Semester 1Exercises and Solutions for Week 11200
--

RSA Cryptosystems

The RSA cryptosystem is based on the difficulty of factoring large integers into its composite primes.

Based on Fermat's little theorem, we know that $a^m \equiv 1 \mod p$ exactly when p-1 divides m. Therefore we recover the identity $a^u \equiv a \mod p$ where u is of the form 1 + (p-1)r. Now given any e such that e and p-1 have no common divisors, there exists a d such that $ed \equiv 1 \mod p - 1$. In other words, u = ed is of the form 1 + (p-1)r. This means that the map

 $a \mapsto a^e \mod p$

followed by

$$a^e \mod p \mapsto (a^e \mod p)^d \mod p \equiv a^{ed} \mod p = a \mod p$$

are inverse maps. This only works for a prime p.

1. Use Magma to find a large prime p and to compute inverse exponentiation pairs e and d. The following functions are of use:

RandomPrime, Random, GCD, XGCD, and InverseMod.

The RSA cryptosystem is based on the fact that for primes p and q and any integer e with no common factors with p-1 and q-1, it is possible to find an d_1 such that

$$ed_1 \equiv 1 \mod (p-1),$$

 $ed_2 \equiv 1 \mod (q-1).$

Using the Chinese remainder theorem, it is possible to then find the unique d such that

$$d = d_1 \mod (p-1)$$
 and $d = d_2 \mod (q-1)$

in the range $1 \leq d < (p-1)(q-1)$. This d has the property that

$$a^{ed} \equiv a \mod n.$$

The send a message securely, the public key (e, n) is used. First we encoding the message as an integer $a \mod n$, then form the ciphertext $a^e \mod n$. The recipient recovers the message using the secret exponent d.

Solution The function call RandomPrime(100) returns a random prime of up to 100 bits. Suppose that the primes

p = 1172991670841347272989353064539,q = 300997517969507552061104346547, are found with this function, and set e = 5. We want to build the inverse exponent d such that $ed \equiv 1 \mod (p-1)$ and $ed \equiv 1 \mod (q-1)$. Note first that GCD(e, p-1) = 1 and GCD(e, q-1) = 1, so that such a d must exist. We first compute each of $d \mod (p-1)$ and $d \mod (q-1)$.

```
> d1 := InverseMod(e,p-1);
> d1;
703795002504808363793611838723
> d2 := InverseMod(e,q-1);
> d2;
240798014375606041648883477237
```

The value of d can now be computed modulo the value LCM(p-1, q-1) — this is sufficient to determine the inverse, rather than the larger value of the product (p-1)(q-1).

We would like to compute the value of d, but the Magma function CRT complains that the moduli p-1 and q-1 have a common factor.

```
> GCD(p-1,q-1);
6
> TrialDivision(p-1);
[ <2, 1>, <3, 3>, <13, 1> ]
[ 1670928306041805232178565619 ]
> TrialDivision(q-1);
[ <2, 1>, <3, 1>, <17, 1>, <5297, 1> ]
```

We can divide q-1 by 6 to remove the common factor, and so compute the Chinese remainder lifting as follows. Note first that the system is consistent — d_1 and d_2 are the same modulo 6 since they are both inverses to $e \mod 6$.

```
> d1 mod 6;
5
> d2 mod 6;
5
```

Since (q-1)/6 is not divisible by 2 or 3, we can proceed with the Chinese remainder lifting with p-1 and (q-1)/6.

```
> d := CRT([d1,d2],[p-1,(q-1) div 6]);
> d;
35306758152215111348997570443072341096420788599987705538575
```

Alternatively we could have computed the inverse exponent d in one step by

```
> d := InverseMod(e,LCM(p-1,q-1));
> d;
35306758152215111348997570443072341096420788599987705538575
```

2. Use your exponents e, d, verify the identities mod p:

 $(a^e)^d \equiv a \mod n, \ (a^d)^e \equiv a \mod n, \text{ and } a^{ed} \equiv a \mod n,$

for various random values of a.

Note that after construction of d, the primes p and q are not needed, but that without knowing the original factorization of n, Fermat's little theorem does not apply, and finding the inverse exponent for e is considered a hard problem.

Solution Now we can verify that e and d are inverses modulo p-1 and modulo q-1, and, moreover, that they determine inverse exponential maps modulo the RSA modulus n = pq.

```
> (e*d) mod (p-1);
1
> (e*d) mod (q-1);
1
> n := p*q;
> m := Random(n);
> c := Modexp(m,e,n);
> m eq Modexp(c,d,n);
true
> m eq Modexp(m,e*d,n);
true
```

We use the RSA cryptosystem in Magma as follows. First begin with encoding ASCII text numerically:

Note that, as with LFSR cryptosystems, RSA encoding uses the information-preserving ASCII bit encoding, and encoding and decoding are true inverses. **Caution:** we note that decoding ciphertext might render an xterm nonfunctional, since the resulting ASCII text might contain escape characters which reset the terminal display.

To encipher, first we must create a key pair:

```
> K, L := RandomKeys(C);
> K;
[ 49338921862830381807760291818994034053,
86398677368792768067556452456311743331 ]
```

This returns a pair of inverse keys K and L. We will consider K to be the public K and L to be the private key.

N.B. The argument to **RSACryptosystem** specifies the number of bits in the RSA modulus. With a value of 128, the modulus is of size 2^{128} , or about 39 decimal digits. Each of the primes is of size approximately 20 decimal digits. This particular example can be easily broken by the factorization:

```
> time Factorization(86398677368792768067556452456311743331);
[ <6046864213681032211, 1>, <14288178850339607921, 1> ]
Time: 3.310
```

3. Use the above factorization to reproduce the private key L (generated but not printed above) for this K.

Solution Given the factorization

86398677368792768067556452456311743331

 $= 6046864213681032211 \cdot 14288178850339607921,$

we can find the inverse to the exponent

```
e = 49338921862830381807760291818994034053.
```

```
> e := 49338921862830381807760291818994034053;
> p := 6046864213681032211;
```

```
> q := 14288178850339607921;
```

```
> d := InverseMod(e,LCM(p-1,q-1));
```

```
> d;
```

285484457605725559400259141876035917

It is now possible to verify as above that (e, n) and (d, n) server as inverse RSA keys.

4. Why is the choice for which key is the public key and which key is the private key arbitrary? Practice encoding, decoding, enciphering, and deciphering with the RSA cryptosystem. Why do the functions **Enciphering** and **Deciphering** return the same values?

Solution Provided that e is chosen as a random number in the range

$$1 \le e \le \mathrm{LCM}(p-1, q-1),$$

which has no common factors with p-1 or q-1, then its inverse is a similarly random value in this range. Therefore after creation, the decision of which key to publish as the public key, and which key to guard as the private key is arbitrary.

N.B. Sometimes a special value, such as 3, 5, 17, 257, or 65537, is chosen as the public exponent. These are each of the form $2^r + 1$, so that the enciphering can be done rapidly using only r squarings and one multiplication. In such a case it is clear that no such "obvious" value is suitable for the private key, and the symmetry of the choice between public and private keys is broken.