RSA algorithm

RSA Public-Key Encryption Algorithm

- One of the first, and probably best known public-key scheme;
- It was developed in 1977 by R.Rivest, A.Shamir and L. Adleman;
- RSA is a block cipher in which the plaintext and ciphertext are **integers** between **0** and **k-1**, where **k** is some number;
- Every integer can be represented, of course, as a sequence of bits;

Encryption and decryption in RSA

- **Encryption**

$$C = M^e \bmod n$$

- **Decryption**

$$M = C^d \bmod n = (M^e)^d \bmod n = M^{ed} \bmod n$$

Here M is a block of a plaintext, C is a block of a ciphertext and e and d are some numbers. Sender and receiver know *n* and *e*. Only the receiver knows the value of *d*.

Private and Public keys in RSA

- Public key KU = {e,n};
- Private key KR = {d,n};

Requirements:

- It is possible to find values *e,d,n* such that
- It is easy to calculate

Requirements

- It is possible to find values *e,d,n* such that
  $$M^{ed} = M \bmod n \text{ for all } M < k$$
  (key generation) , where k is some number , k < n
- It is easy to calculate M^e and C^d modulo *n*
- It is difficult to determine *d* given *e* and *n*

Key generation

- Select two prime numbers *p* and *q;*
- Calculate *n* = *p* x *q*;
- Calculate $\phi(n)$ = (p-1)(q-1);
- Select integer *e* less than $\phi(n)$ and relatively prime with $\phi(n)$;
- Calculate *d* such that $de \bmod \phi(n) = 1$;
- Public key *KU* = *{e,n}*;
- Private key *KR* = *{d,n}*;

Fermat – Euler Theorem

Correctness of RSA can be proved by using Fermat-Euler theorem:

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

Where *p* is a prime number *and* $x \neq 0 \bmod p$

Chinese Remainder Theorem

For relatively prime *p* and *q* and any *x* and *y*

$$x = y \bmod p$$
$$x = y \bmod q$$

Implies

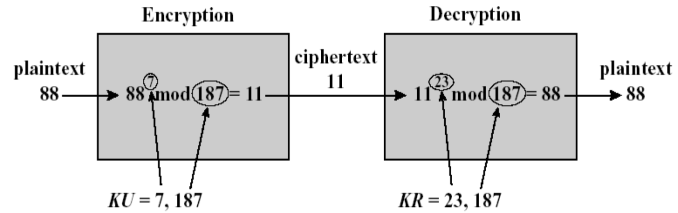
$$x = y \bmod pq$$

Example

- Select two prime numbers, p = 17, q = 11;
- Calculate $n = pq$ = 187;
- Calculate $\phi(n)$ = 16 x 10 = 160;
- Select e less than 160 and relatively prime with 160;
- Determine d such that de mod 160 = 1 and d < 160. The correct value is d = 23, indeed 23 x 7 = 161 = 1 mod 160.
- Thus KU = {7,187} and KR = {23,187} in that case.

Encryption and decryption

Let a plaintext be M = 88; then encryption with a key {7,187} and decryption with a key {23,187} go as follows



How to break RSA

- **Brute-force approach**: try all possible private keys of the size *n.* Too many of them even for moderate size of *n;*
- **More specific approach**: given a number *n,* try to find its two prime factors *p* and *q;* Knowing these would allow us to find a private key easily.

Security of RSA

Relies upon complexity of factoring problem:

- Nobody knows how to factor the big numbers in the reasonable time (say,in the time polynomial in the size of (binary representation of) the number;
- On the other hand nobody has shown that the fast factoring is impossible;

RSA challenge

RSA Laboratories to promote investigations in security of RSA put a challenge to factor big numbers. Least number, not yet factored in that challenge is 704 bit, or 212 decimal digit number

74037563479561712828046796097429573142593188889231289084936232638972765034028266276891996419625117843995894330502127585370118968098286733173273108930900552505116877063299072396380786710086096962537934650563796359

Cache prize is 30000 USD