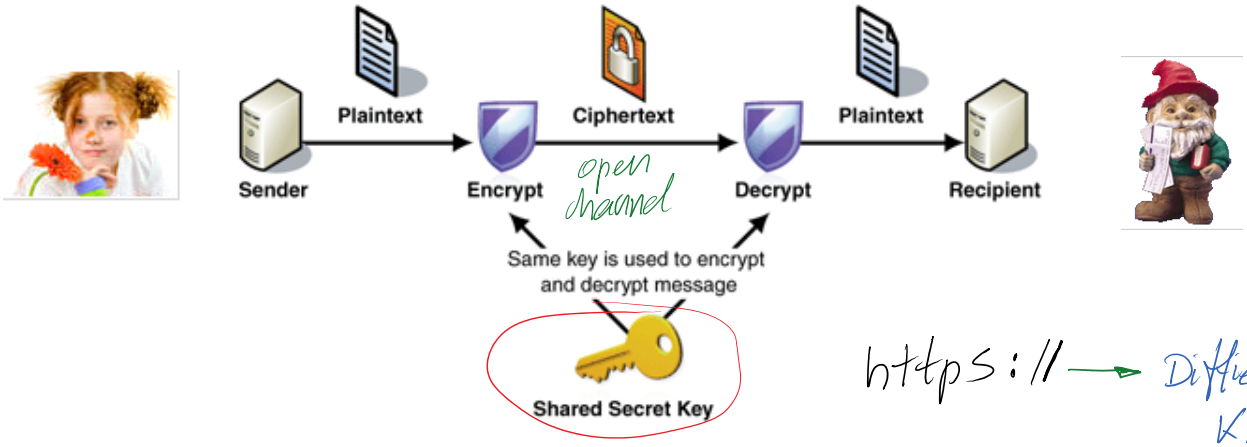
http://crypto.fmf.ktu.lt/telekonf/archyvas/inf3047%20Kript.Duom.Sauga/

Asymmetric Cryptography --> Public Key Cryptosystems - PKC

Public Parameters for **PKC** are **PP** = (**p**, **g**).
Every person has dedicated to him private key **PrK** and public key **PuK** e. g.
$\mathbf{PrK_A}$=**x** and $\mathbf{PuK_A}$=**a** for Alice.
$\mathbf{PrK_B}$=**y** and $\mathbf{PuK_B}$=**b** for Bob.
Private keys must be keeped secret but in contrary public keys are distributed among all users of PKC.

Symmetric Cryptography

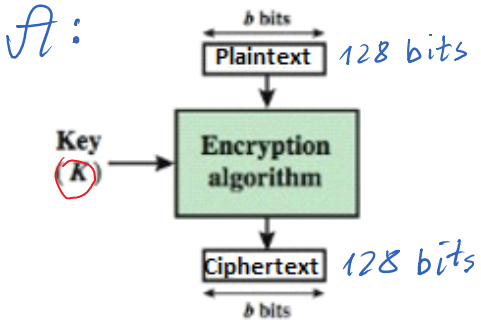
There is no any **PP** except the methods of encryption, e.g. AES-128 or H-functions.
There are only symmetric keys shared between the parties.
In contrary to **PKC** parties are sharing the same secret key **k** in Symmetric Cryptography.



https: // → Diffie-Hellman KAP

A:
M – message to be encrypted
E(k, M) = C

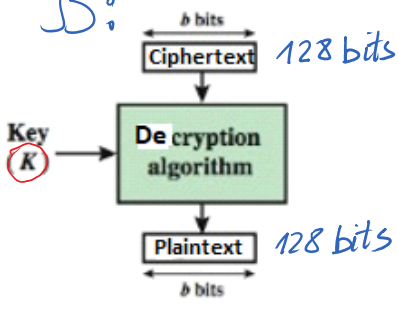
A:



- **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.

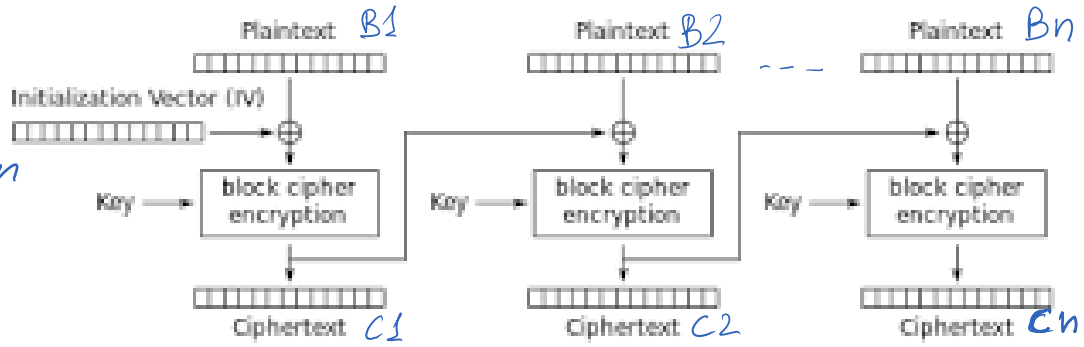
in which messag (plain text) of any finite length is divided into the number of same length block and every block is

