RSA Algorithm

Euler Totient Function $\varnothing(n)$

- to compute ø(n) need to count number of elements to be excluded
- in general need prime factorization, but
  - for p (p prime) $\varnothing(p) = p-1$
  - for p.q (p,q prime) $\varnothing(p.q) = (p-1)(q-1)$
- eg.
  - $\varnothing(37) = 36$

RSA

- by Rivest, Shamir & Adleman of MIT in 1977
- best known & widely used public-key scheme
- based on exponentiation in a finite (Galois) field over integers modulo a prime
  - nb. exponentiation takes $O((\log n)^3)$ operations (easy)
- uses large integers (eg. 1024 bits)
- security due to cost of factoring large numbers
  - nb. factorization takes $O(e^{\log n \log \log n})$ operations (hard)

RSA algorithm

- Picks (randomly) two large prime numbers and calls them **p and q**.
- Calculates their **product** and calls it **n**.
- Calculates the **totient of n** ; it is simply **(p –1)(q –1).**
- Picks a random integer that is coprime to **ϕ(n)** and call this e.
- A simple way is to just pick a **random number>max(p,q)**
- Calculates (via the Euclidean algorithm) the **multiplicative inverse of e modulo ϕ(n)** and call this number d.

- P=47 Q=71,   N = P * Q = 3337
- **ϕ(n) = (p -1) (q-1) = 3220 Factors 2,2,5,7 and 23 (2*2*5*7 = 3220)**
- **Choose e such that it is none of the factors of e is 2,5,7,23**
  - **Eg. 4 cannot be chosen because 2 is the factor of 4**
  - **15 cannot be chosen because 5 is a factor of 15**
  - **E =79 ok, because it does not have the above factors**
- **E is the encryption key (Public key)**
- **Choose d (Decryption key) the private key such that**

  (d * e) mod (p-1) * (q-1) = 1,   (d*79) mod (46) * (70) =1

  (d * 79) mod (3220) =1   d= 1019

  since (1019 * 79) mod (3220) = 80501 mod 3220 = 1

Encryption

M = 688 , CT = 688^{79} mod 3337 = 1570

Send 1570 to receiver

Decryption

C=1570 , m = 1570^{1019} mod 3337 = 688

Example

- Choose **$p = 3$ and $q = 11$**
- Compute **$n = p * q = 3 * 11 = 33$**
- Compute **$\phi(n) = (p - 1) * (q - 1) = 2 * 10 = 20$**
- Choose e such that **$1 < e < \phi(n)$** and e and n are coprime. Let **$e = 7$**
- Compute a value for d such that **$(d * e) \% \phi(n) = 1$**. One solution is **$d = 3$ [$(3 * 7) \% 20 = 1$]**
- Public key is **$(e, n) \Rightarrow (7, 33)$**
- Private key is **$(d, n) \Rightarrow (3, 33)$**
- The encryption of ***$m = 2$ is $c = 2^7 \% 33 = 29$***
- The decryption of ***$c = 29$ is $m = 29^3 \% 33 = 2$***

RSA algorithm

- **Select two large prime numbers p, q**
- **Compute**

  $n = p \times q$

  $\phi(n) = (p-1) \times (q-1)$

- **Select small odd integer k relatively prime to v $\gcd(k, \phi(n)) = 1$**
- **Compute d such that**

  $(d \times k)\% \phi(n) = (k \times d)\% \phi(n) = 1$

- **Public key is (k, n)**
- **Private key is (d, n)**

- example

  $p = 11$

  $q = 29$

  $n = 319$

  $\phi(n) = 280$

  $k = 3$

  $d = 187$

- public key

  $(3, 319)$

- private key

  $(187, 319)$

Encryption and decryption

- Alice and Bob would like to communicate in private
- Alice uses RSA algorithm to generate her public and private keys
  - Alice makes key (k, n) publicly available to Bob and anyone else wanting to send her private messages
- Bob uses Alice's public key (k, n) to encrypt message M:
  - compute $E(M) = (M^k)\%n$
  - Bob sends encrypted message $E(M)$ to Alice
- Alice receives $E(M)$ and uses private key (d, n) to decrypt it:
  - compute $D(M) = (E(M)^d)\%n$
  - decrypted message $D(M)$ is original message M

