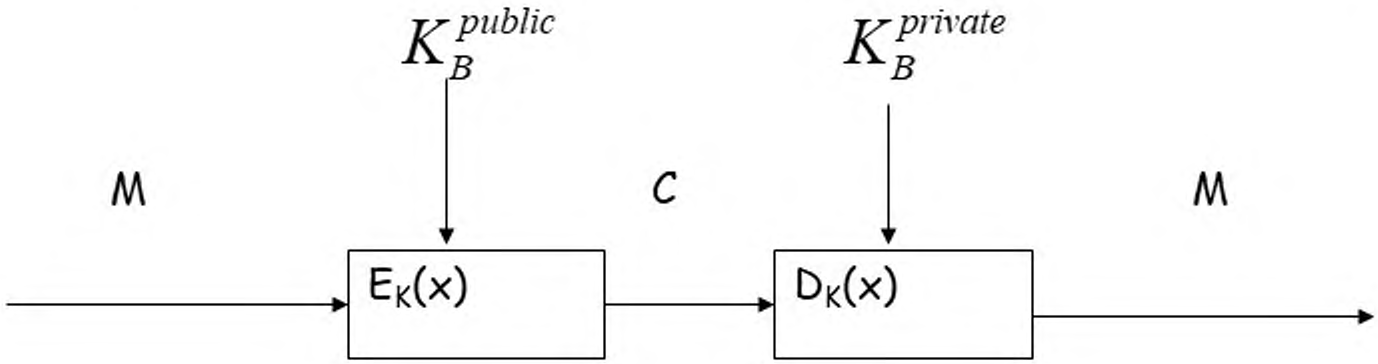
Asymmetric Cryptography



A(Armun) and B(Bolat) have own private and public key. Here message is ***M***, ciphertext is ***C***.

Public keys may be published

Some information from Numbers Theory

Definition. Integer numbers *a* and *b* are equivalent on mod n, and written

$$a \equiv b \pmod{n}$$

if their difference for some integer *k* you can write:

$$a-b=kn$$

Here n ($n \geq 1$).
Example. $25 \equiv 4 \pmod{7}$, $-5 \equiv \pmod{13}$

The class

Definition . All of the integer numbers, that are equivalent for number *a* on mod *n*, form set. This set is called class on mod n.

Example.
The class 5 on modulo 6: {...-7,-1, 5,11,17,23,...}
And there are infinite numbers in this class

Deductions

Definition. The class Deduction is any of numbers from this class

But we can use only limited set for our calculations on PC. So we will work only with smallest positive number from this set.
And we will write $b = a \bmod n$ instead $b \equiv a \pmod{n}$

Example

It is true

$25 \equiv 4 \pmod 7$

$4 \equiv 25 \pmod 7$

$4 = 25 \bmod 7$

It is wrong

$25 \equiv 4 \bmod 7$

Modular Arithmetic

In modular arithmetic we use the ordinary arithmetic rules, but from any result, we subtract that multiple n which make the result fall within the range $0,\ldots,n-1$.

Modular arithmetic is commutative, associative, distributive.

$$(a + b) \bmod n = ((a \bmod n) + (b \bmod n)) \bmod n$$
$$(a - b) \bmod n = ((a \bmod n) - (b \bmod n)) \bmod n \qquad (1)$$
$$(a * b) \bmod n = ((a \bmod n) * (b \bmod n)) \bmod n$$
$$(a*(b + c)) \bmod n = (((a * b) \bmod n) +((a * c) \bmod n)) \bmod n$$

Eiler's function

Definition. Eiler's function is the function $\varphi(n)$ that is equal the quantity of the positive integer numbers i which smaller than n and (i,n)=1. Here n is the positive integer number.

For example $\varphi(10)=4$.

There are following characters of this function:
if $n=pq$, where p and q- primes, then

$$\varphi(n) = (p-1)(q-1) \qquad (2)$$

Some theorems

Theorem 1 (Fermat's) The following is true for any prime p and any $a \geq 1$, which isn't divide on p,

$$a^{p-1} \equiv 1 (\mathrm{mod}\ p)$$

Theorem 2 (Eiler's) The following is true for any modulo n and any a , (a,n)=1

$$a^{\varphi(n)} \equiv 1 (\mathrm{mod}\ n) \qquad (3)$$

RSA

The RSA system is probably the most widely used public-key cryptosystem in the world. RSA was invented by ***Ronald Rivest , Adi Shamir*** and ***Leonard Adleman*** and first published in 1978.

