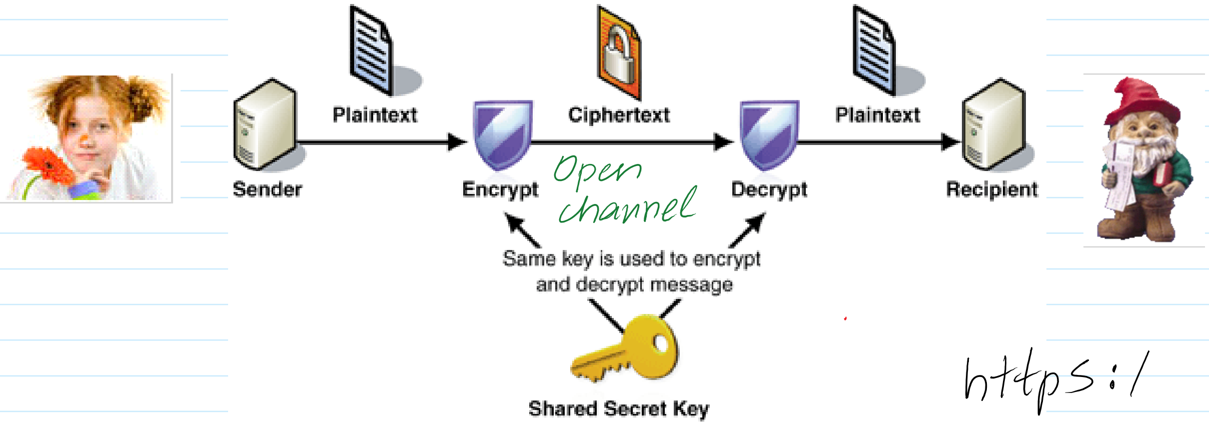
Cryptography: Information confidentiality, integrity, authenticity, person identification

Symmetric cryptography --------------------- Asymmetric cryptography

Symmetric encryption
H-functions, Message digest
HMAC H-Message Authentication Code

Asymmetric encryption
E-signature - Public Key Infrastructure - PKI
Data authenticity
Person identification
E-money, Crypto currencies
E-voting
Digital Rights Management - DRM
Etc.

Symmetric encryption

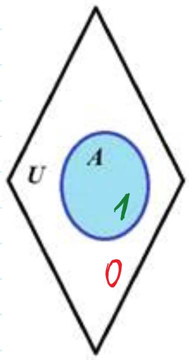


https:/ KAP

Vernam cipher (1917) - One Time Pad

Logical operations

Target

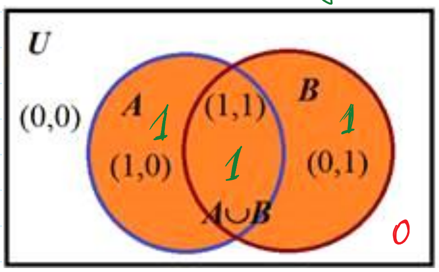


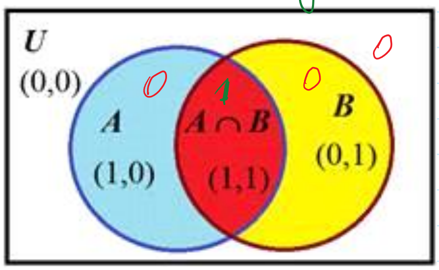
Darts: if arrow hits A, then the score is equal to 1.
Othewise is equal to 0.

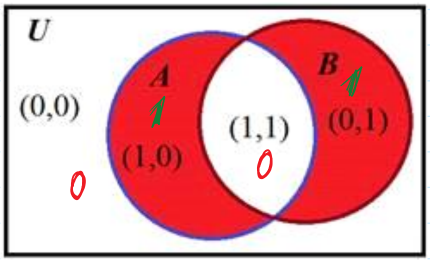
$A \cup B$ OR - Disjunction

$A \cap B$ AND-Conjunction

$A \oplus B$ XOR-Exclusive OR







„0" No

„1" Yes

$m \in \{0,1\}$

$k \leftarrow rand\{0,1\}$; $k = 1$ with $Prob = 1/2$.

