Applied Cryptography: Homework 8

(Deadline: 10:00am, 2020/11/18)

Justify your answers with calculations, proofs, and programs.

1. (30 points, question 6.14, page 248 of the textbook)

   A common way to speed up RSA decryption incorporates the Chinese remainder theorem, as follows. Suppose that $d_K(y) = y^d \bmod n$ and $n = pq$. Define $d_p = d \bmod (p-1)$ and $d_q = d \bmod (q-1)$; and let $M_p = q^{-1} \bmod p$ and $M_q = p^{-1} \bmod q$. Then consider the following algorithm:

   Algorithm 6.15: CRT-OPTIMIZED RSA DECRYPTION $(n, d_p, d_q, M_p, M_q, y)$

   $x_p \leftarrow y^{d_p} \bmod p$
   $x_q \leftarrow y^{d_q} \bmod q$
   $x \leftarrow M_p q x_p + M_q p x_q \bmod n$
   return (x)

   Algorithm 6.15 replaces an exponentiation modulo n by modular exponentiations modulo p and q. If p and q are ℓ-bit integers and exponentiation modulo an ℓ-bit integer takes time $c\ell^3$, then the time to perform the required exponentiation(s) is reduced from $c(2\ell)^3$ to $2c\ell^3$, a savings of 75%. The final step, involving the Chinese remainder theorem, requires time $\mathcal{O}(\ell^2)$ if d_p, d_q, M_p, and M_q have been pre-computed.

   (a) Prove that the value x returned by Algorithm 6.15 is, in fact, $y^d \bmod n$.

   (b) Given that $p = 1511, q = 2003$, and $d = 1234577$, compute d_p, d_q, M_p, and M_q.

   (c) Given the above values of p, q, and d, decrypt the ciphertext $y = 152702$ using Algorithm 6.15.

2. (30 points, question 6.21, page 251 of the textbook)

   Write a program to evaluate Jacobi symbols using the four properties presented in Section 6.4. The program should not do any factoring, other than dividing out powers of two. Test your program by computing the following Jacobi symbols:

   $$(\tfrac{610}{987}), (\tfrac{20964}{1987}), (\tfrac{1234567}{11111111})$$

3. (10 points, question 6.26, page 253 of the textbook)

   Using various choices for the bound, B, attempt to factor 262063 and 9420457 using the $p-1$ method. How big does B have to be in each case to be successful?

4. (20 points)

   Implement the Miller-Rabin algorithm. Test whether the following number is a prime:

   $n = 173978897835149618964830869535987635323514892205280914573211561972426967968508524037559789148741712061708378481205017865753096554838058038306206137432977766565647608464326829661627503278711528968113028092931678865263972111666498370475757652374822169740788077289029083381001639360536816949453583901891631791126287195624228878035176617274504146394448620505476647251926632748223972512934907559647678981406615285048 7`

97823803644786231218175243214840513078474972799108333261435785431915660660121001469
47056419459797906018918465410663341843118614746568200081683099400746255691962446351
17397707694130269777094500036249022 4529