B:



D(k, C) = M

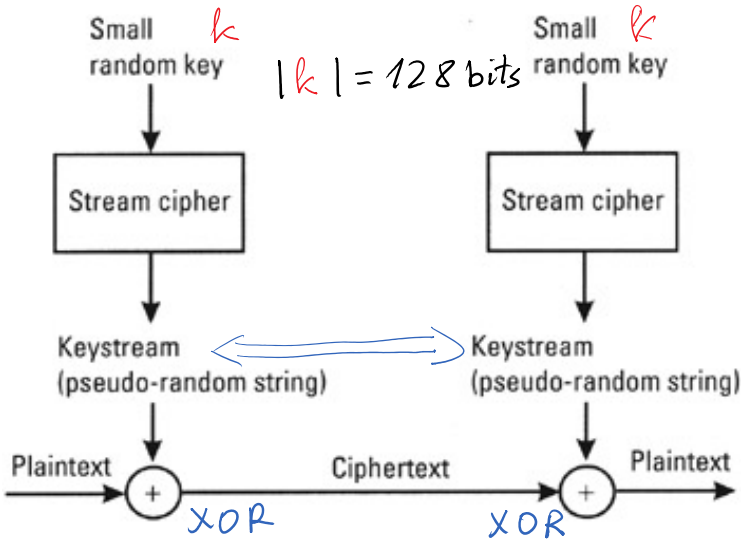
encrypted with the same
relatively short key of
length 128 bits, 192 bits, 256 bits
or the similar length



IV - is open

Cipher Block Chaining (CBC) mode encryption

AES-128-CBC : $|B1| = |B2| = \dots = |Bn| = 128$ 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.

Diffie-Hellman Key Agreement Protocol - KAP

Public Parameters = (p, g) = PP

>> p = genstrongprime(28) // |p| = 28 bits

To establish KAP Public Parameters - PP are required.

$p = 11$: defines the set of integers $\mathbb{Z}_p^* = \{1, 2, 3, \dots, p-1\}$

$\mathbb{Z}_p^* = \{1, 2, 3, \dots, 10\}$ with defined operations mod 11.

Let us fix p – as a prime, then any integer z could be expressed in the form

$$z = t \cdot p + r$$

Let $p = 11$ and $z = 37 \Rightarrow z = 3 \cdot 11 + 4$

$37 \bmod 11 = 4$

$z \bmod p = r$

$37 : 11 = 3$, remainder 4

$(t \cdot p + r) : p = t$, remainder r

$\mathbb{Z}_{11}^* = \{1, 2, 3, \dots, 10\}$ $\quad * \bmod 11$: it is a group of integers mod p.

Multiplication Tab.		Z11*									
*	1	2	3	4	5	6	7	8	9	10
1	1	2	3	4	5	6	7	8	9	10
2	2	4	6	8	10	1	3	5	7	9
3	3	6	9	1	4	7	10	2	5	8
4	4	8	1	5	9	2	6	10	3	7
5	5	10	4	9	3	8	2	7	1	6
6	6	1	7	2	8	3	9	4	10	5
7	7	3	10	6	2	9	5	1	8	4
8	8	5	2	10	7	4	1	9	6	3
9	9	7	5	3	1	10	8	6	4	2
10	10	9	8	7	6	5	4	3	2	1

$2 \cdot 6 \bmod 11 = 12 \bmod 11 = 1$

$32 : 11 = 2$, remainder 10

$2^5 \bmod 11 = 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \bmod 11 = 32 \bmod 11 = 10$

