

Mid Term Exam (MTE) will be held on **31-st of March, at 13:30 contactly in 516 a.**, contactly. Please participate with your own computers with installed Octave and my .m files.

During the MTE you must solve 2 problems:

1. Diffie-Hellman Key Agreement Protocol - DH KAP.
2. Man-in-the-Middle Attack (MiMA) for Diffie-Hellman Key Agreement Protocol - DH KAP.

The problems are presented in the site:

imimsociety.net

In section 'Cryptography':

[Cryptography \(imimsociety.net\)](http://imimsociety.net/Cryptography)

Please register to the site and after that you receive 10 Eur virtual money to purchase the problems.

For registration you should input the first 2 letters of your Surname and full Name, e.g. John Smith Should register as **Sm John**.

Please purchase the only one problem at a time.

If the solution is successful then you are invited to press the green button [**Get reward**].

No any other declaration about the solution results is required.

If the solution failed, then you must press the button [**Go Back**] on the **top-left** side.

Then 'Knowledge bank' will pay you the sum twice you have paid.

So, if the initial capital was 10 Eur of virtual money and you buy the problem of 2 Eur, then if the solution is correct your budget will increase up to 12 Eur.

You can solve the problems in imimsociety as many times as you wish to better prepare for MTE.

I advise you to try at first to solve the problem in 'Intellect' section to exercise the brains.

It is named as 'WOLF, GOAT AND CABBAGE TRANSFER ACROSS THE RIVER ALGORITHM'.

< <https://imimsociety.net/en/home/15-wolf-goat-and-cabbage-transfer-across-the-river-algorithm.html> >

The questions concerning the MTE you can ask at the end of the lectures.



Cryptography:
Information confidentiality, integrity,
authenticity & person identification

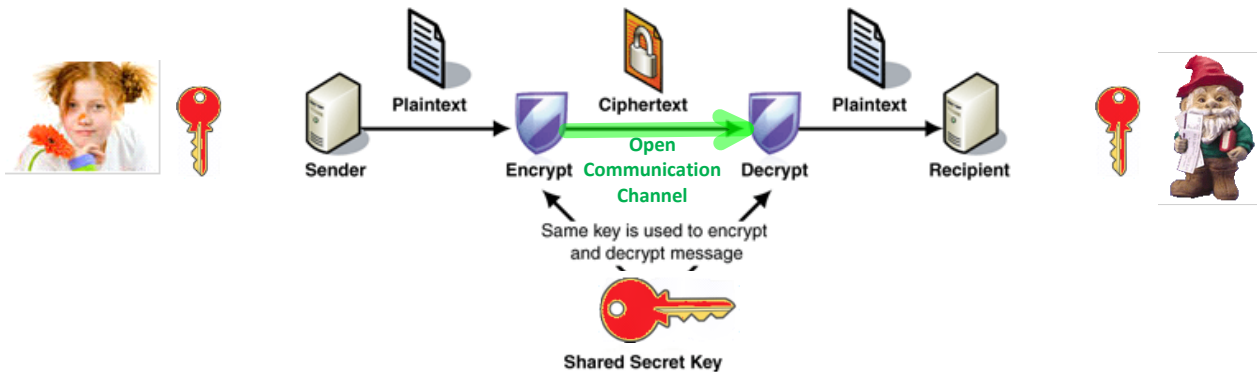
Symmetric Cryptography ----- Asymmetric Cryptography

Public Key Cryptography

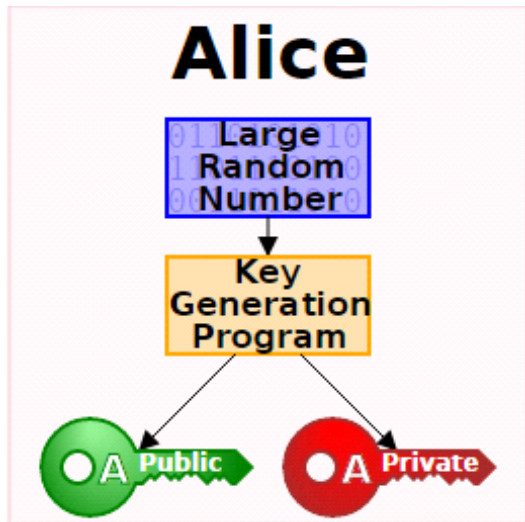
Symmetric encryption
 H-functions, Message digest
 HMAC H-Message Authentication Code

