

# Rivest-Shamir-Adleman Scheme

Reference: Algorithms book by Dasgupta, Papadimitriou, and Vazirani.

## Basis in Number Theory

- Uses modular arithmetic.
- Scheme based on two fundamental problems in Number theory:
  1. *Primality*: Given a number  $N$  determine if it is prime.
  2. *Factoring*: Given a number find its prime factors.
- The first one is tractable and the second one is not known to be tractable.
- Uses Fermat's little theorem from Number Theory.

## Modular Arithmetic

Consider the following addition and multiplication tables:

- Addition modulo 5

	0	1	2	3	4
0	0	1	2	3	4
1	1	2	3	4	0
2	2	3	4	0	1
3	3	4	0	1	2
4	4	0	1	2	3

- Multiplication modulo 5

	0	1	2	3	4
0	0	0	0	0	0
1	0	1	2	3	4
2	0	2	4	1	3
3	0	3	1	4	2
4	0	4	3	2	1

## Modular Arithmetic

- Multiplication modulo 6

	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	1	2	3	4	5
2	0	2	4	0	2	4
3	0	3	0	0	3	0
4	0	4	2	0	4	2
5	0	5	4	3	2	1

- No Multiplicative inverse for 2, 3, 4 modulo 6.

$$- \gcd(2, 6) = 2, \gcd(3, 6) = 3, \gcd(4, 6) = 2.$$

- 1 and 5 have multiplicative inverses modulo 6.

$$- \gcd(1, 6) = 1 \text{ and } \gcd(5, 6) = 1.$$

## Modular Arithmetic

- Assume all numbers are given in binary and have  $n$  bits each.

- $x \equiv y \pmod{N}$  means  $(x - y)$  is divisible by  $N$ .

- Subtraction modulo  $N$  is addition modulo  $N$ :

$$(x - y) \equiv x + (N - y) \pmod{N}.$$

- Let  $x \equiv x' \pmod{N}$  and  $y \equiv y' \pmod{N}$ . Then:

$$(x + y) \equiv (x' + y') \pmod{N}.$$

$$(xy) \equiv (x'y') \pmod{N}.$$

- Associative, commutative, and distributive laws hold as in usual arithmetic operations.
- In a sequence of arithmetic operations, reduce intermediate results modulo  $N$ .
- Addition of two numbers can be done in  $O(n)$  time.
- Multiplication of two numbers can be done in  $O(n^2)$  time.

## Multiplicative Inverses

- $x$  is the *multiplicative inverse* of  $a$  modulo  $N$  if  $ax \equiv 1 \pmod{N}$ .
- For any  $a \in \{1, 2, \dots, N - 1\}$ ,  $a$  has a multiplicative inverse if and only if  $\gcd(a, N) = 1$ .
- There is a polynomial time algorithm to decide if  $\gcd(a, N) = 1$ .
  - Euclid's algorithm with time bound  $O(n^3)$ .
- There is a polynomial time algorithm to compute the multiplicative inverse of  $a$  modulo  $N$  if the inverse exists.
  - Extended Euclid's algorithm with time bound  $O(n^3)$ .

## Modular Exponentiation

- To form  $a^b \pmod{N}$ .
- Recursive algorithm:
  - FUNCTION  $MODEXP(a, b, N)$ 
    1.  $b = 0$ : Return (1).
    2.  $b > 0$ :
      - (a)  $z = MODEXP(a, \lfloor \frac{b}{2} \rfloor, N)$ .
      - (b)  $b$  is even: Return  $(z^2 \pmod{N})$ .
      - (c)  $b$  is odd: Return  $(a \cdot z^2 \pmod{N})$ .
- Time:  $O(n^3)$  assuming  $|a| = |b| = |N| = n$ .

## Generating a Random Prime Number

- A random  $n$ -bit number has about  $\frac{1.44}{n}$  chance of being a prime number:
  - Based on a theorem by Lagrange:
    - \* The number of primes  $\leq x$  is very close to  $x/\ln x$ .
  - In the population consisting of numbers up to  $x$ , about  $x/\ln x$  are prime numbers.
  - If a number is chosen randomly and uniformly from this population, the probability that it is a prime is  $1/\ln x$ .
- Repeat Until success:
  - Choose a random  $n$ -bit number  $N$ .
  - Test if  $N$  is prime (using an efficient primality test algorithm).
- On average this will take  $O(n)$  repetitions:
  - The probability that the randomly chosen  $N$  is a prime number is at least  $\frac{1}{n}$ .
  - That is, the probability that an iteration halts is at least  $\frac{1}{n}$ .
  - The average number (expected number) of iterations is then  $O(n)$ .
  - To generate a 1024-bit prime requires testing  $\ln 2^{1024}$  randomly chosen 1024-bit numbers.
    - \*  $\ln 2^{1024}$  is approximately 712.
- The prime generation algorithm uses primality test.

## Primality Testing

- Given a number  $N$  determine if it is prime.
  - A deterministic polynomial time algorithm discovered recently.
  - A simple and more efficient randomized algorithm exists.

## Primality Test: A Randomized Algorithm

- Based on Fermat's little theorem:

– If  $p$  is prime, then for every  $1 \leq a \leq (p - 1)$ :

$$a^{p-1} \equiv 1 \pmod{p}.$$

– That is,  $a^{p-1} - 1$  is divisible by  $p$ .