A	B	A⊕B
0	0	0 ←
0	1	1
1	0	1
1	1	0 ←

$c = m \oplus k$

$Pr(k=0) = 1/2$

$Pr(k=1) = 1/2$

$\xrightarrow{c}$

if $c = 0$
if $c = 1$ } eavesdropping adversary

m	k	m⊕k=c
0	0	0 ←
0	1	1 ←
1	0	1 ←
1	1	0 ←

$c - k = m$

$\oplus$ - is inverse to itself: $k \oplus k = 0$ $\Longrightarrow$ $m \oplus \cancel{k} - \cancel{k} = m$

$k - k = 0$

$m \oplus k \oplus k =$

$= m \oplus 0 = m$

Alice: $k \leftarrow rand\{0,1\}$; Let $k = 1$;

Let $m = 1$; $k = 1$: $Pr(k=1) = 1/2$

$c = m \oplus k = 1 \oplus 1 = 0$ $\xrightarrow{c=0}$

Bob: $k = 1$.

$c \oplus k = 0 \oplus 1 = 1 = m$.

But nevertheless, the reader confusing implication and equivalence operations (functions) can accept the following proposition as valid:

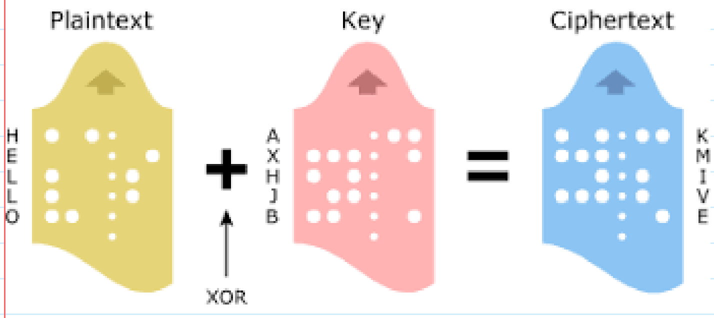
if talker has a head and donkey has a head, then talker is a donkey.
To accept this proposition as valid means that thinker confuses notions of implication and equivalence. If reader is afraid to make such a mistake, we recommend to read about that in any external source.

```
>> m=77000
m =  77000
>> mb=dec2bin(m)
mb = 10010110011001000
```

messag m consist of 17 bits : $|m| = 17$ bits.

Block ciphers

Vernam cipher (1917) - One Time Pad



$p = 77000$

$pb = dec\ bin(p)$

$k = randi(100000)$

$kb = dec2bin(k)$

$cb = Binaryxor(pb, kb)$

A:

Bit strings: ***p***, ***k***

$c = p \oplus k$ $\xrightarrow{c}$ B: k $p = c \oplus k$

Bitwice XOR operation:

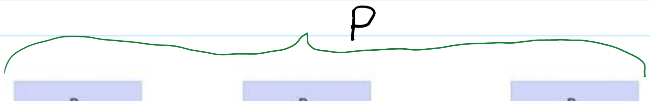
$$\begin{array}{rr} pb = & 1010 \\ \oplus \quad kb = & 0110 \\ \hline cb = & 1100 \end{array}$$

Properties: it is a perfectly secure cipher if :