Asymmetric encryption
 E-signature - Public Key Infrastructure - PKI
 E-money, Blockchain
 E-voting
 Digital Rights Management - DRM (Marlin)
 Etc.

Symmetric - Secret Key Encryption - Decryption



Asymmetric - Public Key Cryptography



PrK and PuK are related

$$\text{PuK} = F(\text{PrK})$$

F is one-way function - OWF

Having PuK it is infeasible to find

$$\text{PrK} = F^{-1}(\text{PuK})$$

$F(x)=a$ is OWF, if:

1. It is easy to compute a , when F and x are given.
2. It is infeasible to compute x when F and a are given.

$$\text{PrK} = x \leftarrow \text{randi} \implies \text{PuK} = a = g^x \text{ mod } p$$

Public Parameters PP = (p, g)

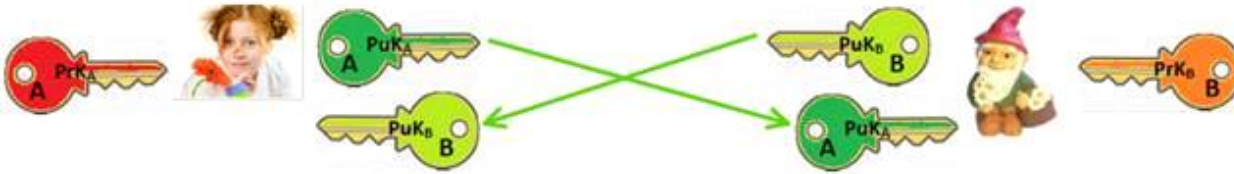
Threats of insecure PrK generation

$$\mathcal{L}_p^* = \{1, 2, 3, \dots, p-1\}; * \text{ mod } p$$

$$p \sim 2^{2048} \implies |p| \cong 2048 \text{ bits}$$

$$p \sim 2^{28} \implies |p| \cong 28 \text{ bits}$$

Public Parameters PP = (p, g)



Asymmetric Signing - Verification

$M = \text{'Hello Bob'} \rightarrow h\text{-value computation: } h = H(M)$

$\text{Sign}(\text{Pr}K_A, h) = \sigma = (r, s)$

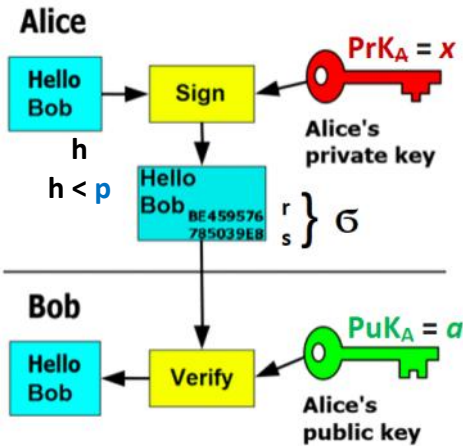
$V = \text{Ver}(\text{Pu}K_A, h, \sigma), V \in \{\text{True}, \text{False}\} \equiv \{1, 0\}$

Asymmetric Encryption - Decryption

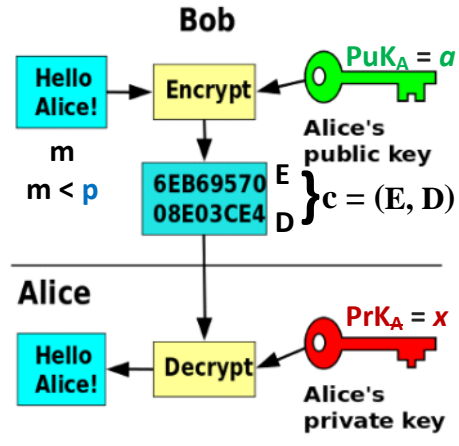
Message m must be encoded by number $m < p$.

