> 姓名：佘崧林
>
> 学号：1613574

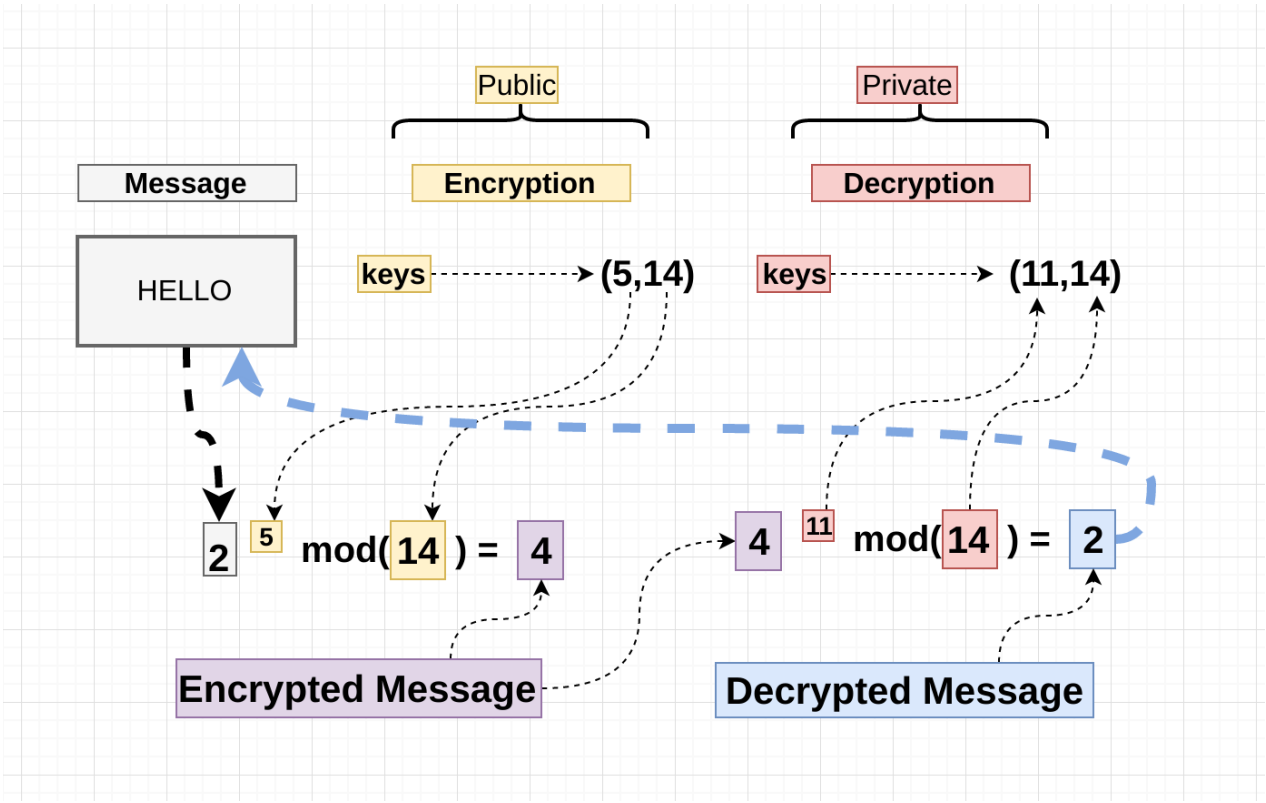
RSA

简介

RSA 是一种非对称加密，也就是需要一对密钥，公钥用于加密，私钥用于解密。

RSA 算法涉及五个关键参数: 公钥: e, N; 私钥: d, N; 明文: m; 密文: c; 其中 N 是两个大素数 p，q 的积，e、d 满足 $ed \equiv 1 \mod (\phi(N))$。

举个例子，实现 RSA 算法的流程图如下:



密钥的生成

RSA 算法密钥的生成是很麻烦的，因为生成大素数是一个不太容易的事情，同样，在破解 RSA 时，如果攻击者获得 了 n、e 和密文 c，为了破解密文必须计算出私钥 d，为此需要先分解 n。当 n 的长度为 512 比特时，在目前还是安 全的，但从因式分解技术的发展来看，512 比特并不能保证长期的安全性。为了达到更高的安全性，要求在一般的商 业应用中使用 1024 比特的长度，在更高级别的使用场合，要求使用 2048 比特长度。 本次实验的第一个关键问题就 是如何快速生成 512 比特的素数。

1. 随机数的生成：除了 2 之外的素数都是都是奇数，因此首先生成一个大奇数，然后判断它是否为一个素数，若不是，则将其加 2，用该相邻的奇数继续判断，只到通过素性检验，即可认定生成了一个大素数。

2. Miller—Rabin 素性检验

   在进行素性检验时，一般采用 Miller-Rabin 素性检验算法。若该算法返回值为false，则说明输入的 n 一定是合数; 若返回值为 true，也不能肯定 n 一定是一个质数，要多检验几次，一般来说检测 5 次，若 5 次返回值均为 true，则 可认为输入的 n 为一个质数。 Miller-Rabin 算法的理论基础:如果 n 是一个奇素数， 将 n-1 表示成 2^sr (r) a n a^r≡1(mod n) j(0≤j ≤s 1 j Z) a^(2^jr) ≡-1(mod n) 成 立。 这个理论是通过一个事实经由 Fermat 定理推导而来: n 是一个奇素数，则方程 x^2 ≡ 1 mod n 只有±1 两个 解。

实验原理

加密:

```
mpz_class Rsa::Encrypt(mpz_class m) {
    mpz_t c; mpz_init(c);
    mpz_powm(c, m.get_mpz_t(), e, n);
    mpz_class cipher = mpz_class(c);
    mpz_clear(c);
    return cipher;
}
```

解密:

```
mpz_class Rsa::Decrypt(mpz_class c) {
    mpz_t m; mpz_init(m);
    mpz_powm(m, c.get_mpz_t(), d, n);
    mpz_class message = mpz_class(m);
    mpz_clear(m);
    return message;
}
```

其他详细算法可以直接参考代码实现。

实验结果

```
$ ./bin/lab4
Generate Random Prime p:
fbea50c4fbd8a14e9d36fc3208090f38f8f59747b775c0b65aa44c06984c116c9c47903c7434933
ffc5be158a95146a34f03b06330053f4681f4ec22f11c540f
Generate Random Prime q:
b9a55fb71814373ca2300f5eead0140e02a61d8b609fd31b7f2850ede4144426313ac469c0179ac
10e0794b4113cf850efc3f521644c2d56fbe55d340b293d99
Using Most Common e: 10001
--------------------------------
Plain Text: 10
Cipher Text:
4629031ef946289bb232991c0f6522f2e4916d831da47063b30539187d8fa181dfcfc58541a8f48
edd600a18a3e4831f65f313b272f52a47cf4e312894142d0d25c582d16175691162f0f5e1521042
43bd19a1060d0081de1c52af3a3a64823a136729fe9993c2226a6dc94b07381c6f49705fa48e707
68bcaeb149d8350209d
Deplain Text: 10
```