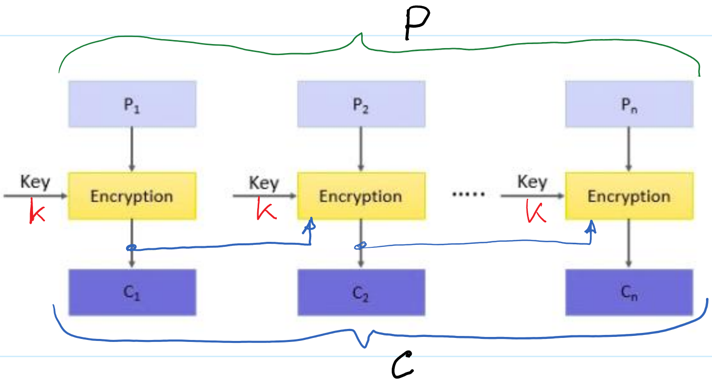
1. $|p| \leq |k|$
2. Key ***k*** must be used once.
3. The bits of ***k*** must be uniformly distributed.

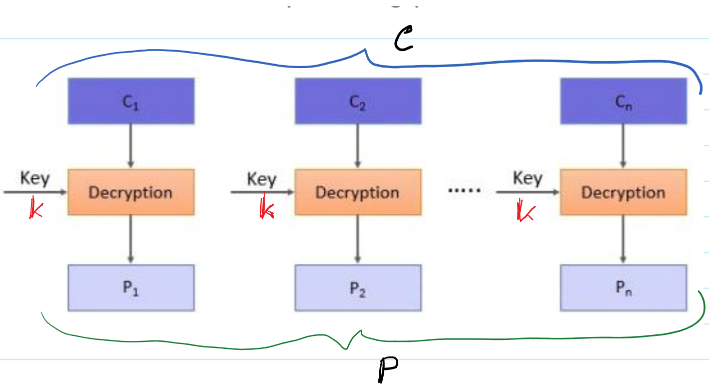
AES: Advanced Encryption Standard (2000)
Electronic Codebook Mode (ECB) mode

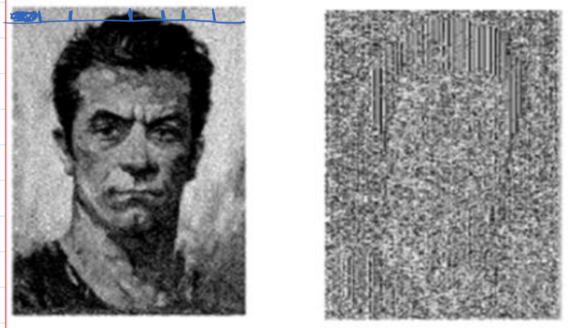
The plain text is divided into the blocks, each of N-bit. Each block is encrypted one at a time to produce the cipher block.



The ciphertext is again divided into blocks, each of N-bit and each block is decrypted independently one at a time with the same key to obtain the corresponding plain text block.

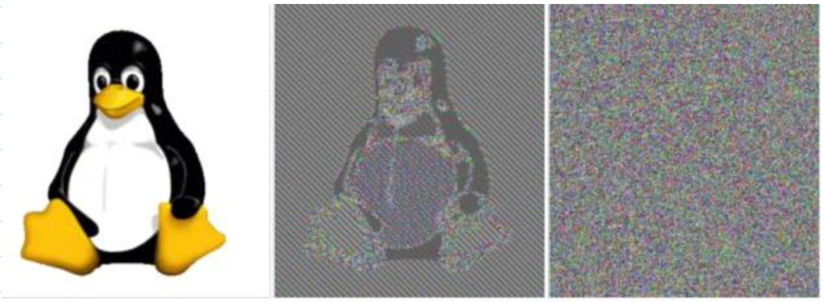






(a) plaintext

(b) plaintext encrypted in ECB mode using AES



Original image

Encrypted using ECB mode

Modes other than ECB result in pseudo-randomness

AES - 128, 192, 256 Block cipher --> Encryption --> Decryption $k \in \{128, 192, 256\}$ bit.

$|P_i| = |k|$ if AES-128 bit $\Rightarrow$ $|P_i| = 128$ bit, where $i = 1, 2, \dots, n$.

The same key k can be used multiple times, e.g. 2^{64} times.

