } M = D(K, C))$$

where

- M is a plaintext block and C is a ciphertext block
- K is a secret (a symmetric or a private key)

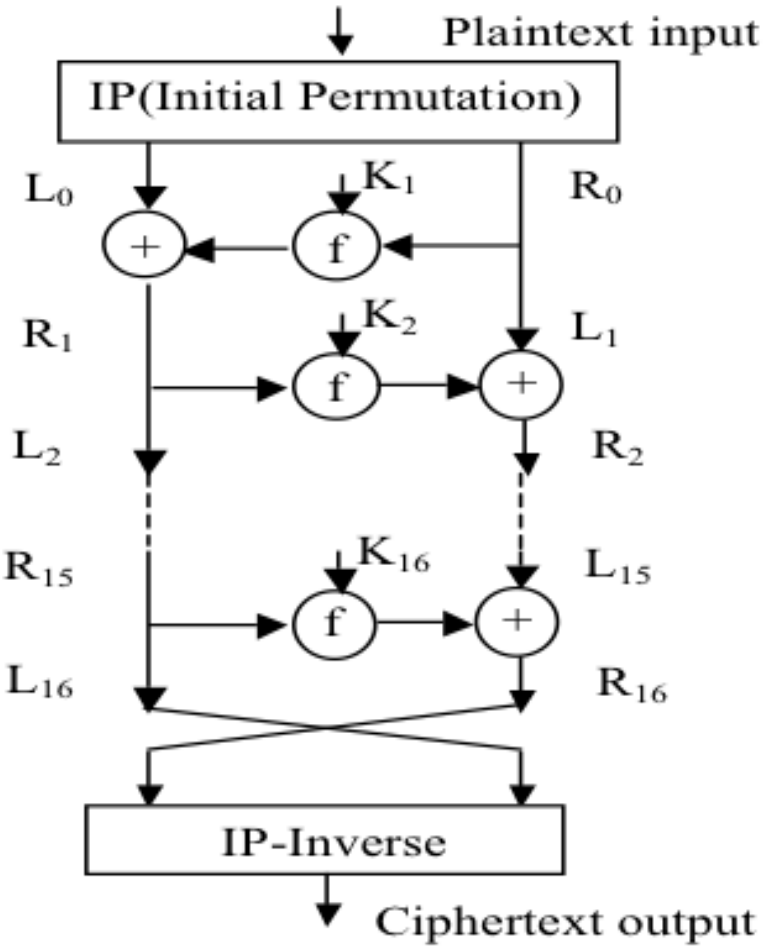
Block Cipher Design Criteria

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

Block Cipher Design

- Claude Shannon identified that confusion and diffusion are two properties of the operation of a secure cipher and these properties can be achieve by using substitution and permutation.
- Horst Feistel provided an implementation of this idea: a complex encryption function can be built out of some simple operations (round function) by repeatedly using them.
- Examples of simple operations
  - substitutions
  - permutations
  - XOR
  - modular multiplication