$c = \text{Enc}(\text{Pu}K_A, m) = (E, D)$

$m = \text{Dec}(\text{Pr}K_B, c)$



Authenticity



Confidentiality

ElGamal Cryptosystem

1. Public Parameters generation $PP = (p, g)$.

Generate strong prime number p : $\gg p = \text{genstrongprime}(28)$ % strong prime of 28 bit length
 Strong prime number p : defines the set $Z_p^* = \{1, 2, 3, \dots, p-1\}$, where multiplication operations $\text{mod } p$ are defined.

Z_p^* is an algebraic group where division operations are defined as well.

Find a generator g in $Z_p^* = \{1, 2, 3, \dots, p-1\}$ using condition:

Strong prime $p = 2q + 1$, where q is prime, then g is a generator of Z_p^* iff

$g^q \neq 1 \text{ mod } p$ and $g^2 \neq 1 \text{ mod } p$.

Declare Public Parameters to the network $PP = (p, g)$;

$p = 268435019$; $g = 2$;

$\text{Pr}K = x \leftarrow \text{randi} \implies \text{Pu}K = a = g^x \text{ mod } p$

```
>> p=int64(268435019)
```

```
p = 268435019
```

```
>> dec2bin(p)
```

```
ans=1111 1111 1111 1111 1110 0100 1011
```

```
      F  F  F  F  E  4  B
```

```
>> g=2
```

```
g = 2
```

```
>> 2^28-1
```

```
ans = 2.6844e+08
```

```
>> int64(2^28-1)
```

```
ans = 268435455
```

```
>> dec2bin(ans)
```

```
ans=1111 1111 1111 1111 1111 1111 1111
```

```
>> x=int64(randi(p-1))
```

```
x = 235919127
```

```
>> a=mod_exp(g,x,p)
```

```
a = 44870925
```

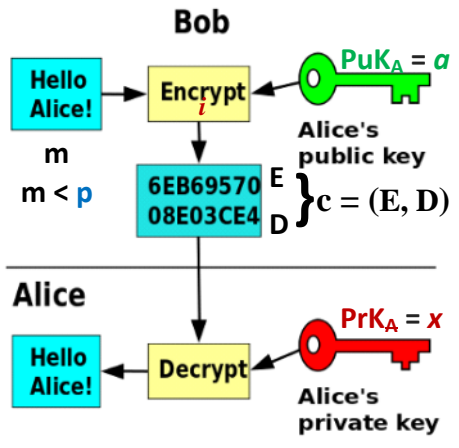
Asymmetric Encryption-Decryption: El-Gamal Encryption-Decryption

$p=268435019; g=2;$

Let message m needs to be encrypted, then it must be encoded in decimal number $m: 0 < m < p$.
 E.g., $m = 111222$. Then $m \bmod p = m$.

$27 \bmod 54 = 27$
 $27 \bmod 21 = 6 \neq 27$

A: $PuK_A = a$ \rightarrow B: is able to encrypt m to A: $m < p$



B: $i \leftarrow \text{randi}(p-1)$
 $E = m \cdot a^i \bmod p$
 $D = g^i \bmod p$
 $c = (E, D) \rightarrow A$

A: is able to decrypt $C = (E, D)$ using her $PrK_A = x$.
 1. $D^{-x} \bmod (p-1) \bmod p$
 2. $E \cdot D^{-x} \bmod p = m$

