Introduction to Cryptology

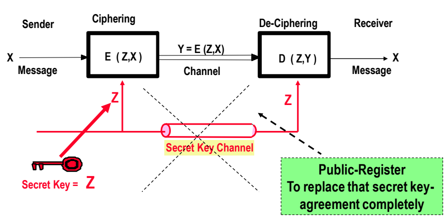
Lecture-10
Public-Key Cryptography
RSA Rivest-Shamir-Adelmann Public-Key System

09.05.2023, v41

Lecture Outlines

- Historical Overview !
- RSA Public-Key Encryption System
- RSA Public-Key Signature System
- RSA Security considerations

The Target of Public Key Cryptography

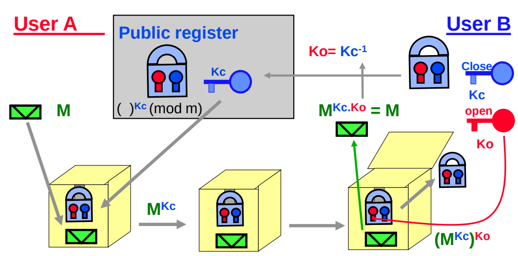


Basic Public Key Secrecy System (RSA system1978)

RSA: Rivest-Shamir-Adleman, MIT, USA

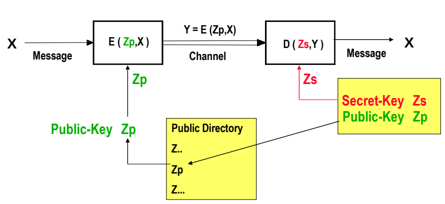
(Mechanical Lock simulation: user A sends a message to B)

All operations in Z_m



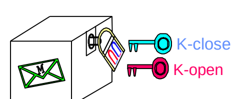
Conventional Public-Key Crypto-system

(using asymmetric keys)



Public-Key Secrecy System RSA 1978

(Rivest Shamir Adelmann) MIT, USA !!



Trap-door One Way Function !

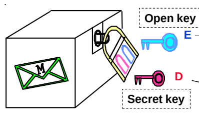
RSA key idea to implement such a lock: is based mainly on Euler theorem and on the two unproved claims:

1. Euler function for any integer m is only computable if the factorization of m is known.

   for $m = p_1^{e_1}\, p_2^{e_2}\, p_3^{e_3} \ldots\, p_t^{e_t} \longrightarrow \varphi(m) = m\,(1-\frac{1}{p_1})\,(1-\frac{1}{p_2})\ldots$

2. Factorization is considered as computationally hard and unsolved problem

RSA-Lock (Hiding Function) Uses Exponentiation in the Ring Z_m
Where m = p.q , p and q are two large secret primes



Open key

E

D

Secret key

M^E (mod m) **ENCRYPTION**

$(M^E)^D$ (mod m) **DECRYPTION**

$$(M^E)^{D \bmod \varphi(m)} = M^{E.D \bmod \varphi(m)} = M$$

To get M, the following should hold: E . D = 1 or $D = E^{-1}$ in the exponent
That is E and D should be invertible modulo φ(m) ! Or gcd (E, φ(m)) =1

Security Considerations: m is a large composite (**m=p q**), p and q are two large secret primes. To break the system φ(m) is required to compute **$D = E^{-1}$ modulo φ(m).** However, φ(m) can only be computed if p and q are known. Therefore, the system can only be broken if and only if: m can be factored OR φ(m) can be found somehow without factorization!

Design Template for RSA Public Key Secrecy System

Open directory

User A
N_a
E_a

User B
N_b
E_b

⋮

USER A:

$N_a = p_a . q_a$ open modulus of A
$p_a . q_a$ tow secret large primes
$\varphi(N_a) = (p_a-1).(q_a-1)$

E_a = open Encryption key of A
$D_a = E_a^{-1}$ [mod $\varphi(N_a)$]
Condition: gcd [E_a , $\varphi(N_a)$] = 1

Number of possible keys = $\varphi[\varphi(N_a)]$

USER B:

$N_b = p_b . q_b$ open modulus of B
$p_b . q_b$ tow secret large primes
$\varphi(N_b) = (p_b-1).(q_b-1)$

E_b = open Encryption key of B
$D_b = E_b^{-1}$ [mod $\varphi(N_b)$]
Condition: gcd [E_b , $\varphi(N_b)$] = 1

Number of possible keys = $\varphi[\varphi(N_b)]$

A sends Message M to B:

$Y = M^{E_b}$ mod N_b (Encrypt) → $Y^{D_b} = M^{E_b D_b}$ mod N_b = M (Decrypt)

Security of RSA Public Key System

Is Exponentiation $y = a^x$ in Z_m a One-Way Function ?
- **Theoretically** ***not*** **(no proof that φ(m) is not computable if we do not know p and q !!)**
- **Practically φ(m)** ***computation is dificult*** **if : m is a product of two large strong primes!**

RSA system can be broken by:
1. Factoring m = p . q
2. Computing φ(m) somehow without factoring m.

However, factorization is computationally equivalent to computing Euler function φ(m)
Proof:

$\varphi(m) = (p-1)(q-1) = m - p - q + 1$

$\Rightarrow$ s = (p + q) = m - φ(m) + 1
m = p . q

$\Rightarrow$ p or q = $(s \pm \sqrt{s^2 - 4m})/2$

RSA Security and State of the Art in Factorization

No consistent and reliable answer (only claims according to the state of the art!):

In general:

Factorization Complexity is O (√m)

That is, if the modulus **m** is an integer in the range of 2^n bits
To factor m, a **computational complexity** proportional to $2^{n/2}$ is required·

! There are still ongoing secret and open research on factorization!
! Therefore, there are published results and unpublished results!

In the public literature;
In number theory, the general number field sieve (GNFS) is the most efficient classical algorithm known for factoring integers larger than 10^{100}. Heuristically, its complexity for factoring an integer n (consisting of $\lfloor \log_2 n \rfloor + 1$ bits) is of the form:

$$\exp\left(\left(\sqrt[3]{\frac{64}{9}} + o(1)\right)(\ln n)^{\frac{1}{3}}(\ln \ln n)^{\frac{2}{3}}\right)$$

Factorization is a business of mathematicians !

Designing adequate and good RSA cryptosystem

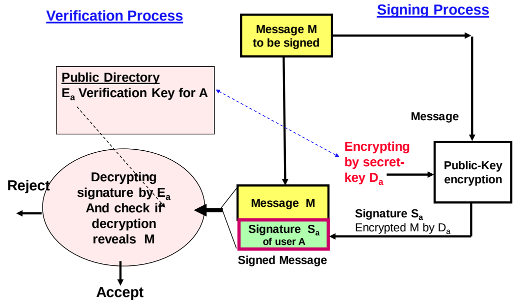
1. How to choose large primes *p,q* *for the modulus m=pq*?
Select primes randomly by using "**miller test**" or "**Pocklington theorem**" or other refined versions for generating primes.

2. Relationship between *p* and *q*
- Difference |*p*-*q*| should be neither too small nor too large.
- gcd(*p*-1, *q*-1) should not be large.
- Both *p*-1 and *q*-1 should contain large prime factors (strong primes). The ideal case is: *q*, *p* should be strong primes - such that (*p*–1)/2 and (*q*-1)/2 are primes.
Examples: **83** = 2x41 + 1 , **107**= 2x53 + 1

3. Selecting *e* and *d* ?
- Neither ***d*** nor ***e*** should be small.
- ***d*** should not be smaller than $n^{1/4}$.
(For $d < n^{1/4}$ a polynomial time algorithm may determine *d*).

Many other considerations and refinements may appear according to the current state of the open research!