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** $f(R_{i-1}, K_i)$

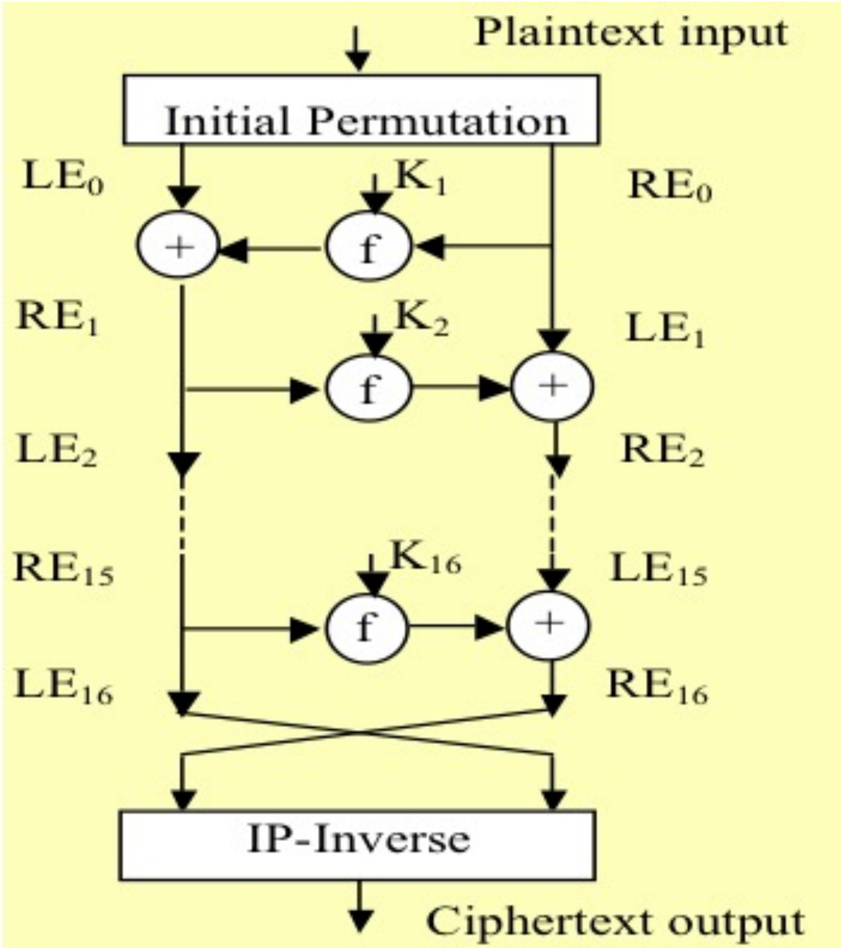
Subkeys K_i is derived from key K

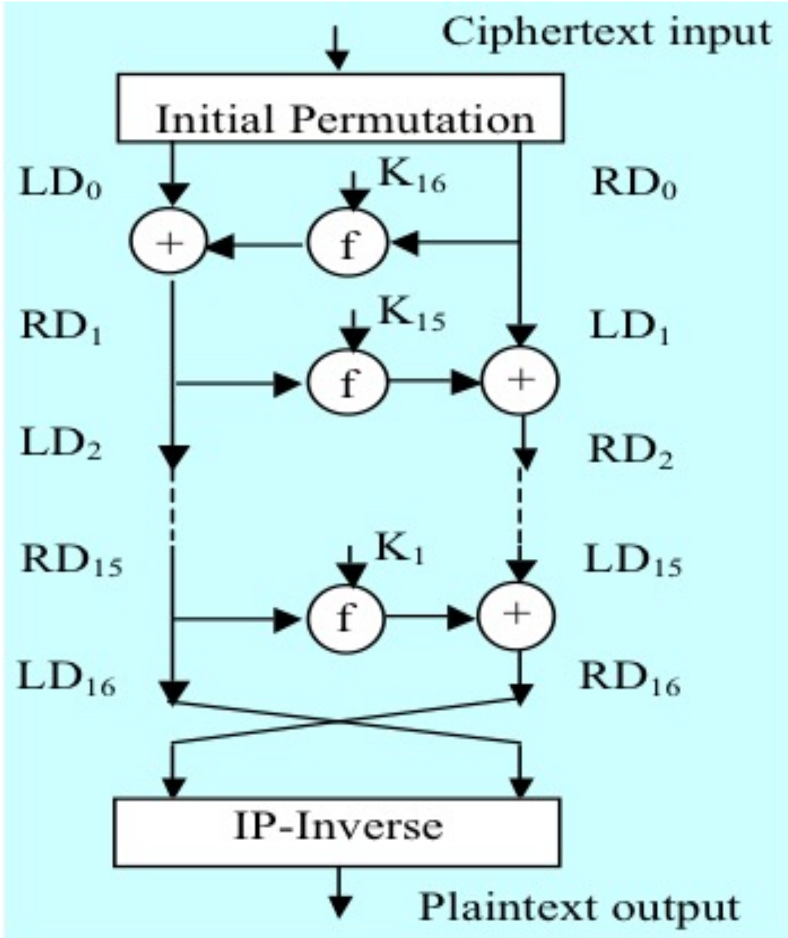
Ciphertext = (R_r, L_r)

Decryption:

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

Feistel Block Cipher





Block Cipher Design - Feistel Block Cipher

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

MANCHESTER 1824

The University of Manchester

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; actually 8 bytes, but the 8^{th} bit in each byte is a parity-check bit.
- 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 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 due to the meet-in-the-middle attack, the effective key lengths for both cases are much shorter.

AES - Background

- US NIST issued call for algorithms to replace DES in 1997.
  - stronger & faster than 3DES
  - active life of 20-30 years (+ archival use)
  - provide full specification & design details
  - both C & Java implementations
- 15 candidates accepted in 98
- 5 were shortlisted in 99
- Rijndael was selected as the AES in 2000, and formally nominated as the Advanced Encryption Standard (AES) in 2001.
- Designers
  - Vincent Rijmen, Joan Daemen → Rijndael.
- Website: http://www.nist.gov/aes/

MANCHESTER
1824

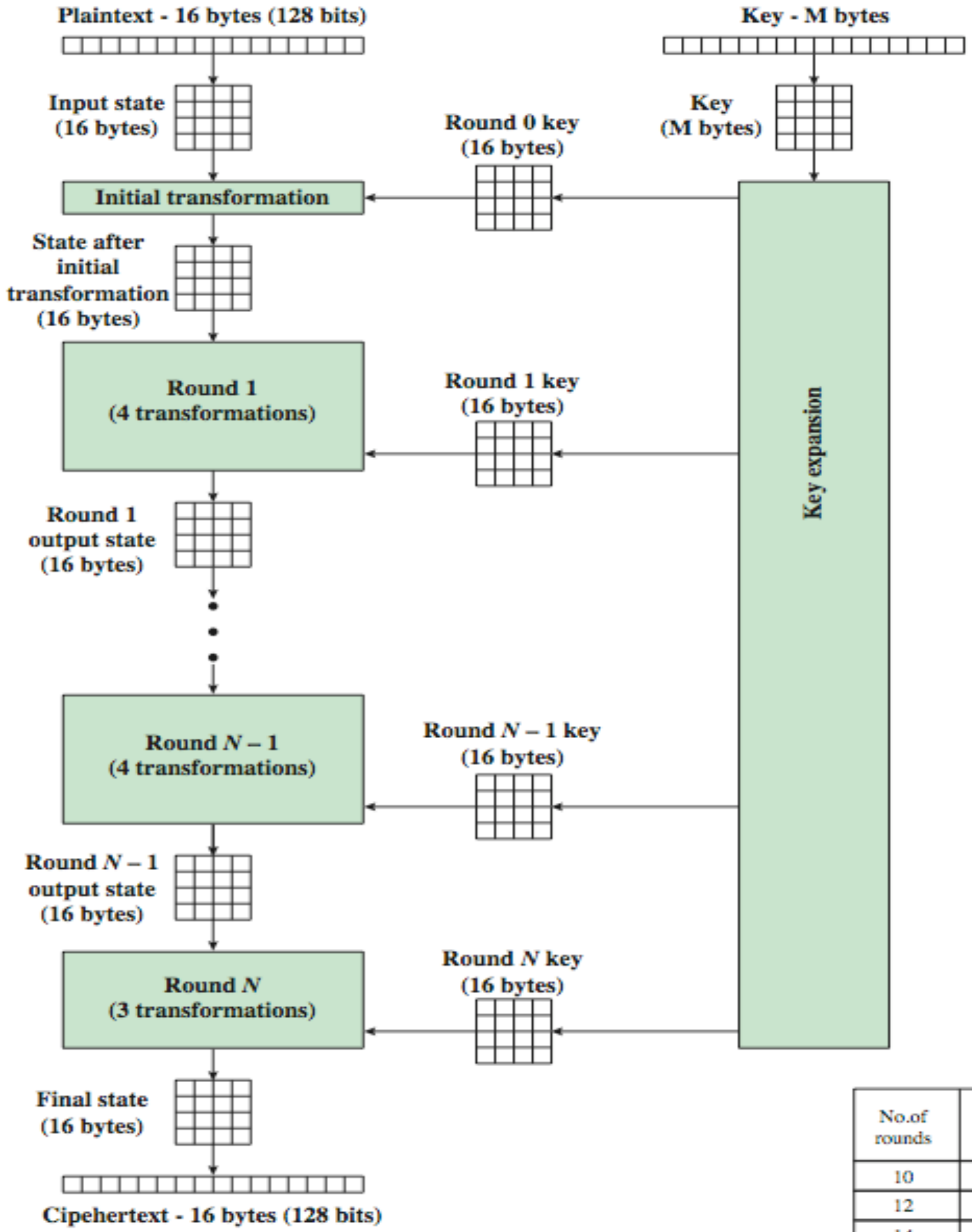
The University of Manchester

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.
- Different from DES, it is an **iterative** rather than **feistel** cipher.
- 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 – Encryption Process