Power Tab.		Z11*									
^	0	1	2	3	4	5	6	7	8	9	10
1	1	1	1	1	1	1	1	1	1	1	1
2	1	2	4	8	5	10	9	7	3	6	1
3	1	3	9	5	4	1	3	9	5	4	1
4	1	4	5	9	3	1	4	5	9	3	1
5	1	5	3	4	9	1	5	3	4	9	1
6	1	6	3	7	9	10	5	8	4	2	1
7	1	7	5	2	3	10	4	6	9	8	1
8	1	8	9	6	4	10	3	2	5	7	1

gen. 1

gen. 2

gen. 3

gen. 4

$\Gamma = \{2, 6, 7, 8\}$

$|\mathbb{Z}_{11}^*| = 10$

7	1	7	5	2	3	10	4	6	9	8	1
8	1	8	9	6	4	10	3	2	5	7	1
9	1	9	4	3	5	1	9	4	3	5	1
10	1	10	1	10	1	10	1	10	1	10	1

gen. 3
gen. 4

$|\mathcal{I}_{11}^*| = 10$
$|\Gamma| = 4$

The probability (chance) to find a generator in $\mathcal{I}_{11}^*$ (or in $\mathcal{I}_p^*$) is approximately the following

Prob(g is a generator in $\mathcal{I}_p^*$) $\approx 4/10 = 2/5$

$g \leftarrow randi$; $g \in \{2, 3, 4 \dots\}$

$PP = (p, g)$

For the security reason $p \approx 2^{2048}$; $|p| \sim 2048$ bits.

$1K \rightarrow 2^{10} = 1024 > 10^3 = 1000$
$1M \rightarrow 2^{20} \quad ---- > 10^6$
$1G \rightarrow 2^{30} \quad ---- > 10^9$
$1T \rightarrow 2^{40} \quad ---- > 10^{12}$

$2^{2048} \sim 10^{700}$

$\mathcal{I}_p^* = \{1, 2, 3, \dots, p-1\}$; $\bullet \bmod p$

C.5.3 Finding generators.
We have to look inside Z_P^* and find a generator. How?
Even if we have a candidate, how do we test it?
The condition is that ***g*** is a generator would take $|Z_P^*|$ steps to check: $p \sim 2^{2048}$ --> $|Z_P^*| \sim 2^{2048}-1$.
In fact, finding a generator given ***p*** is in general a hard problem.

We can exploit the particular prime numbers names as **strong primes**.
If ***p*** is prime and $p=2q+1$ with ***q*** prime then ***p*** is a **strong prime**. Ex. $p = 11 = 2 \cdot 5 + 1$
Note that the order of the group Z_P^* is $p-1=2q$, i.e. $|Z_P^*|=2q$. $q = (p-1)/2$
Fact C.23. Say $p=2q+1$ is prime where ***q*** is prime, then ***g*** in Z_P^* is a generator of Z_P^* iff $g^q \neq 1 \bmod p$ and $g^2 \neq 1 \bmod p$.

Testing whether ***g*** is a generator is easy given strong prime ***p***.

Now, given $p=2q+1$, the generator can be found by randomly generation numbers $g<p$ and verifying Fact C.23. The probability to find a generator is ~0.4.

How to fing more generators when **g** one is found?

Fact C.24. If ***g*** is a generator and ***i*** is not divisible by ***q*** and **2** then g^i is a generator as well, i.e.
If ***g*** is a generator and gcd(***i***,***q***)=1 and gcd(***i***,2)=1, then g^i is a generator as well.

```
>> p=genstrongprime(28)
p =  268435019
>> q=(p-1)/2
q =  134217509
>> isprime(p)
ans = 1
>> isprime(q)
ans = 1
>> g=2
g =  2
>> mod_exp(g,q,p)
ans = 268435018
>> mod_exp(g,2,p)
ans = 4
```

```
>> 2^28-1
ans =  268435455
>> dec2bin(ans)
ans = 11111111 1111111111 1111111111
>> dec2hex(268435455)
ans = FFFFFFF
ans = 1111 1111 1111 1111 1111 1111 1111
```

p = 264043379; Check that *p* is strong prime.	**p = 268435019;**
g = 2; Check that *g* is a generator.	**g = 2;**

Public Parameters - **PP** = (**p**=268435019, **g**=2) for Key Agreement Protocol - KAP

secret random number

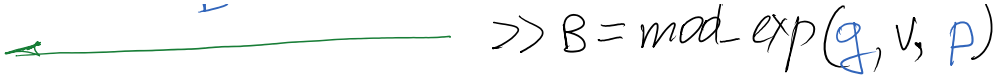
```
>> u = randi(p-1)
   A = g^u mod p
>> A = mod_exp(g,u,p)
```

A →

secret random number

```
>> v = randi(p-1)
   B = g^v mod p
>> B = mod_exp(g,v,p)
```

← B



$$k_{AB} = B^u \bmod p \quad = \quad k \quad = \quad k_{BA} = A^v \bmod p$$

$$k_{AB} = B^u \bmod p = (g^v)^u \bmod p = g^{vu} \bmod p =$$

$$= g^{uv} \bmod p = (g^u)^v \bmod p = A^v \bmod p = k_{BA}$$

$$k_{AB} = k_{BA} = k$$