RSA – Algorithm

Definition:

Block cipher asymmetric algorithm developed by Rivest, Shamir & Adleman . It is the best known & widely used public-key scheme and based on exponentiation in a finite (Galois) field over integers modulo a prime. Its security due to cost of factoring large numbers

Factorization takes $O(e^{\log n \log \log n})$ operations (hard)

Each user will be provided with pair of keys one of which is public key used for encryption and the other is private used for decryption. Plaintext and cipher text are integers between 0 and n – 1 for some n. (eg . 1024 bits)

Ingredients of RSA Algorithm

The ingredients are the following:

p, q, two prime numbers	(private, chosen)
n = pq	(public, calculated)
e, with gcd(ϕ (n),e) = 1; $1 < e < \phi(n)$	(public, chosen)
$d \equiv e^{-1} (\text{mod } \phi (n))$	(private, calculated)

RSA Key Setup:

This key setup is done once (rarely) when a user establishes (or replaces) their public key.

1. Each user generates a public/private key pair by: selecting two large primes at random - p, q
2. Computing their system modulus N=p .q
3. ø (N)= (p-1) (q-1)
4. Selecting at random the encryption key e where 1< e < ø(N), gcd (e ,ø (N)) =1
5. Solve following equation to find decryption key d
   e. d=1 mod ø(N)
6. Publish their public encryption key: KU = {e, N}
7. Keep secret private decryption key: KR = {d, p, q}

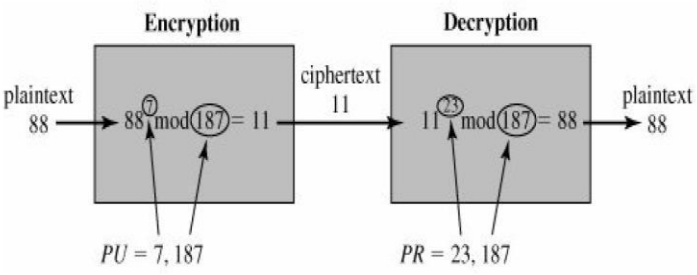
RSA Use

8. To encrypt a message M the sender:
   - obtains public key of recipient KU={e ,N}
   - computes: $\mathbf{C=M^e \bmod N}$, where $0 \leq M < N$
9. To decrypt the ciphertext C the owner:
   - uses their private key KR={d,p,q}
   - computes: $\mathbf{M=C^d \bmod N}$

Example:

1. Select primes: p=17 & q=11
2. Compute $n = pq$ =17×11=187
3. Compute ø(n)=(p–1)(q-1)=16×10=160
4. Select e *:* gcd(e,160)=1; choose e=7

5. Determine d*:* $de=1 \bmod 160$ and $d < 160$ Value is d=23 since $23\times7=161$
6. Publish public key KU={7,187}
7. Keep secret private key KR={23,17,11}
8. Given message M = 88 (88<187)
9. Encryption:C = $88^7 \bmod 187 = 11$



$88^7 \bmod 187 = [(88^4 \bmod 187) \times (88^2 \bmod 187) \times (88^1 \bmod 187)] \bmod 187$

$88^1 \bmod 187 = 88$

$88^2 \bmod 187 = 7744 \bmod 187 = 77$

$88^4 \bmod 187 = 59969536 \bmod 187 = 132$

$88^7 \bmod 187 = (88 \times 77 \times 132) \bmod 187$

$= 894432 \bmod 187 = 11$

10. Decryption: $M = 11^{23} \bmod 187 = 88$

Note : Finding private key d (ie) multiplicative inverse of e^{-1} using extended Euclidean algorithm) ie) $d \equiv e^{-1} \bmod \Phi(n)$

$d^* e \equiv 1 \bmod \Phi(n)$ Here $d * 3 \equiv 1 \bmod 160$

According Extended Euclidean algorithm initial values

A1	**=**	**A3**	**=**
1		**A2 = 0**	**160**

B1 = 0 B2 = 1 B3 = 7

Find Q = $\lfloor$ A3/B3$\rfloor$ (take lowest nearest integer)

Then A1 = B1 ; A2= B2 ; A3 = B3

B1 = A1+QB1 ; B2 = A2+QB2; B3 = A3-QB3

Q	A1	A2	A3	B1	B2	B3
	1	0	160	0	1	7
22	0	1	7	1	22	6
1	1	22	6	1	23	1

Since B3 = 1 ;

Multiplicative inverse B2 = 23

d * 3 ≡ 1 mod 160

23 * 7 ≡ 1 mod 160

d = 23

Computational aspects of RSA

This includes i) Encryption / decryption ii) Key generation.

Both encryption and decryption in RSA involve raising an integer to an integer power, mod *n*. we can make use of a property of modular arithmetic: [(*a* mod *n*) x (*b* mod *n*)] mod *n* = (*a* x *b*) mod *n*

Efficient Operation Using the Public Key

To speed up the operation of the RSA algorithm using the public key, a specific choice of *e* is usually made. The most common choice is 65537 two other popular choices are 3 and 17.

Key generation :

Each participant must generate a pair of keys.

This involves the following tasks:

- Determining two prime numbers, *p* and *q*
- Selecting either *e* or *d* and calculating the other

---

1. Pick an odd integer *n* at random (e.g., using a pseudorandom number generator).
2. Pick an integer $a < n$ at random.
3. Perform the probabilistic primality test, such as Miller-Rabin, with *a* as a parameter. If *n* fails the test, reject the value *n* and go to step 1.
4. If *n* has passed a sufficient number of tests, accept *n*; otherwise, go to step 2.