- Operate on bytes – efficient in software implementation



No.of rounds	Key Length (bytes)
10	16
12	24
14	32

AES – State

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

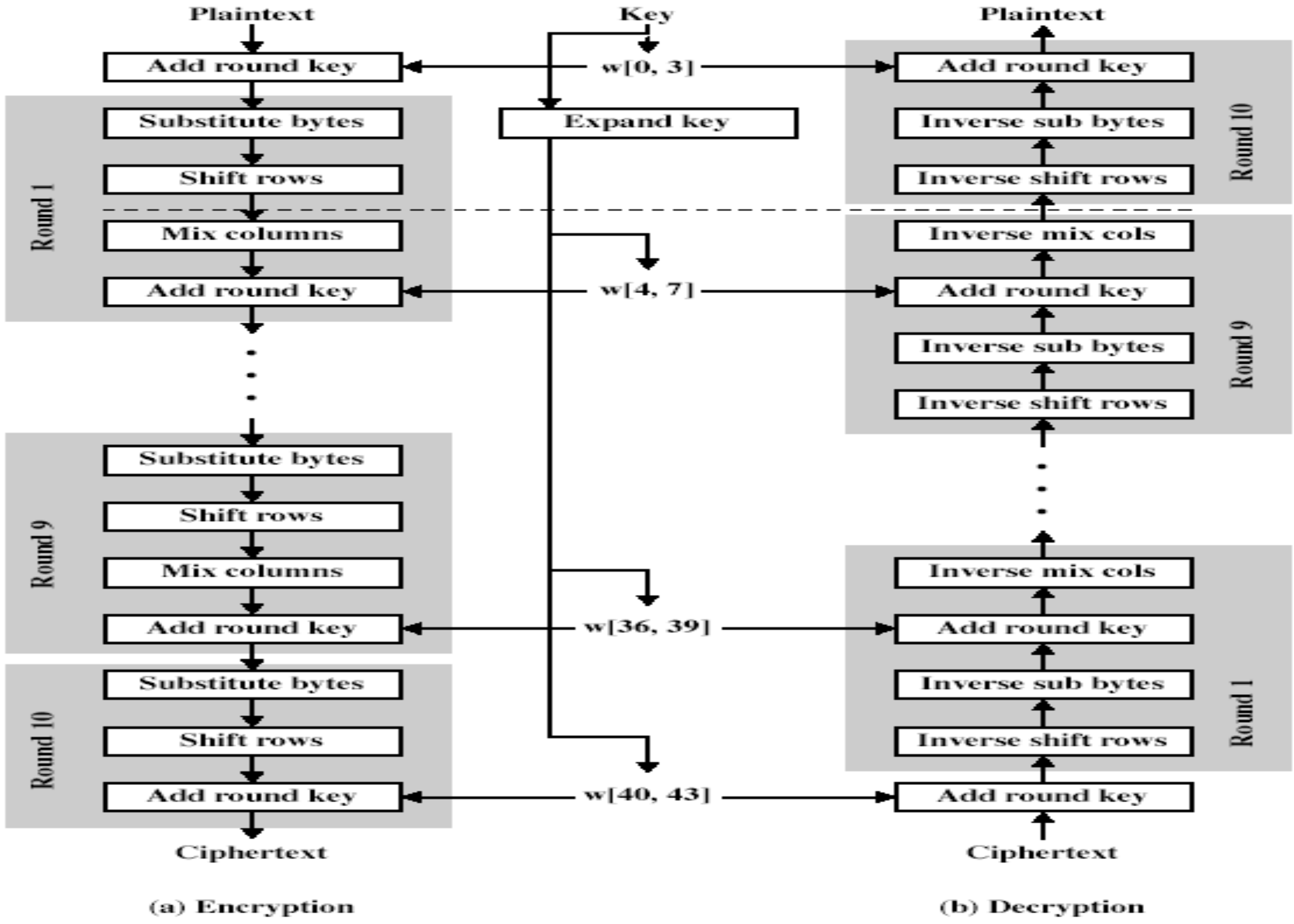
ABCDEFGHIJKLMNOP

A	E	I	M
B	F	J	N
C	G	K	O
D	H	L	P

$\xrightarrow{\text{ASCII}}$

41	45	49	4D
42	46	4A	4E
43	47	4B	4F
44	48	4C	50

AES


Plaintext
Key
Plaintext
Add round key
w[0, 3]
Add round key
Substitute bytes
Expand key
Inverse sub bytes
Shift rows
Inverse shift rows
Round 1
Round 10
Mix columns
Inverse mix cols
Add round key
w[4, 7]
Add round key
Inverse sub bytes
Inverse shift rows
Round 9
Substitute bytes
Shift rows
Round 9
Mix columns
Inverse mix cols
Add round key
w[36, 39]
Add round key
Substitute bytes
Inverse sub bytes
Shift rows
Round 10
Inverse shift rows
Round 1
Add round key
w[40, 43]
Add round key
Ciphertext
Ciphertext
(a) Encryption
(b) Decryption

AES – Structure

- Round transformation consists of:
  - Substitute bytes (SubBytes).
  - Shift rows (ShiftRows).
  - Mix columns (MixColumns).
  - Add round key (AddRoundKey).

- Sequential and light-weight key schedule.