Confidentiality

$(-x) \bmod (p-1) = (0-x) \bmod (p-1) =$
 $= (p-1-x) \bmod (p-1)$

$(p-1) \bmod (p-1) = 0$ since

$(-x) \bmod (p-1) = (p-1-x)$

$D^{-x} \bmod (p-1) = D^{p-1-x} \bmod (p-1)$

$\gg D^{-x} \bmod p = \text{mod_exp}(D, p-1-x, p-1)$

$D^{-x} \bmod p = D^{p-1-x} \bmod p$

% Verification $D^x * D^{-x} = 1 \bmod p$:
 $D^x * D^{-x} \bmod p = D^{x-x \bmod p-1} \bmod p =$
 $D^0 \bmod p = 1 \bmod p$

```
>> p=int64(268435019)
p = 268435019
>> g=2
g = 2
>> x=int64(randi(p-1))
x = 144332522
```

```
% Verification  $x+(-x) = 0 \bmod (p-1)$ 
>> mx=mod(-x,p-1)
mx = 124102496
>> mod(x+mx,p-1)
ans = 0
```

```
>> i=int64(randi(p-1))
i = 17305576
>> D=mod_exp(g,i,p)
D = 43916598
>> D_x=mod_exp(D,x,p)
D_x = 251866400
```

```

>> x=int64(randi(p-1))
x = 144332522
>> a=mod_exp(g,x,p)
a = 99915647

```

```
ans = 0
```

```

>> D_x=mod_exp(D,x,p)
D_x = 251866400
>> D_mx=mod_exp(D,mx,p)
D_mx = 64836527
>> mod(D_x*D_mx,p)
ans = 1

```

Encryption:

```

>> m=111222;
>> i=int64(randi(p-1))
i = 17305576
>> a_i=mod_exp(a,i,p)
a_i = 251866400
>> E=mod(m*a_i,p)
E = 11463017
>> D=mod_exp(g,i,p)
D = 43916598

```

Decryption: m=111222

```

>> D_mx=mod_exp(D,mx,p)
D_mx = 64836527
>> mm=mod(E*D_mx,p)
mm = 111222

```

Correctness

$$\text{Enc}(PK_A = a, i, m) = c = (E, D) = (E = m \cdot a^i \bmod p; D = g^i \bmod p)$$

$$\text{Dec}(PK_A = x, c) = E \cdot D^{-x} \bmod p = m \cdot a^i \cdot (g^i)^{-x} \bmod p =$$

$$= m \cdot (g^x)^i \cdot g^{-ix} = m \cdot g^{xi} \cdot g^{-ix} = m \cdot g^{xi - ix} \bmod p = m \cdot g^0 \bmod p =$$

$$= m \cdot 1 \bmod p = m \bmod p = m = 111222$$

Since $m < p$

If $m > p \rightarrow m \bmod p \neq m$; $27 \bmod 5 = 2 \neq 27$.

If $m < p \rightarrow m \bmod p = m$; $19 \bmod 31 = 19$.

Decryption is correct if $m < p$.

ASCII: 8 bits per char.
 $\frac{2048}{8} \leq 256$ char.

Necessity of probabilistic encryption.

Encrypting the same message with textbook RSA always yields the same ciphertext, and so we actually obtain that any deterministic scheme must be insecure for multiple encryptions.

Tavern episode
 Enigma --> Alan Turing

Authenticated Key Agreement Protocol using ElGamal Encryption and Signature.

Hybrid encryption for a large files combining asymmetric and symmetric encryption method.

Hybrid encryption. Let M be a large finite length file, e.g. of gigabytes length.

Then to encrypt this file using asymmetric encryption is extremely ineffective since we must split it into millions of parts having 2048 bit length and encrypt every part separately.

The solution can be found by using **asymmetric encryption** together with **symmetric encryption**, say AES-128.

It is named as **hybrid encryption method**.

For this purpose the **Key Agreement Protocol (KAP)** using **asymmetric encryption** for the same symmetric secret key **k** agreement must be realized and encryption of **M** realized by **symmetric encryption** method, say AES-128.

AKAP: Asym.Enc & Digital Sign.

How to encrypt large data file **M**: Hybrid enc-dec method.

Lo: Parties must agree on common symmetric secret key **k**.
for symmetric block cipher, e.g. AES-128, 192, 256 bits.