Advanced Encryption Standard ~ 2000

Key length $\underbrace{128}, \underbrace{192}, \underbrace{256}$ bits: $k \in \{128b, 192b, 256b\}$

Data to be encripted : message m

B1	B2	B3	----	Bn

The length of any block B_i should be $|B_i| = 128$ bits
192 bits
256 bits

$|B_i| = |k|$

$$\mathrm{EncAES}(k, B1) = C1$$
$$\mathrm{EncAES}(k, B2) = C2$$
$$- - - - - - - - - -$$
$$\mathrm{EncAES}(k, Bn) = Cn$$

$m = B1 \| B2 \| \dots \| Bn$

$c = C1 \| C2 \| \dots \| Cn$

$$\mathrm{EncAES}(k, m) = c \xrightarrow{\quad c \quad} \mathrm{DecAES}(k, c) = m$$

$EncAES(k, m) = c \longrightarrow DecAES(k, c) =$

```
% AES128()
%in - text/ciphertext
%key - shared secret key
%Nr - number of rounds
%EnDec - letter which determines either encryption or decryption
%% 'e' for encryption 'd' for decryption
%Example:
%key =  '000102030405060708090a0b0c0d0e0f';
%P = '00112233445566778899aabbccddeeff';
%Nr = 10;
%C = AES128(in,key,Nr,'e')
%>>C = 69c4e0d86a7b0430d8cdb78070b4c55a
%AES128(C,key,Nr,'d')
%>>'00112233445566778899aabbccddeeff'
```

>> k = randi (2^{128}) ??

>> k = randi (2^{28})

```
>> C = AES128(P,key,Nr,'e')
```

P – plaintextext in hexadecimal
key – the lengts in 128 bits in hexadecimal
Nr – the number of rounds N = 10
'e' – for encryption
'd' – fa decryption

Modes of encryption: CBC – cipher block chain mode
For files encr. → OFB – output feedback mode
For hard drive encr. → CTR – counter Mode