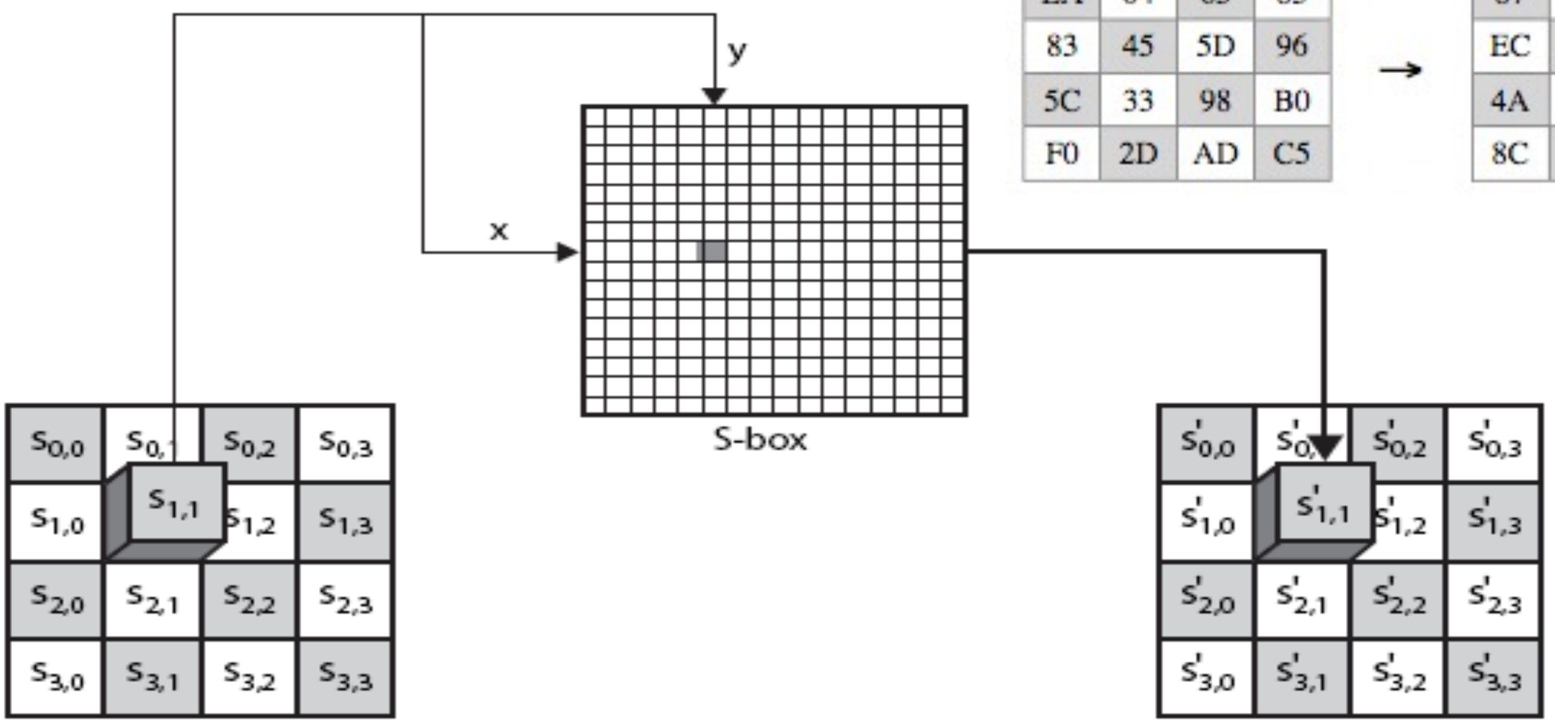
AES – Substitute Bytes

- The SubBytes transformation is via a simple table/S-box lookup.
- 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 byte is replaced by a new byte indexed by row (left 4 bits) and column (right 4 bits) of the S-box.

EA	04	65	85
83	45	5D	96
5C	33	98	B0
F0	2D	AD	C5

→

87	F2	4D	97
EC	6E	4C	90
4A	C3	46	E7
8C	D8	95	A6



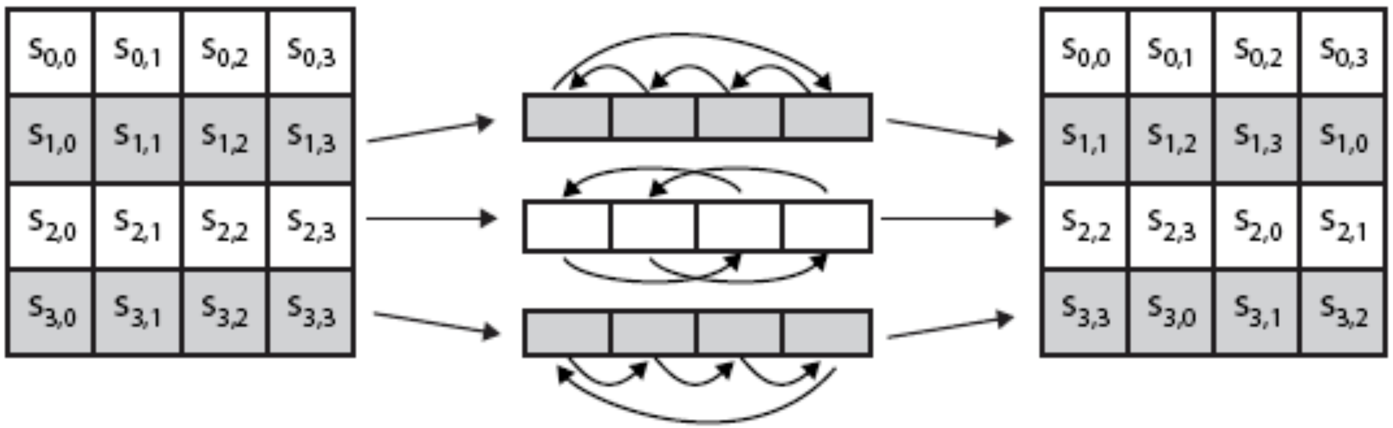
AES – Shift Rows

- The ShiftRows transformation is a simple permutation (circular byte shift):
  - 1st row: no change;
  - 2nd row: 1-byte circular left shift;
  - 3rd row: 2-byte circular left shift;
  - 4th row: 3-byte circular left shift.
- Decryption uses circular right shift.
- This step permutes bytes between the columns.

87	F2	4D	97
EC	6E	4C	90
4A	C3	46	E7
8C	D8	95	A6

→

87	F2	4D	97
6E	4C	90	EC
46	E7	4A	C3
A6	8C	D8	95



AES – Mix Columns

- A few notes about modular polynomial arithmetic, Galois Field, GF(P^n) (in AES, p=2, n=8)
- A bit-string (a_{n-1}, a_{n-2} , …, a_1, a_0) is expressed in the form of a polynomial, i.e.

$$f(x)=a_{n-1}x^{n-1}+ a_{n-2}x^{n-2} + \ldots + a_1x+a_0$$

- Arithmetic follows the ordinary rules of polynomial arithmetic using the basic rules of algebra, with the following two refinements:
  - Arithmetic on the coefficients is performed modulo p
    - when p=2, addition and subtraction are done by bitwise XOR.

AES – Mix Columns

- If a multiplication result is a polynomial of degree greater than $(n-1)$, then the polynomial is reduced by modulo some irreducible polynomial $m(x)$ of degree n, i.e., divide it by $m(x)$ and keep the remainder.
  - In AES, $m(x)=x^8+x^4+x^3+x+1$, i.e. 100011011 (or 11B). If the result is more than 8 bits, the extra bits are cancelled out by XORing the result with the 9-bit string (100011011).

