Белорусский государственный университет информатики и радиоэлектроники

Кафедра информатики

Лабораторная работа № 3

Асимметричная криптография. RSA.

Выполнила студентка гр. 653502: Сулима М.Ф.

Проверил ассистент КИ: Артемьев В. С.

Минск, 2019

Введение

RSA (аббревиатура от фамилий Rivest, Shamir и Adleman) — криптографический алгоритм с открытым ключом, основывающийся на вычислительной сложности задачи факторизации больших целых чисел.

Криптосистема RSA стала первой системой, пригодной и для шифрования, и для цифровой подписи.

В рамках лабораторной работы необходимо реализовать программные средства шифрования и дешифрования при помощи алгоритма RSA.

Блок-схема алгоритма

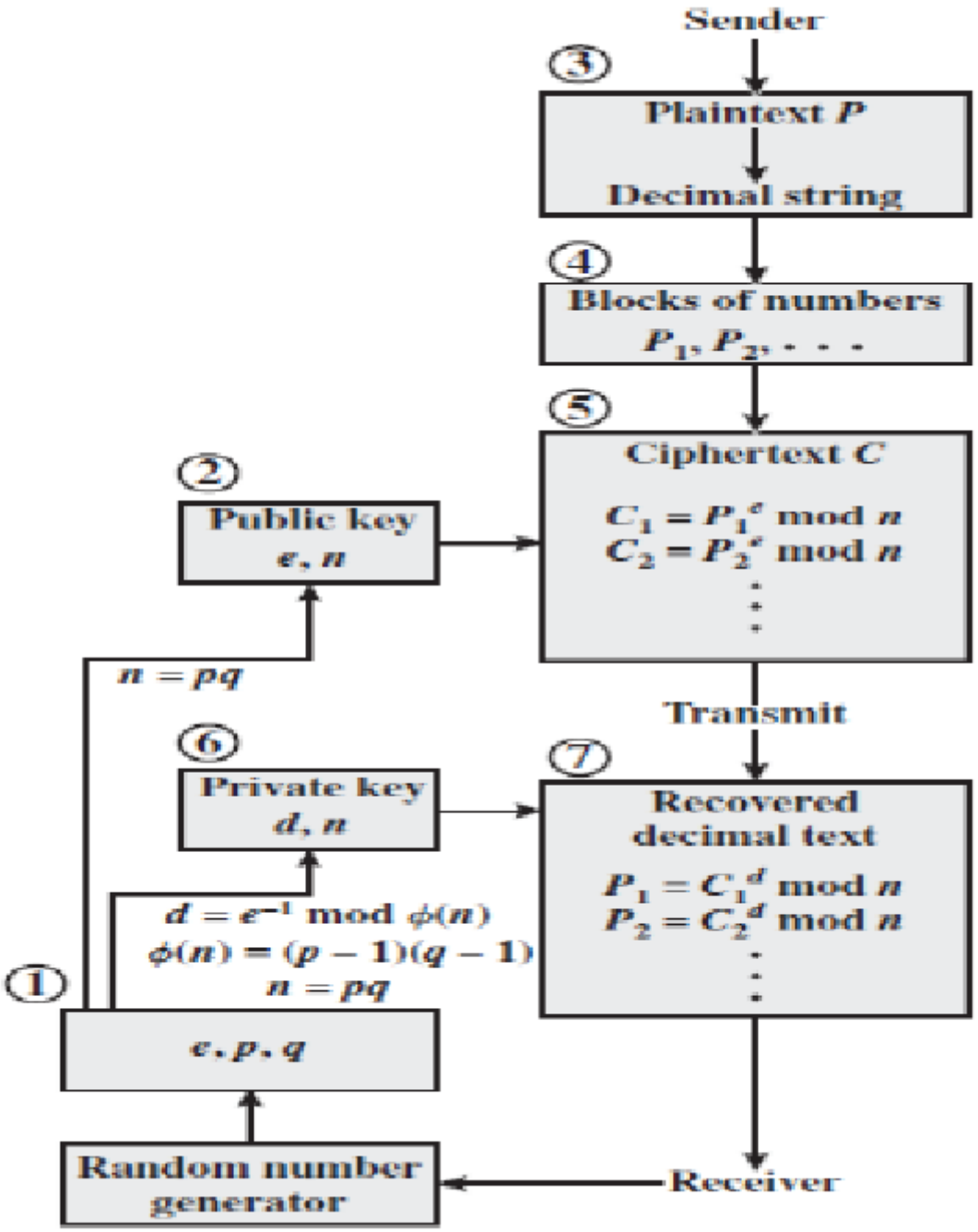

Рис.1. Схема алгоритма

Пример работы программы

```
p: 29 , q: 23
enc text :  [176, 317, 453, 137, 116, 9, 257, 271, 585, 538, 136, 52, 292,
 195, 9, 116, 137, 453, 317, 176]
dec text:  qwerty12345678ytrewq
```

Рис.2. Пример работы

Код программы

```
def encrypt(text, e, n):
    ascii_arr = char_to_ascii(text)
    enc_arr = []
    for num in ascii_arr:
        c = (num ** e) % n
        enc_arr.append(c)
    return enc_arr

def decrypt(enc_arr, d, n):
    dec_text = ''
    for num in enc_arr:
        _ascii = (num ** d) % n
        dec_text += chr(_ascii)
    return dec_text

def get_primes(start, stop):
    if start >= stop:
        return []
    primes = [2]
    for n in range(3, stop + 1, 2):
        for p in primes:
            if n % p == 0:
                break
        else:
            primes.append(n)
    while primes and primes[0] < start:
        del primes[0]
    return primes

def make_key_pair(length):
    start = 1 << (length // 2 - 1)
    stop = 1 << (length // 2 + 1)
    primes = get_primes(start, stop)
    n_min = 1 << (length - 1)
    n_max = (1 << length) - 1
    while primes:
        p = random.choice(primes)
        primes.remove(p)
        q_candidates = [q for q in primes
```

```
                    if n_min <= p * q <= n_max]
    if q_candidates:
        q = random.choice(q_candidates)
        break

m = (p - 1) * (q - 1)
e = 3
while gcd(e, m) != 1:
    e += 2
d = 1
while (e * d) % m != 1:
    d += 1
print("p:", p, ", q:", q)
return p * q, e, d
```

Вывод

В ходе написания лабораторной работы были изучены алгоритмы шифрования и дешифрования RSA, а также написаны их программные реализации. Были получены навыки усложнения и увеличения криптостойкости алгоритма RSA, а также изучены модификации и режимы работы алгоритма RSA.