$C_{CBC} = AES\text{-}128\text{-}CBC\,(IV, P, k, 'e')$ IV - Initiation Vector

$C_{OFB} = AES\text{-}128\text{-}OFB\,(\quad — \text{''} — \quad)$

$C_{CTR} = AES\text{-}128\text{-}CTR\,(\quad — \text{''} — \quad)$

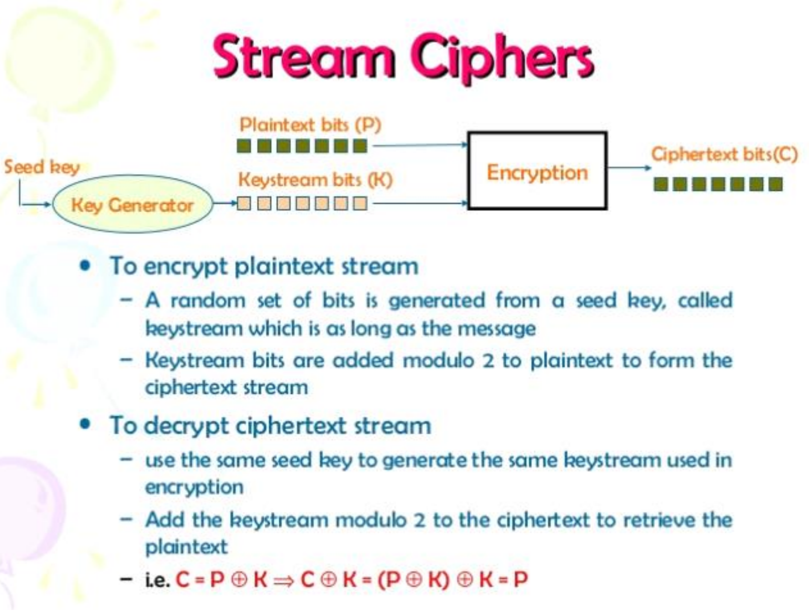
http://crypto.fmf.ktu.lt/xdownload/

- MENEZES_Handbook_Applied Cryptography M-3.pdf

till this place

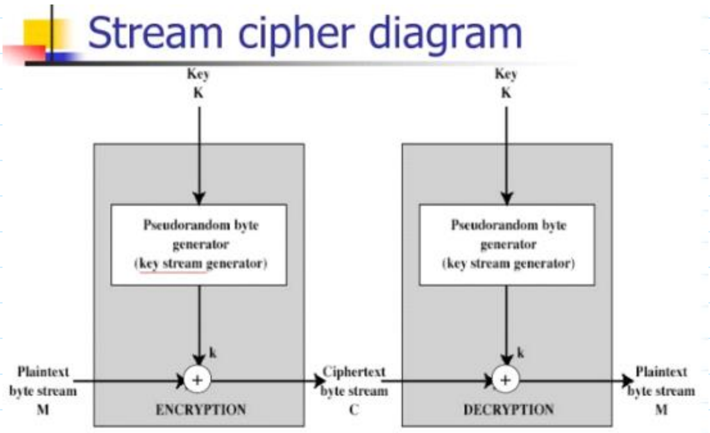
till this place

Stream ciphers



Stream Ciphers

- To encrypt plaintext stream
  - A random set of bits is generated from a seed key, called keystream which is as long as the message
  - Keystream bits are added modulo 2 to plaintext to form the ciphertext stream
- To decrypt ciphertext stream
  - use the same seed key to generate the same keystream used in encryption
  - Add the keystream modulo 2 to the ciphertext to retrieve the plaintext
  - i.e. $C = P \oplus K \Rightarrow C \oplus K = (P \oplus K) \oplus K = P$

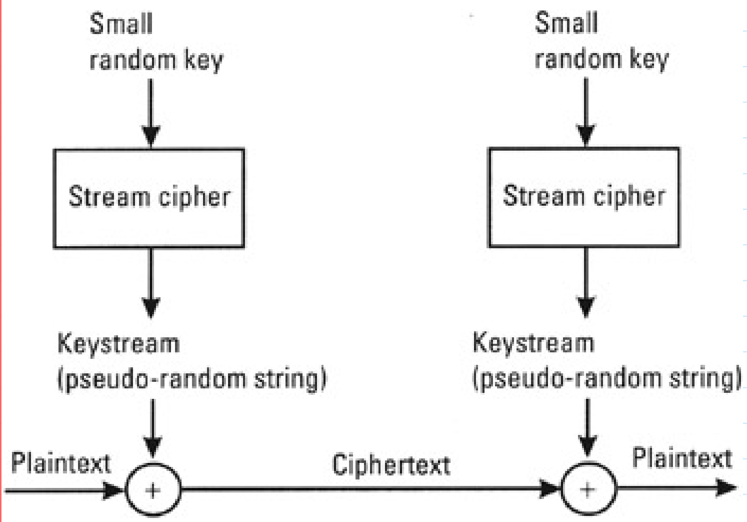


Stream cipher diagram

Key
K
Pseudorandom byte generator (key stream generator)
k
Plaintext byte stream M
ENCRYPTION
Ciphertext byte stream C
Key
K
Pseudorandom byte generator (key stream generator)
k
DECRYPTION
Plaintext byte stream M