- **mixColumn, along with shiftRows, provides diffusion.**

AES – Mix Columns

- Each byte of a column is mapped into a new value that is a function of all four bytes in the column; effectively a matrix multiplication in GF(2^8) using irreducible polynomial $m(x) = x^8+x^4+x^3+x+1$ (or {11B})

87	F2	4D	97
6E	4C	90	EC
46	E7	4A	C3
A6	8C	D8	95

→

47	40	A3	4C
37	D4	70	9F
94	E4	3A	42
ED	A5	A6	BC

$$(\{02\} \bullet \{87\}) \oplus (\{03\} \bullet \{6E\}) \oplus \{46\} \oplus \{A6\} = \{47\}$$

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

AES – Mix Columns

- Arithmetic in the finite field GF(2^8) with irreducible polynomial $m(x) = (x^8+x^4+x^3+x+1)$ → (1 0001 1011) or {11B}

For example:

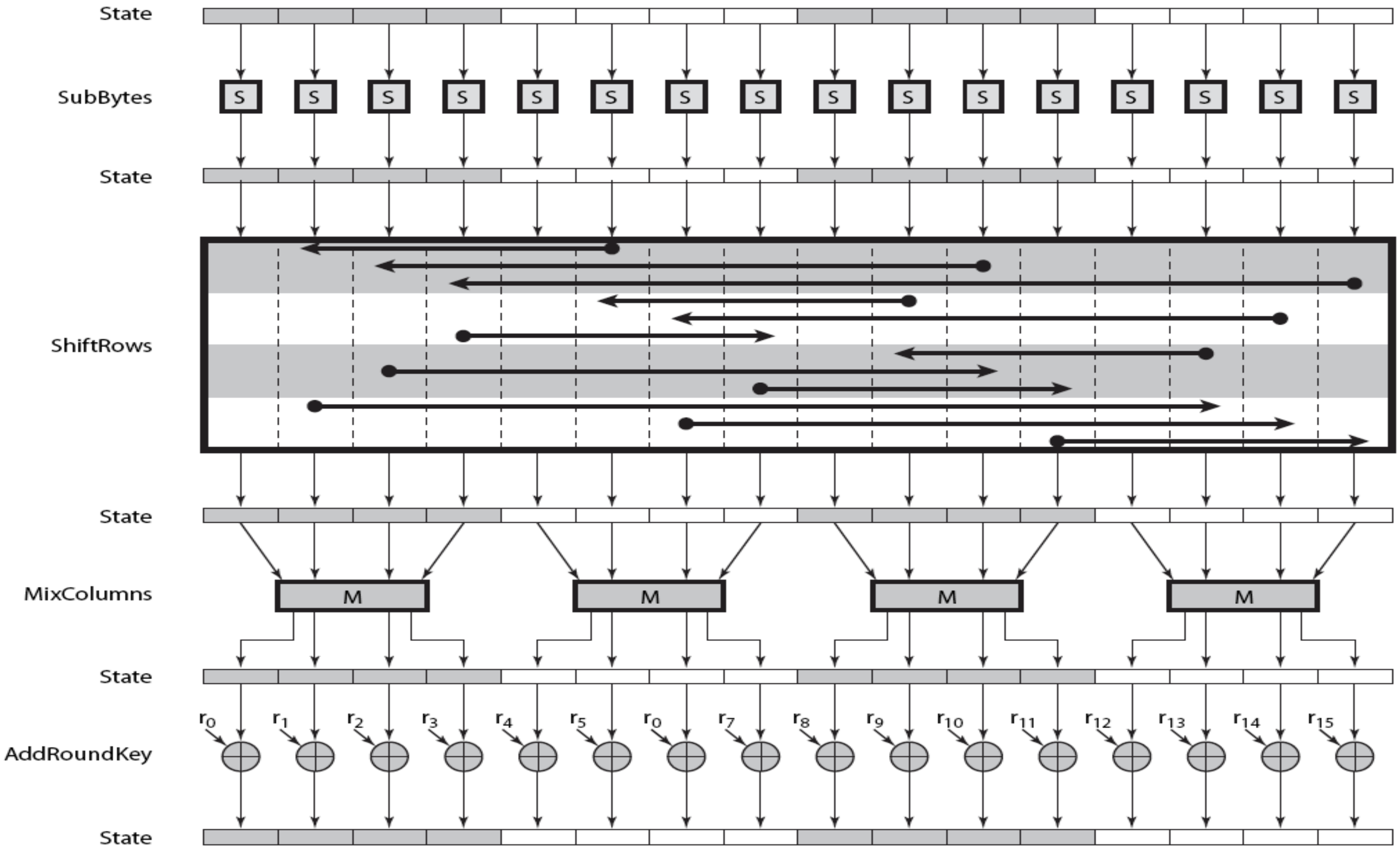
- {02} • {87} mod {11B}= (0000 0010)(1000 0111) = $x\,(x^7+x^2+x+1)$ mod $m(x)$
  $= (x^8+x^3+x^2+x)$ mod $(x^8+x^4+x^3+x+1)$
  $= x^4+x^2+1$ = {0001 0101}
- {03} •{6E} = {11}{0110 1110} = $(x+1)\,(x^6+x^5+x^3+x^2+x)$ mod $m(x)$
  $= (x^7+x^6+x^4+x^3+x^2+x^6+x^5+x^3+x^2+x)$ mod $(x^8+x^4+x^3+x+1)$
  $= x^7+x^5+x^4+x$ ={1011 0010}

- 0001 0101$\oplus$1011 0010$\oplus$0100 0110$\oplus$1010 0110=0100 0111=47

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



The University of Manchester

MANCHESTER 1824

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)

AddRoundKey(S,K[10]);

