Белорусский государственный университет информатики и радиоэлектроники

Кафедра информатики

Лабораторная работа № 4

Асимметричная криптография. Алгоритм Эль-Гамаля.

Выполнила студентка гр. 653502: Сулима М.Ф.

Проверил ассистент КИ: Артемьев В. С.

Минск, 2019

Введение

Схема Эль-Гамаля (Elgamal) — криптосистема с открытым ключом, основанная на трудности вычисления дискретных логарифмов в конечном поле. Криптосистема включает в себя алгоритм шифрования и алгоритм цифровой подписи. Схема Эль-Гамаля лежит в основе бывших стандартов электронной цифровой подписи в США(DSA) и России (ГОСТ Р 34.10-94).

Схема была предложена Тахером Эль-Гамалем в 1985 году. Эль-Гамаль разработал один из вариантов алгоритма Диффи-Хеллмана. Он усовершенствовал систему Диффи-Хеллмана и получил два алгоритма, которые использовались для шифрования и для обеспечения аутентификации. В отличие от RSA алгоритм Эль-Гамаля не был запатентован и, поэтому, стал более дешевой альтернативой, так как не требовалась оплата взносов за лицензию. Считается, что алгоритм попадает под действие патента Диффи-Хеллмана.

В рамках лабораторной работы необходимо реализовать программные средства шифрования и дешифрования при помощи алгоритма Эль-Гамаля.

