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 n-1, where n is some number;
• Every integer can be represented, of course, as a sequence of bits;

Encryption and decryption in RSA

• Encryption
\[ C = M^e \mod n \]

• Decryption
\[ M = C^d \mod n = (M^e)^d \mod n = M^{ed} \mod n \]

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 $M^{ed} = M \mod n$ 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 $n = p \times q$;
- Calculate $\phi(n) = (p-1)(q-1)$;
- Select integer $e$ less than $\phi(n)$ and relatively prime with $\phi(n)$;
- Calculate $d$ such that $de \mod \phi(n) = 1$;
- Public key $KU = \{e, n\}$;
- Private key $KR = \{d, n\}$.

Fermat – Euler Theorem

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

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

Where $p$ is a prime number and $x \neq 0 \mod p$.

Chinese Remainder Theorem

For relatively prime $p$ and $q$ and any $x$ and $y$:

$$x = y \mod p$$
$$x = y \mod q$$

Implies

$$x = y \mod pq$$
Example

- Select two prime numbers, \( p = 17 \), \( q = 11 \);
- Calculate \( n = pq = 187 \);
- Calculate \( \phi(n) = 16 \times 10 = 160 \);
- 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 \( 23 \times 7 = 161 = 1 \mod 160 \).
- 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

RSA-230 =
17969491597941066732916128449573246156367561808
0126000708891883553172664034149093349337224786
86507552308558641999292218144366847228740520652
57937495694348389263171152522525654410980819170
611742509702440718010364831638288518852689

762 bits, or 230 decimal digits

RSA challenge, recent news

RSA-220 =
226013852620340578494165404861019751350803891517197767183211977681
09445941817966676608593121306582577250631582886676637448070001811
14571186300211245792819946746296607013109659864098332798280356037
920539198013399446496955261 = 6863564122675662743823714992884378001308422399791648446212449933
215410614414642667938213644208420192054999867
x
32929074394863498120493015492128352919164551965362339524526880511
692903449049652463373924866390738191765712603

(> ~300 CPU years, S. Bai et al., May 2016 )