A: $PrK_A = x$; $PuK_A = a$.
 $PuK_B = b$.

B: $PrK_B = y$; $PuK_B = b$.
 $PuK_A = a$.

1) $k \leftarrow rand_i(2^{128})$
 $i_k \leftarrow rand_i(2^{128})$

$Enc(PuK_B = b, i_k, k) = c = (E, D)$

2) **M** - large file to be encrypted

$E_k(M) = AES_k(M) = G$

3) Signs ciphertext **G**

3.1) $h = H(G)$

3.2) $Sign(PrK_A = x, h) = \sigma = (r, s)$

c, G
 σ, PuK_A
 $Cert_A$

- 1.1. Verify if PuK_A and $Cert_A$ are valid?
- 1.2. Verify if σ on $h = H(G)$ is valid?
 $h' = H(G)$
 $Ver(PuK_A, \sigma, h') = True$
- 2. $Dec(PrK_B, c) = k$
- 3. $D_k(G) = AES_k(G) = M$.

A was using so called encrypt-and-sign (E-&-S) paradigm.

(E-&-S) paradigm is recommended to prevent so called chosen ciphertext attacks - CCA: it is most strong attack but most complex in realization.

Till this place

$PP = (p, g)$

Lo: $z \leftarrow rand_i(p-1)$
 $v = g^z \text{ mod } p$

{ Dear **B** I am **A**
and I am sending
you my $PuK = v$ }

B: Believes that
 $PuK = v$ is of **A**

$$v = g^z \text{ mod } p$$

and I am sending you my $PuK = v$ } $\rightarrow PuK = v$ is of A

$m = \text{'Bob get out'}$

$$G = \text{Sign}(z, m) = (r, s)$$

$$m, G = (r, s)$$

B : verifies the signature G on m using $PuK = v$ and verification passes.

Before Bob verifies any signature with someone PuK he must be sure that this PuK is got from the certain person, e.g. A but not from anybody else!

It is achieved by creation of PKI - Public Key Infrastructure when Trusted Third Party (TTP) such as Certification Authority is introduced. CA is issuing PuK Certificates for any user by signing PuK when user proves his/her identity to CA.

A : Identification Card - ID

$$PrK_A = x; PuK_A = a.$$



ID

CA : PrK_{CA} ; PuK_{CA} .

PuK_A | Cert_A

Cert_A

PuK_A

$$\text{Sign}(PrK_{CA}, PuK_A || \text{Data}_A) = G_A.$$

B : $\text{Ver}(PuK_{CA}, \text{Cert}_A) = \text{True}$

$$\text{Cert}_A = \text{' } G_A, PuK_A, \text{Data}_A \text{'}$$

Is sure that PuK_A is of A

Since CA is TTP & B can download PuK_{CA} using his browser with known to everyone link

<https://CertificationAuthority.Tusted.com>

<https://certicom.com>

ElGamal encryption is probabilistic: encryption of the same message m two times yields the different cyphertexts c_1 and c_2 .

1-st encryption:

$$\begin{aligned} i_1 &\leftarrow \text{rand}_i(\mathbb{Z}_p^*) \\ E_1 &= m \cdot a^{i_1} \text{ mod } p \\ D_1 &= g^{i_1} \text{ mod } p \end{aligned} \left. \vphantom{\begin{aligned} E_1 \\ D_1 \end{aligned}} \right\} C_1 = (E_1, D_1)$$

2-nd encryption

$$\begin{aligned} i_2 &\leftarrow \text{rand}_i(\mathbb{Z}_p^*) \\ E_2 &= m \cdot a^{i_2} \text{ mod } p \\ D_2 &= g^{i_2} \text{ mod } p \end{aligned} \left. \vphantom{\begin{aligned} E_2 \\ D_2 \end{aligned}} \right\} C_2 = (E_2, D_2)$$

$$i_1 \neq i_2$$

$c_1 \neq c_2$ Enigma