Public-Key Signature Scheme



Design Template for the RSA Public Key Signature System

USER A:

$N_a = p_a \cdot q_a$ open modulus of A
$p_a \cdot q_a$ two secret large primes
$\varphi(N_a) = (p_a-1).(q_a-1)$

E_a = open Encryption key of A
$D_a = E_a^{-1}$ [mod $\varphi(N_a)$]

gcd [E_a , $\varphi(N_a)$] = 1

Open directory

User A
N_a
E_a

User B
N_b
E_b

USER B:

$N_b = p_b \cdot q_b$ open modulus of B
$p_b \cdot q_b$ twosecret large primes
$\varphi(N_b) = (p_b-1).(q_b-1)$

E_b = open Encryption key of B
$D_b = E_b^{-1}$ [mod $\varphi(N_b)$]

gcd [E_b , $\varphi(N_b)$] = 1

A signs Document M for B:

M^{D_a} = S mod N_a $\xrightarrow{(M,S)}$ $S^{E_a} = M^{D_a E_a}$ mod N_a=M´ **(Verify M´=M)?**

Signed Message

If M´= M then the signature is true

13

Example: Construct RSA secrecy and signature system using the two prime pairs 11, 5 and 3,11. Encrypt the message M=2 sent to user B. Let B signs M and send his signature back to A.

Solution:

USER A:

N_a = 11 x 5 open modulus of A
$p_a \cdot q_a$ two secret large primes
$\varphi(N_a) = (p_a-1).(q_a-1)$
= (11-1)(5-1) = 40

7= E_a= open Encryption key of A
gcd [E_a , $\varphi(N_a)$] = 1
gcd(7,40)=1

D_a = 7^{-1} [mod 40)] =23

Open directory

User A
55
7

User B
33
3

USER B:

N_b = 3. 11 open modulus of B
$p_b \cdot q_b$ two secret large primes
$\varphi(N_b) = (p_b-1).(q_b-1)$ =20

3 = E_b=open Encryption key of B
gcd [E_b , $\varphi(N_b)$] = 1
gcd(3,20)=1

$D_b = E_b^{-1}$ [mod $\varphi(N_b)$] =7

A sends Message M=5 to B:

$Y = 5^3$ mod 33= 26 (A Encrypts M) $\xrightarrow{Y = 26}$ 26^7 mod 33 = 5 = M (Decrypt)

Security Gap: Notice that anybody can decrypt S to disclose M! Any other solution?

M'=14^3 mod 33 = 5 = M (Verify) $\xleftarrow{S = 14}$ S= 5^7 mod 33 = 14 (B signes M)

14

Live Example: sending a secret document M from B to A encrypted as Y_{ba} expecting B to sign it back securely. Then A decrypts it and encrypts it to B as Y_{ab} and signs Y_{ab} to B.

Solution:

USER A:

N_a = 13 x 11 open modulus of A
$p_a \cdot q_a$ two secret large primes
$\varphi(N_a) = (p_a-1).(q_a-1)$
= (13-1)(11-1) = 120

43= open Encryption key of A
gcd [E_a , $\varphi(N_a)$] = 1
gcd(43,120)=1

D_a= 43^{-1} mod 120=-53=-53+120= 67

Open directory

User A
143
43

User B
77
17

USER B:

N_b = 7. 11 open modulus of B
$p_b \cdot q_b$ two secret large primes
$\varphi(N_b) = (p_b-1).(q_b-1)$ =60

17 = open Encryption key of B
gcd [E_b , $\varphi(N_b)$] = 1
Gcd(3,20)=1

$D_b = E_b^{-1}$ [mod $\varphi(N_b)$] =-7=53

B encrypts Message M=24 to A: (1)

$Y_{ba} = 24^{43}$ mod 143 = 41
(B Encrypts M)

A decrypts Y_{ba}: (2)

$\xleftarrow{Y_{ba} = 41}$

$M=Y^{Da} = 41^{67}$ mod 143 = 24
(Decrypt)

A encrypts M back to B as Y_{ab} then signs Y_{ab}.: (3)

$Y_{ab}=M^{Eb}$ mod 77= 24^{17} mod 77=40
S = $(Y_{ab})^{Da}$ mod 143
S = $(40)^{67}$ = 105
(Signing the encrypted M to B)

$\xrightarrow{40,\ S=105}$ (Signature)

B verifies: (4)

105^{43} mod 143 = 40= Y_{ab}
M= 40^{53} mod 77 = 24
(B checks)

15