Content Provider

Customer



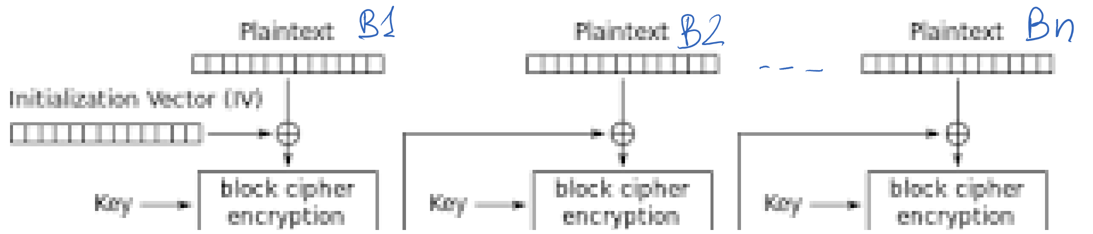
Vernam cipher

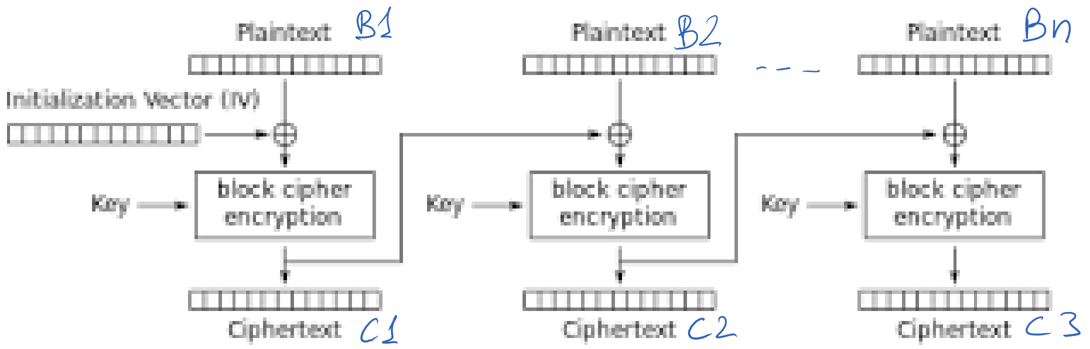
```
>> m=77000
m =  77000
>> mb=dec2bin(m)
mb = 10010110011001000
```

Symmetric encryption

- **block cipher** is one in which a block of plaintext is treated as a whole and used to produce a cipher text block of equal length.

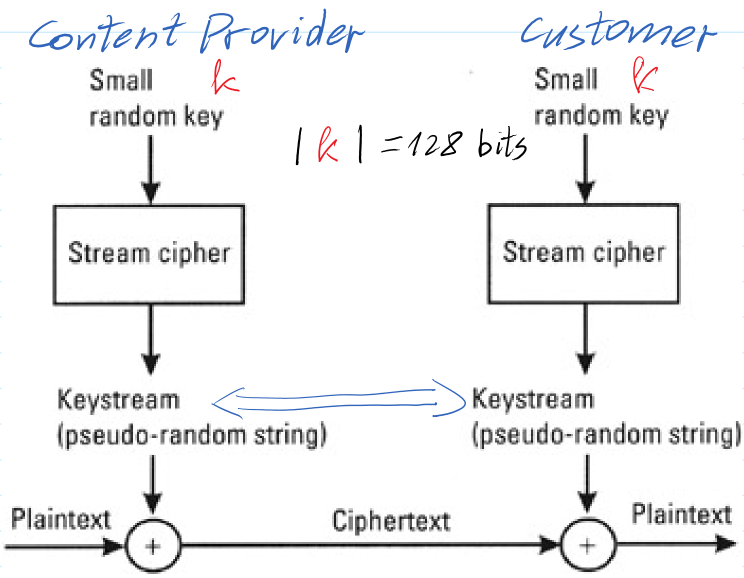
in which messag (plain text) of any finite length is divided into the number of same length block and every block is encrypted with the same relatively short key of length 128 bits, 192 bits, 256 bits or the similar length





Cipher Block Chaining (CBC) mode encryption

AES-128-CBC : $|B1| = |B2| = \ldots = |Bn| = 128$ bits



- A **stream cipher** is one that encrypts a digital data stream one bit or one byte at a time. Examples of classical stream ciphers are the auto keyed Vigenère cipher and the Vernam cipher.