It provides both digital signatures and public-key encryption.

RSA Defined

1. Start by randomly choosing two different large primes p and q. And compute $n=pq$.

2. We choose the public exponent e that $(e,((p-1)(q-1)) =$ 1. Then we calculate d, such, that $de \equiv 1(\text{mod } (p-1)(q-1))$

RSA encryption

3. **Encryption**. Message M is divided to cipher blocks that the volume is smaller than n. For binary data the length of the block is l , where l is the largest number such that $2^l < n$. To encrypt a message M, the sender computes the ciphertext as following

$$c_i = m_i^e \bmod n$$

RSA descryption

4 To descrypt the ciphertext the receiver must calculate

$$m_i = c_i^d \bmod n$$

Really, using (1), (2) and (3) we get

$$c_i^d \bmod n = (m_i^e \bmod n)^d \bmod n = m_i^{ed} \bmod n = m_i^{k\varphi(n)+1} \bmod n =$$

$$= (((m_i^k)^{\varphi(n)} \bmod n) * (m_i \bmod n)) \bmod n = m_i$$

So we have *Public keys - n,e*
Private key - d.

The Author's Example

As opentext were chosen the following

ITS ALL GREEC TO ME

Then, put instead _ --0, symbol A --1, B--2, …, Z--26.
Get the following large number

***m*1 =09201900011212000718050511002015001305.**

Have chosen ***e* = 9007,**
***n*=1143816257578888676692357799761466120102182**
967212423625626518429357069352457338978305971
235639587050589890751475992900268795435 41

The Author's Example Continued

Then they have got the result:

c=19993513149780510045231712274026064742320401705839146310370371740625997160894892750439920962672582675012893554461353823769748026

To find inverse number *d*

As you remember in RSA algorithm we have to choose inverse number d, such, that

$$de \equiv 1 (\mathrm{mod}\ (p-1)(q-1))$$

To find this number we can use the following theorems

Euclid's algorithm

Let we have $r_0 > r_1 > 0$. To find their largest common divider(LCD) we can calculate following

$$r_0 = q_1 r_1 + r_2, \quad 0 < r_2 < r_1$$

$$r_1 = q_2 r_2 + r_3, \quad 0 < r_3 < r_2$$

.....

$$r_{m-2} = q_{m-1} r_{m-1} + r_m, \quad 0 < r_m < r_{m-1}$$

$$r_{m-1} = q_m r_m$$

So LCD(r_0,r_1)=r_m

Extended Euclid's algorithm

Let consider number's sequence $\{t_i\}_{i=0}^{m}$

Where

$$t_0 = 0,\ t_1 = 1,\ t_j = t_{j-2} - q_{j-1}t_{j-1},\ \ j \geq 2 \qquad (4)$$

Theorem.If LCD(r_0,r_1)=1, then

$$r_1 t_m \equiv 1 (\mathrm{mod}\, r_0) \qquad (5)$$

or

$$t_m \equiv r_1^{-1} (\mathrm{mod}\, r_0)$$

Our example

Choose p=11, q=17. Then $n = pq = 187$,
$\varphi(n) = (p-1)*(q-1)=160$.

We choose the public exponent e that
$(e,((p - 1)(q - 1)) = 1$. That is $(e,160) = 1$. Let $e = 7$.
Number d we have to find from this equation:

$$7d = 1 \bmod 160$$

To do it we will use formula 5(Extended Euclid's algorithm) from previous slide.

Example Continued

We calculate the following according Euclid's algorithm to find q_i.

160=22*7+6
7=1*6+1
6=6*1

We have:

q_1=22; q_2=1;q_3=6
t_0=0; t_1=1; t_2=0-22*1=-22; t_3=1+1*22=23;
d =t_3=23.

Example Continued

We publicate keys $e = 7$, $n = 187$, and kept in secret key $d = 23$.