Outline of implementation

- **RSA algorithm for key generation**
  - select two prime numbers p, q
  - compute $n = p \times q$
    $v = (p-1) \times (q-1)$
  - select small odd integer k such that
    $\gcd(k, v) = 1$
  - compute d such that
    $(d \times k)\%v = 1$
- **RSA algorithm for encryption/decryption**
  - encryption: compute $E(M) = (M^k)\%n$
  - decryption: compute $D(M) = (E(M)^d)\%n$

RSA algorithm for key generation

- **Input: none**

- **Computation:**
  - select two prime integers p, q
  - compute integers $n = p \times q$

    $\boldsymbol{\phi(n)} = (p-1) \times (q-1)$
  - select small odd integer k such that $\gcd(k, \boldsymbol{\phi(n)}) = 1$
  - compute integer d such that $(d \times k) \% \boldsymbol{\phi(n)} = 1$

- **Output:** n, k, and d

RSA algorithm for encryption

- Input: integers k, n, M
  - M is integer representation of plaintext message
- Computation:
  - let C be integer representation of ciphertext

$$C = (M^k)\%n$$

- Output: integer C
  - ciphertext or encrypted message

RSA algorithm for decryption

- Input: integers d, n, C
  - C is integer representation of ciphertext message
- Computation:
  - let D be integer representation of decrypted ciphertext

$$D = (C^d)\%n$$

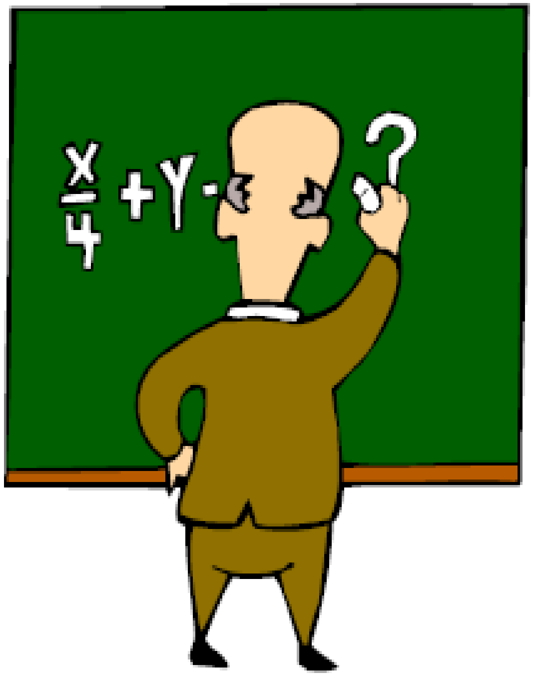
- Output: integer D
  - decrypted message

RSA Security

- three approaches to attacking RSA:
  - brute force key search (infeasible given size of numbers)
  - mathematical attacks (based on difficulty of computing ø(N), by factoring modulus N)
  - timing attacks (on running of decryption)

Math-Based Key Recovery Attacks

- Three possible approaches:
  1. Factor $n = pq$
  2. Determine $\Phi(n)$
  3. Find the private key d directly



- All the above are equivalent to factoring n

Knowing $\Phi(n)$ Implies Factorization

- Knowing both n and $\Phi(n)$, one knows

$$n = pq$$
$$\Phi(n) = (p-1)(q-1) = pq - p - q + 1$$
$$= n - p - n/p + 1$$
$$p\Phi(n) = np - p^2 - n + p$$
$$p^2 - np + \Phi(n)p - p + n = 0$$
$$p^2 - (n - \Phi(n) + 1)\, p + n = 0$$

- There are two solutions of p in the above equation.
- Both p and q are solutions.

Solve

- ***p = 7, q = 17, e = 5 ,M = 19 find d and c***
- ***p = 5, q = 11, e = 3, Find d***

M	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
C	1	8	27	9	15	51	13	17	14	10	11	23	52	49	20	26	18	2
M	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
C	39	25	21	33	12	19	5	31	48	7	24	50	36	43	22	34	30	16
M	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54
C	53	37	29	35	6	3	32	44	45	41	38	42	4	40	46	28	47	54

- In a public-key system using RSA, you intercept the Ciphertext C = 10 sent to a user whose public key is e = 5, n = 35. What is the plaintext M?