for (i = 9; i >= 1; i--); **apply InvMixColumns to the round key**

```
{
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, 128/192/256-bit key sizes.
  - 1 S-box.
  - design optimised for byte-orientated implementations.
  - open design and evaluation process.

Other Symmetrical Ciphers

Ciphers/Algos	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

The University of Manchester

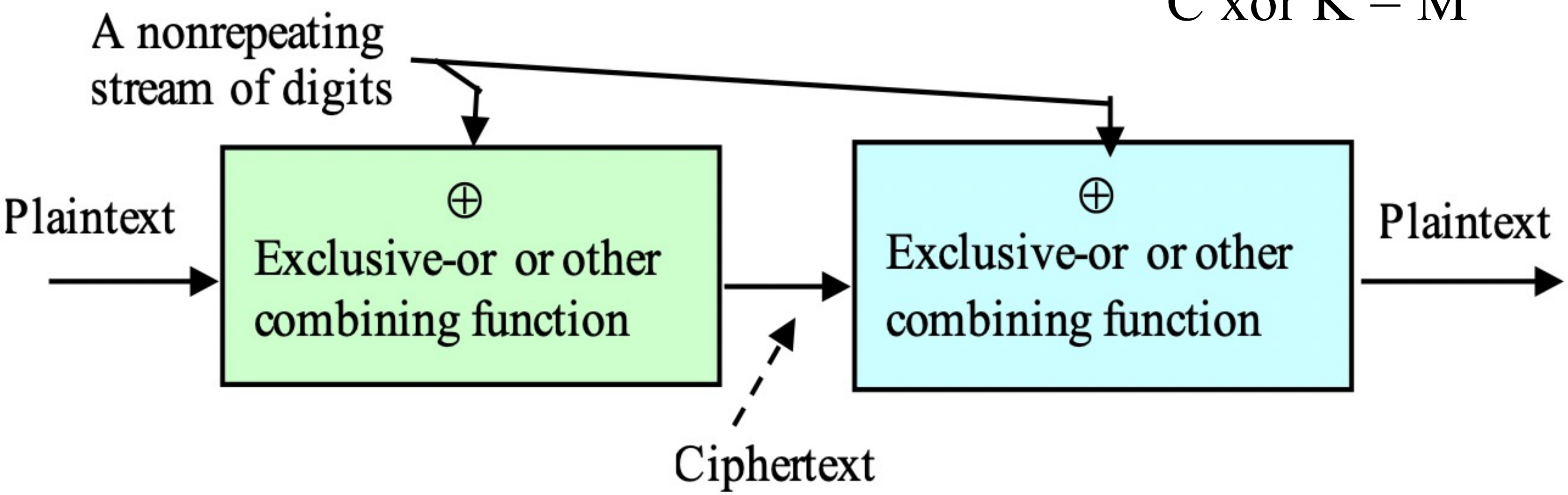
MANCHESTER 1824

Block Cipher Security

- N-bit block cipher permutes over 2^N inputs, so the larger the N value, the longer it takes to attack, thus the more secure the cipher is, but this also means that the slower the encryption/decryption operations.
- Only provide computational secrecy
  - Not impossible to break, just very expensive.
  - If the cipher is secure, then can only break by brute-force attacks, i.e. try every possible keys.
  - Time and cost of breaking the cipher exceed the value and/or useful lifetime of protected data.

Stream Ciphers: One-time Pad

M xor K = C;
C xor K = M



- One-time Pad encrypts bit streams using xor, i.e. ciphertext (C) = plaintext (M) xor keystream (KS)

$$M = m_1 \; m_2 \; m_3 \; \ldots \; m_i \; \ldots$$
$$KS = k_1 \; k_2 \; k_3 \; \ldots \; k_i \; \ldots$$
$$C = c_1 \; c_2 \; c_3 \; \ldots \; c_i \; \ldots$$

where $c_i = m_i$ xor k_i, and M and C are bit-streams.

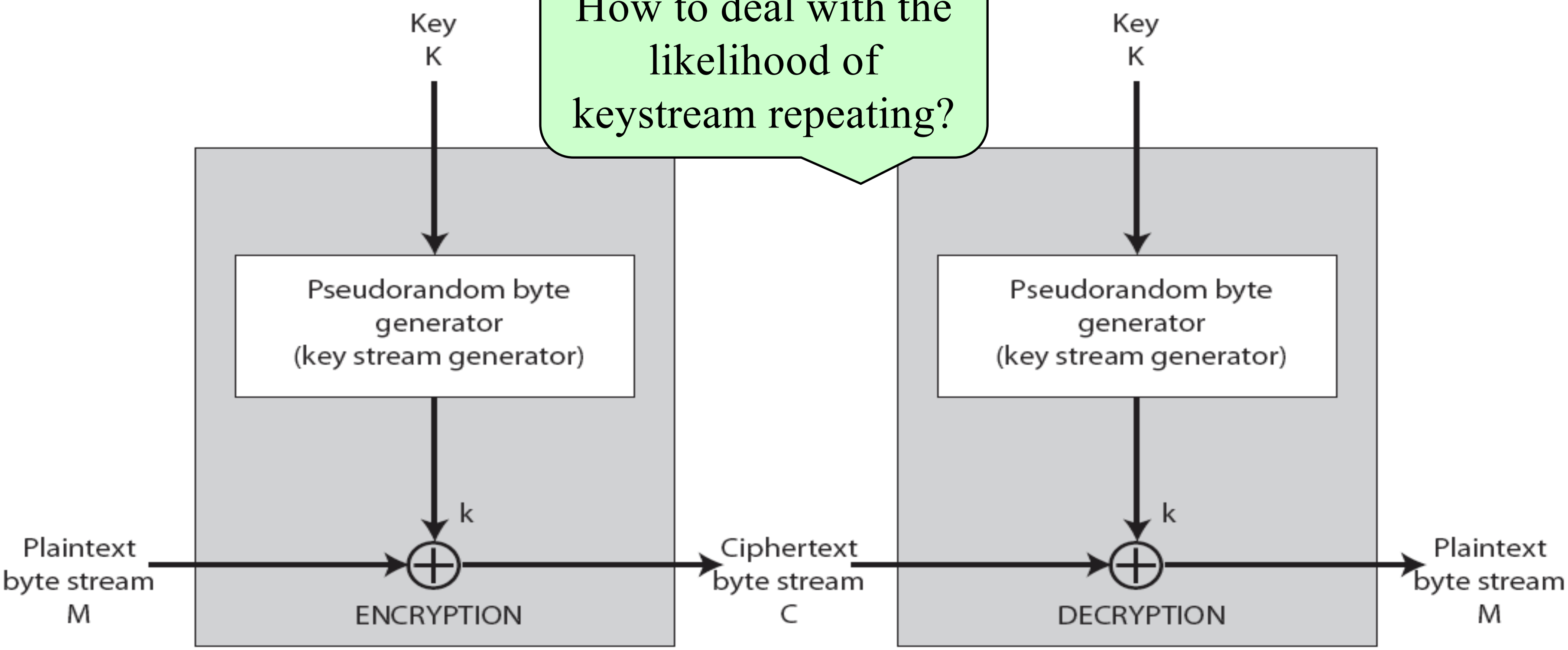
Stream Ciphers

- The cipher achieves perfect secrecy **if and only if** there are as many possible keys as possible plaintexts, and every key is equally likely (Claude Shannon, 1949).
- Benefit
  - Bitwise xor is computationally very efficient.
- Problems
  - Key must be as long as the plaintext, and this is impractical in most application scenarios.
  - Not secure if keys are reused
    - Attacker can obtain XOR of plaintexts:
      - M1 xor K = C1, M2 xor K = C2 and
      - M1 xor M2 = C1 xor C2
    - If attacker gets hold of {M1, C1}, then K =M1 xor C1

Stream Ciphers

Replace the random key in One-time Pad by a pseudo-random sequence, generated by a cryptographic pseudo-random generator that is ‘seeded’ with the **key**.

