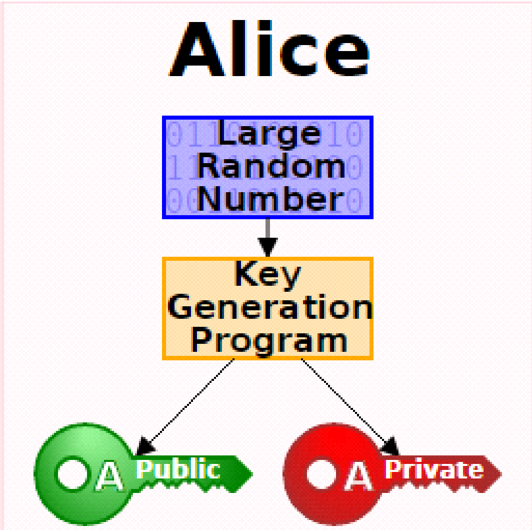
Asymmetric - Public Key Cryptography



PrK and PuK are related
PuK = F(PrK)
F is one-way function
Having PuK it is infeasible to find
PrK = F^{-1}(PuK)

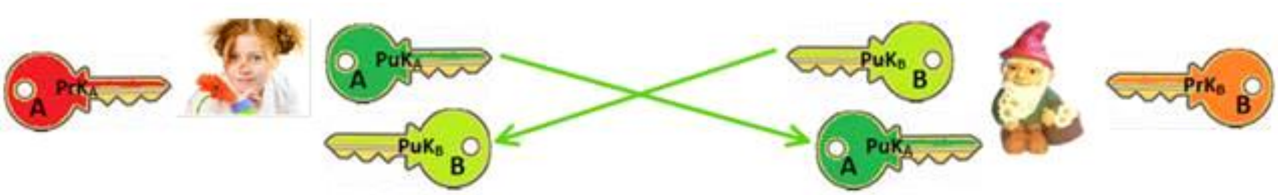
F(*x*)=*a* is OWF, if:
1.It easy to compute *a*, when *F* and *x* are given.
2.It is infeasible compute *x* when *F* and *a* are given.

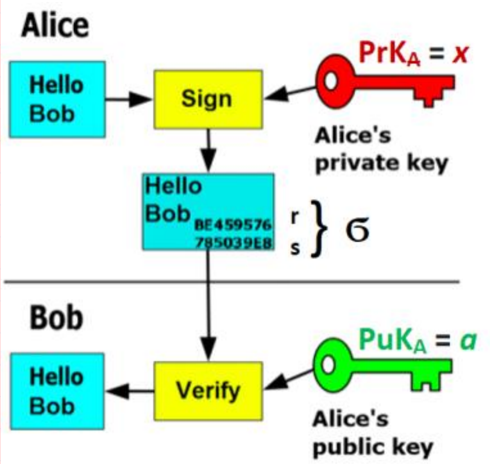
PrK = *x* <-- randi ==> PuK = *a* = g^x mod p
Public Parameters PP = (p, g)

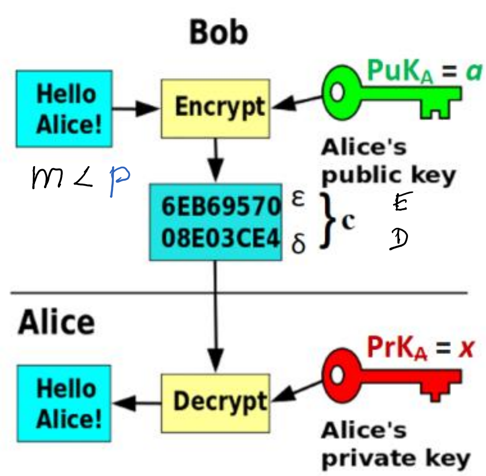
$p \sim 2^{2048}$ → $|p| \cong 2048$ bits
$p \sim 2^{28}$ → $|p| \cong 28$ bits

Threaths of insecure PrK generation

$\mathbb{Z}_p^* = \{1, 2, 3, \ldots, p-1\}$; * mod p







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***: 1< ***m*** < ***p***.
E.g. ***m*** = 111222. Then ***m* mod *p* = *m***.

$27 \bmod 54 = 27$

$27 \bmod 21 = 6 \neq 27$

A: $\xrightarrow{PuK_A = a}$ B: is able to encrypt m to A: $m < p$

Turing → ~~Enigma~~

B: $i \leftarrow randi(\mathcal{Z}_p^*)$

$\mathcal{Z}_{p-1} = \{0, 1, 2, ..., p-1\}$ $*, +, -$ mod$(p-1)$

$$\left.\begin{array}{l} E = m \cdot a^i \bmod p \\ D = g^i \bmod p \end{array}\right\} c = (E, D) \longrightarrow$$

A: is able to decrypt $c = (E, D)$ using her $PrK_A = x$.

1. $D^{-x \bmod (p-1)} \bmod p$
2. $E \cdot D^{-x} \bmod p = m$

$(-x) \bmod (p-1) = (0 - x) \bmod (p-1) =$
$= (\overline{p-1} - x) \bmod (p-1)$

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

$$\begin{array}{r} p-1 \\ -\,p-1 \\ \hline 0 \end{array} \quad \Big| \begin{array}{l} p-1 \\ \hline 1 \end{array}$$

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

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

>> D_mx = mod_exp(D, p-1-x, p)

Correctness

$Enc(PuK_A = a, i, m) = c = (E, D) = (E = m \cdot a^i \bmod p; D = g^i \bmod p)$

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

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

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

since $m < p$

If $m > p \rightarrow m \bmod p \neq m$; $27 \bmod 5 = 2 \neq 27$. ASCII: 8 bits per char.

If $m < p \rightarrow m \bmod p = m$; $19 \bmod 31 = 19$. $\frac{2048}{8} = 256$ char.

Decryption is correct if $m < p$.

Large file encryption → Hybrid encryption

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: Symmetric Enc & Asymmetric Enc & Digital Sign

$\Downarrow$

Hybrid Encryption

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

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

$A \to B$: $c, G, \sigma, PuK_A, Cert_A$

A:

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

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

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

B:

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 recomended to prevent so called choosen Ciphertext Attacks – CCA: it is most strong attack but most complex in realization.

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

1-st encryption:

$i_1 \leftarrow randi(\mathbb{Z}_p^*)$

$E_1 = m \cdot a^{i_1} \bmod p$

$D_1 = g^{i_1} \bmod p$

$\Big\} c_1 = (E_1, D_1)$

$i_1 \neq i_2$

$c_1 \neq c_2$

2-nd encryption

$i_2 \leftarrow randi(\mathbb{Z}_p^*)$

$E_2 = m \cdot a^{i_2} \bmod p$

$D_2 = g^{i_2} \bmod p$

$\Big\} c_2 = (E_2, D_2)$

Enigma

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