Блок-схема алгоритма

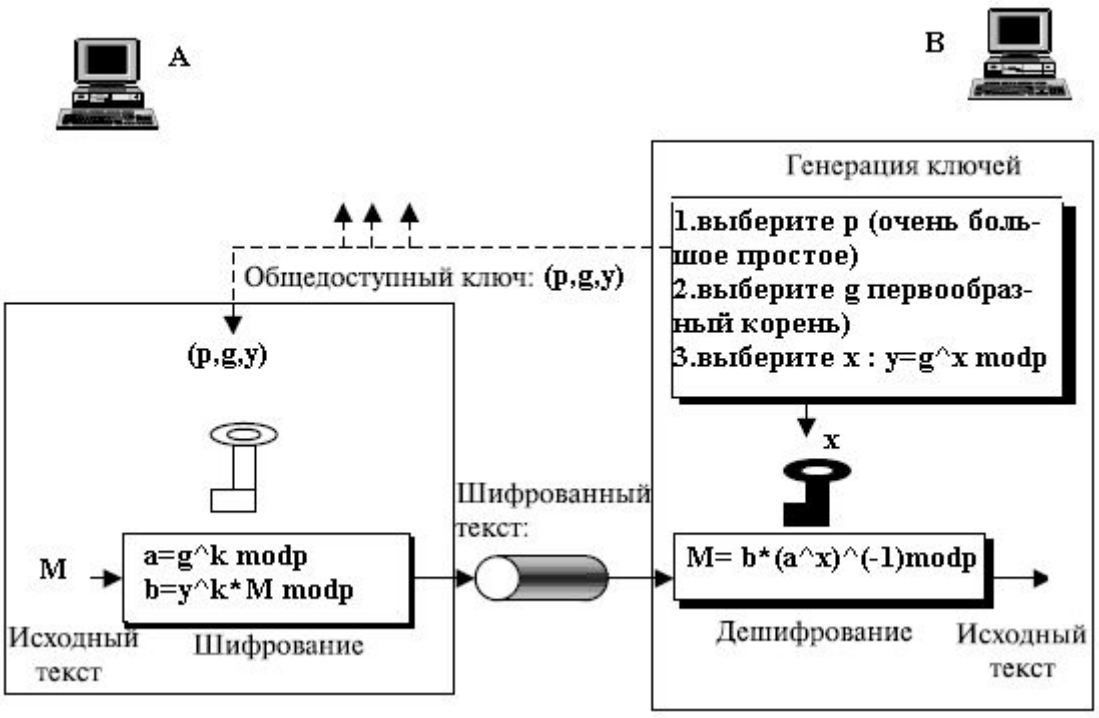

Рис.1. Схема алгоритма

Пример работы программы

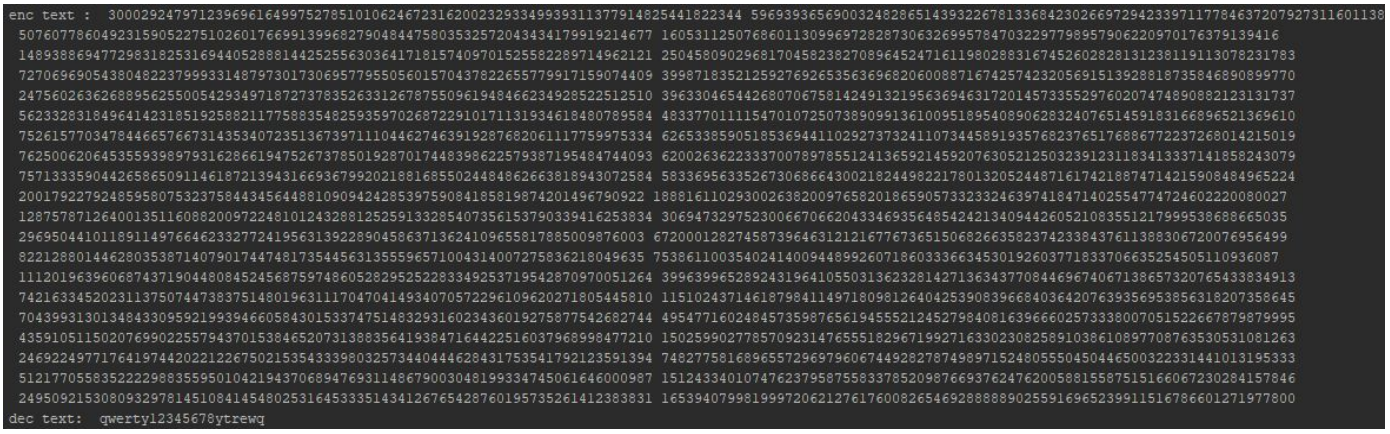

Рис.2. Пример работы

Код программы

```
def generate_keys(confidence=32):
    p = find_prime(256, confidence)
    g = pow(find_primitive_root(p), 2, p)
    x = random.randint(2, p - 1)
    y = pow(g, x, p)
    publicKey = {'p': p, 'g': g, 'y': y}
    privateKey = {'p': p, 'x': x}
    return privateKey, publicKey

def encrypt(key, plainText):
    z = bytearray(plainText, 'utf-8')
    cipher_pairs = []
    for i in z:
        k = random.randint(2, key['p'] - 1)
        a = pow(key['g'], k, key['p'])
        b = (i * pow(key['y'], k, key['p'])) % key['p']
        cipher_pairs.append([a, b])
    encryptedStr = ""
    for pair in cipher_pairs:
        encryptedStr += str(pair[0]) + ' ' + str(pair[1]) + ' '
    return encryptedStr


def decrypt(key, cipher):
    plaintext = []
    cipherArray = cipher.split()
    if len(cipherArray) % 2 != 0:
        return
    for i in range(0, len(cipherArray), 2):
        a = int(cipherArray[i])
        b = int(cipherArray[i + 1])
        s = pow(a, key['x'], key['p'])
        plain = (b * pow(s, key['p'] - 2, key['p'])) % key['p']
        plaintext.append(plain)
    decryptedText = bytearray(plaintext).decode('utf-8')
    return decryptedText
```

Вывод

В настоящее время криптосистемы с открытым ключом считаются наиболее перспективными. К ним относится и схема Эль-Гамаля, криптостойкость которой основана на вычислительной сложности проблемы дискретного логарифмирования, где по известным p, g и y требуется вычислить x, удовлетворяющий сравнению:

$$y \equiv g^x (mod\ p)$$

В ходе написания лабораторной работы были изучены алгоритмы шифрования и дешифрования Эль-Гамаля, а также написаны их программные реализации.