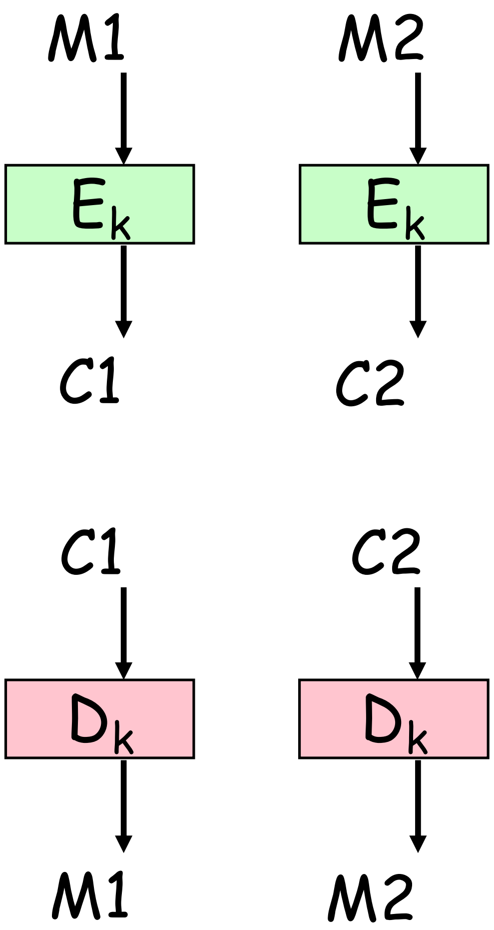
How to deal with the likelihood of keystream repeating?



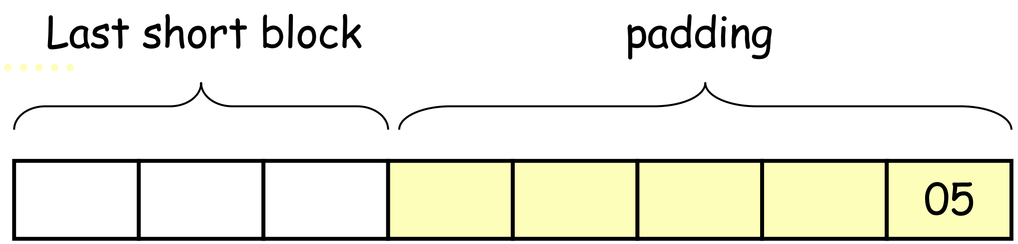
Modes of Encryption – encrypting large messages

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

Modes of Encryption - 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.

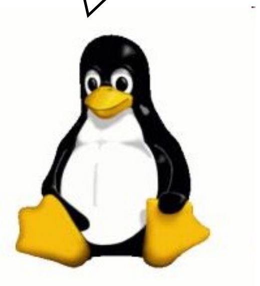


Modes of Encryption - ECB mode

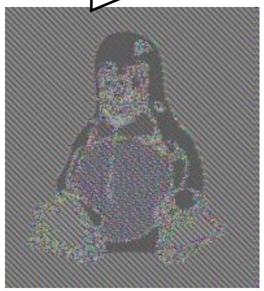
- Blocks are encrypted independently of other blocks
  - Reordering ciphertext blocks results in correspondingly reordered plaintext blocks.
- The same block of plaintext always produces the same ciphertext (with the same key)
  - patterns in plaintext show up in ciphertext.
- Error propagation: errors in one ciphertext block only affects the same plaintext block; they do not propagate to other blocks.
- Not recommended for messages longer than one block of data.

Modes of Encryption - ECB mode

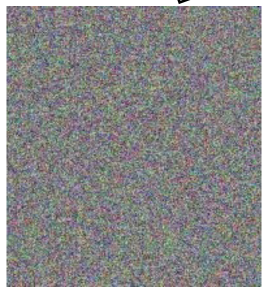
Plaintext



ECB encrypted cipher text



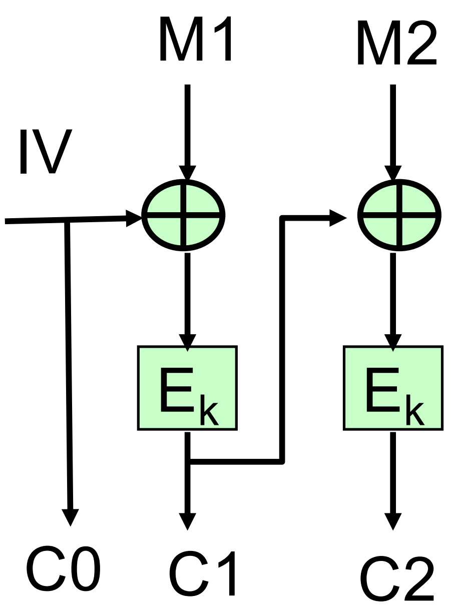
Encrypted using CBC mode



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

Modes of Encryption - CBC mode

CBC encryption

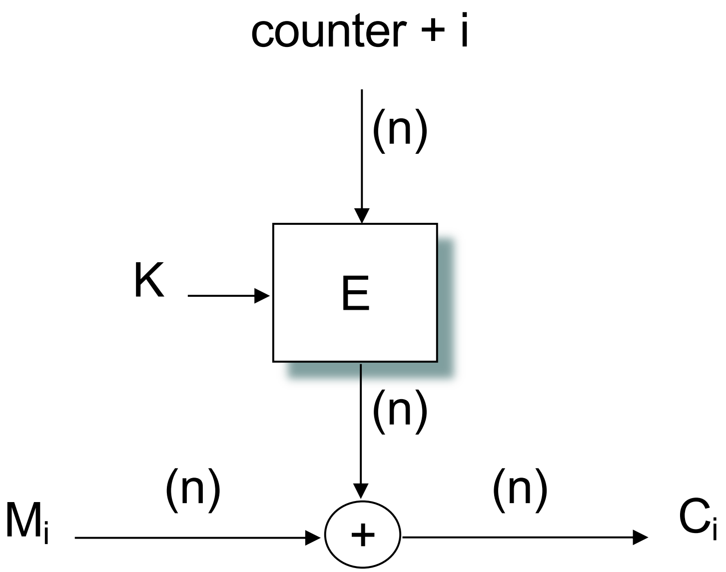


- Equation for encryption: $C_i = E_K(M_i$ **XOR** $C_{i-1})$, where C_0 = IV (Initialization Vector).
- **In this example, the plaintext is *M1M2*, and the ciphertext is *C0C1C2*.**
- Ciphertext block Cj depends on Mj and all the preceding plaintext blocks.
  - Reordering ciphertext blocks affects decryption.
  - Repeated patterns in the plaintext are concealed by the feedback.
  - There is error propagation.
- IV should be randomly selected, but does not have to be secret.
- Using different *IV*s in different encryption operations will make the same plaintext encrypted to different ciphertexts.

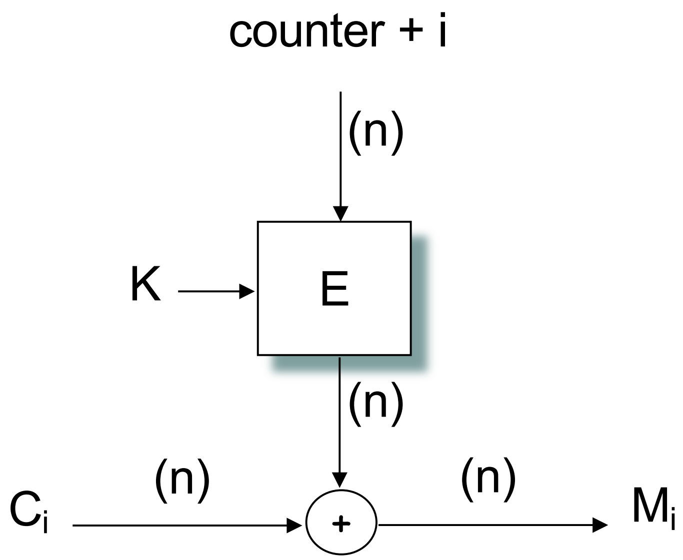
Modes of Encryption – CTR mode

- The idea is to use a block cipher encryption function as the pseudorandom number generator to generate the key stream.