- If Fermat's test is performed for a randomly chosen  $a$  in the range  $1 \leq a \leq (p - 1)$  the probability that a prime number will pass the test is 1.
- Not a necessary and sufficient condition: there are some composite numbers that satisfy the condition.
- Miller/Rabin propose additional tests to guarantee that:
  - The probability that a prime number will pass the test is 1.
  - The probability that a composite number will be declared a prime number is at most  $\frac{1}{4}$ .
- The *error probability* can be reduced considerably by repeating the test for many independently chosen values of  $a$ .

## The Rivest-Shamir-Adleman Cryptosystem

- Private communication between  $\mathcal{A}$  and  $\mathcal{B}$  using encrypted messages.
- Even if intercepted the encrypted messages cannot be decrypted by an Eavesdropper.
- Public-key encryption:
  - Each person has a public key that is published to the world and a private key known only to the person.
  - To send to  $\mathcal{B}$  a secret message,  $\mathcal{A}$  encrypts it using  $\mathcal{B}$ 's public key and transmits the message to  $\mathcal{B}$ .
  - On receipt of the encrypted message,  $\mathcal{B}$  decodes it using his/her private key.

## The Rivest-Shamir-Adleman Cryptosystem

- Messages treated as numbers modulo  $N$ .
- Messages larger than  $N$  broken into smaller pieces.
- The public and private keys of  $\mathcal{B}$ :
  - Public key: Choose two large ( $n$ -bit) primes  $p$  and  $q$ . Let  $N = p \times q$ . Let  $e$  be a  $2n$ -bit number that is relatively prime to  $(p - 1)(q - 1)$ . (Say  $e = 3$ .) Publish  $(N, e)$ .
    - \* The primes  $p$  and  $q$  are generated efficiently by randomly choosing two  $n$ -bit numbers and subjecting them to primality test.
  - Private key: The integer  $d$  which is the multiplicative inverse of  $e$  modulo  $(p - 1)(q - 1)$ .
    - \* The multiplicative inverse of  $e$  modulo  $(p - 1)(q - 1)$  is computed using the extended Euclidean algorithm.

## The Rivest-Shamir-Adleman Cryptosystem

- Sending a message  $x$  to  $\mathcal{B}$ :
  - Send  $y = x^e \bmod N$ .
- Decrypting a message by  $\mathcal{B}$ :
  - Compute  $x = y^d \bmod N$ .

## The Rivest-Shamir-Adleman Cryptosystem

An eavesdropper knows the public key  $(N, e)$  and the encrypted message  $y$ .

Security based on the *assumption*:

It is intractable to compute  $x$  such that  $y = x^e \bmod N$ .

- Trying all possible values for  $x$  takes exponential time.
- Suppose the factors  $p$  and  $q$  of  $N$  are known:
  - Then the private key  $d$  can be computed as the multiplicative inverse of the public key  $e$  modulo  $(p - 1)(q - 1)$ .
- But, it is *believed* that it is intractable to compute the factors of  $N$ .

## The Rivest-Shamir-Adleman Cryptosystem

- Let  $p$  and  $q$  be two prime numbers.
- Let  $e$  be any number that is relatively prime to  $(p - 1)(q - 1)$ .
- Let  $d$  be the multiplicative inverse of  $e$  modulo  $(p - 1)(q - 1)$ :  
$$ed \equiv 1 \pmod{(p - 1)(q - 1)}.$$
- **Claim:** For all  $x \in \{0, 1, \dots, N - 1\}$ ,  $(x^e)^d = x \pmod N$ .
- That is, if the message  $x$  is encrypted as  $y = x^e \pmod N$  then it can be decrypted as  $y^d \pmod N$ .

## The Rivest-Shamir-Adleman Cryptosystem

- To show that, for all  $x \in \{0, 1, \dots, N - 1\}$ ,  $(x^e)^d = x \pmod N$ .
- That is,  $x^{ed} - x$  is divisible by  $N$ .

$$ed \equiv 1 \pmod{(p - 1)(q - 1)}.$$

$$ed = k(p - 1)(q - 1) + 1 \text{ for some } k.$$

$x^{ed} - x$  divisible by  $q$ :

$$\begin{aligned} x^{ed} - x &= x^{k(p - 1)(q - 1) + 1} - x \\ &= x[x^{k(p - 1)(q - 1)} - 1] \\ &= x[(x^{k(p - 1)})^{(q - 1)} - 1]. \end{aligned}$$

$[(x^{k(p - 1)})^{(q - 1)} - 1]$  is divisible by  $q$ :

– By Fermat's little theorem since  $q$  is prime.

Similarly,  $x^{ed} - x$  divisible by  $p$ .

Therefore,  $x^{ed} - x$  divisible by  $N = pq$ .

## The Rivest-Shamir-Adleman Cryptosystem

Example:

- $p = 5$  and  $q = 17$  so that  $(p - 1)(q - 1) = 64$ .
- $N = p \cdot q = 85$ .
- All arithmetic modulo 85.
- Choose  $e = 3$  so that the public key for encryption purposes is  $(85, 3)$ .
- $d = e^{-1} \bmod (p - 1)(q - 1) = e^{-1} \bmod 64$ .
- $d = 43$ . (Check that  $3 \cdot 43 \equiv 1 \bmod 64$ .)
- The private key for decryption purposes is  $d$ .
- For any message  $x$  (a number between 0 and 84), the encrypted message is  $y = x^3 \bmod 85$ .
- For any encrypted message  $y$ , the decrypted message is  $y^{43} \bmod 85$ .