As opentext we take $M=9$ for the easy calculations and find ciphertext C.

$$C = 9^7 \bmod 187 = ((9^2)^3 * 9) \bmod 187 =$$
$$= (((81)^2 \bmod 187) * (729 \bmod 187)) \bmod 187 =$$
$$= (16 \bmod 187) * (168 \bmod 187) \bmod 187 = 70$$

Example Continued

When we calculate sqr(n) we have to do it consistently and subtract multiple n from result. Then we have not number that largest sqr(n).

Receiver decrypts message as following

$70^{23} \bmod 187 = ((70^2)^{11} * 70) \bmod 187 = 9.$

Digital signature

Let Aigul(A) and Bolat(B) calculate

A: Public n_A, e_A , Private d_A
B: Public n_B, e_B , Private d_B

Aigul calculates $$c = m^{d_A} \bmod n_A$$

$$\alpha = c^{e_B} \bmod n_B$$

Bolat calculates

$$\beta = \alpha^{d_B} \bmod n_B = c^{e_B d_B} \bmod n_B = c^{\varphi(n_B)k+1} \bmod n_B =$$
$$= (c^k)^{\varphi(n_B)} c \bmod n_B = c$$

Pitfalls using RSA

Example 1. Bolat has ciphertext c, but she cannot decrypt it. He chooses any casual number $r < n$ and calculates

$$x = r^{e_A} \bmod n_A, y = xc \bmod n_A, t = r^{-1} \bmod n_A$$

Clearly that $r^{-1}x^{d_A} \equiv 1(\bmod n_A)$

Pitfalls Continued

Now Bolat asks Alice to sign y and receive

Then Bolat calculates $u = y^{d_A} \bmod n_A$

$$tu \bmod n_A = r^{-1} y^{d_A} \bmod n_A = r^{-1} x^{d_A} c^{d_A} \bmod n_A =$$
$$= c^{d_A} \bmod n_A = m$$

Next example

Example 2. Eve asks Alice to sign message m_1, m_2. And receives

$$m_1^{\,d_A} \bmod n_A \qquad m_2^{\,d_A} \bmod n_A$$

Then she calculates $$C = (m_1 m_2)^{d_A} \bmod n_A$$

So Alice signs $$m_3 = m_1 m_2$$

Last example

Example 3. Aigul sends signing secret message to Bolat. But she at first encrypts m by open Bolat's and then signs by her private key.

$$\left(m^{e_B} \bmod n_B\right)^{d_A} \bmod n_A$$

Bolat finds number x, such that

$$(m')^x = m \bmod n_B$$

Now he publicate his new public keys xe_B, n_B and tells that Aigul signs m'

Problem

RSA never used in practice for encrypting a message. DES 1000 times faster than RSA. The message that can be encrypted using RSA limited by the size of n.

There are obvious solution for this problem.

Solution

Secret key K is encrypted using RSA. Message m is encrypted by any block or stream cipher algorithm by secret key K. Only a single RSA operation is required.

Hybrid algorithms

This algorithm is called hybrid algorithm.
See how Bolat can send to Aigul their secret key.

Bolat
1. Chooses a random number x from the range $\{0,\ldots,2^l-1\}$, where l is largest number such that $2^l<n$.
2. Finds $K=h(x)$, where h hash function.
3. Encrypts random number x by Bolat's public key and send to Aigul $c = x^e \bmod n$.

Continue

Receiver(Aigul)
1. Aigul calculates $x = c^d \bmod n$.
2. Finds $K=h(x)$.

Diffie-Hellman

RSA gains its security from the difficulty of factorising the product of two large prime numbers.

The Diffie-Hellman key exchange protocol depends on the difficulty of computing discrete logarithms over a finite field:

i.e., if $y = a^x \bmod n$

it is easy to compute y given x, but very difficult to compute x given y.

In this case, the mathematics is reasonably straightforward!

The Diffie-Hellman Protocol

Alice and Bob agree on a large prime p, and a generator g.
These do not need to be kept secret.
The protocol:
Alice chooses a random large integer x, computes $X = g^x \bmod p$ and sends X to Bob.
Bob chooses a large random integer y, computes $Y = gy \bmod p$ and sends Y to Alice.
Alice computes $k = Y^x \bmod p$
Bob computes $k' = X^y \bmod p$
Now both k and k' are equal to $g^{xy} \bmod p$.
All an eavesdropper will see is g, p, X and Y