- A counter value, equal to the block size, is used. The value must be different for each encryption operation.
- 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).

Encryption



Decryption



Modes of Encryption – CTR mode

- The CTR mode actually converts a block cipher into a stream cipher.
- 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.

Implementation flaws

- Notes to avoid some known vulnerabilities in software implementations
  - Should use a secure random function from a crypto library.
    - C has a built-in random function: rand(); shouldn't use rand() for security-critical applications.
  - IVs should be random
    - Have seen implementations which uses NULL as IV – this is a vulnerability.

Exercise Question – E3.1

- By applying DES twice using two different 56-bit keys, K1 and K2, to encrypt a message, M, i.e. $C=E_{K2}[E_{K1}[M]]$, where C is the ciphertext, we have a double DES encryption. Would this double DES encryption double the security level of a single DES encryption? Justify your answer.

Exercise Question – E3.2

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:

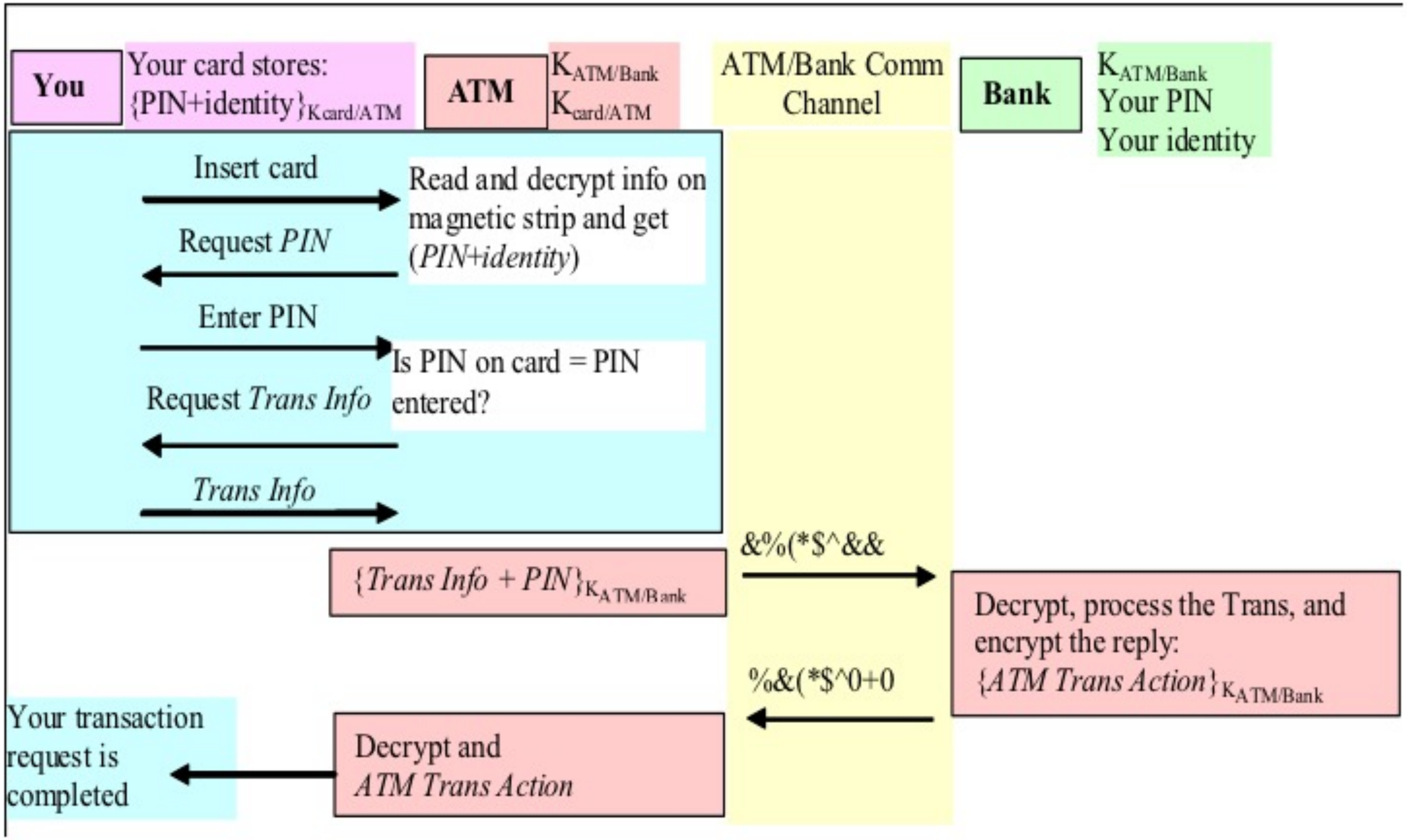
- Cash card stores the ciphertext of the user's Identity (ID) and PIN that are encrypted using a symmetric key, $K_{card/ATM}$.
- The communication between ATM and bank backend office is secured using another symmetric key, $K_{ATM/Bank}$.

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 Question – E3.2 continue



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 applications of a simple round function consisted of substitution, permutation, shift and key addition.
- To use a block cipher, one needs to specify a mode of encryption/operation:
  - the simplest mode is ECB mode, but it is not secure for long message encryptions.
  - CBC mode is the default mode in most commercial applications that encrypt more than one data block.
  - CTR modes can help you to convert a block cipher into a stream cipher.
- Symmetrical ciphers have a key exchange problem and do not support non-repudiation.