MAS433 Cryptography:

Tutorial 4

Public Key Encryption and Digital Signature

19.11.2010

Problem 1. Toy RSA

In a toy RSA encryption scheme, $n = 209$, $e = 7$. Find the value of the private key d. Decrypt the ciphertext $c = 3$.

Solution:

$n = 11 \times 19$

$\varphi(n) = (11-1) \times (19-1) = 180$

Since $e \cdot d \bmod \varphi(n) = 1$, we apply the extended Euclidean algorithm to $(e, \varphi(n))$:

180

7

$5 = 180 - 25 \times 7$

$2 = 7 - 5$

$1 = 5 - 2 \times 2$

$$\therefore \quad 1 = 5 - 2\times 2 = 5 - 2 \times (7-5)$$
$$= -2 \times 7 + 3 \times 5$$
$$= -2 \times 7 + 3 \times (180 - 25 \times 7)$$
$$= 3 \times 180 - 77 \times 7$$

$$\therefore \quad 7^{-1} \bmod 180 \equiv -77 \equiv 103 \pmod{180}$$

$$d = 7^{-1} \pmod{180} = 103$$

//

Problem 2. RSA: Common Modulus
Two users Alice and Bob use RSA public keys with the same modulus n but with different public exponents e_1 and e_2.

(a) Prove that Alice can decrypt messages sent to Bob.

Lecture 14, slide 44. (Alice can factorize n easily) //

(b) Suppose that message padding is not used in RSA encryption. Prove that Eve can decrypt a message sent to Alice and Bob provided that $\gcd(e_1, e_2) = 1$. (Hint: how to find a and b satisfying $a \times e_1 + b \times e_2 = 1$)

Solution:

Since $\gcd(e_1, e_2) = 1$, we can find a, b satisfying $a \cdot e_1 + b \cdot e_2 = 1$ (using the extended Euclidean algorithm)

A message m is sent to Alice and Bob.

Alice receives: $C_1 = m^{e_1} \bmod n$

Bob receives: $C_2 = m^{e_2} \bmod n$

An ~~a~~ attacker can recover the message as follows.

$$
\begin{aligned}
& C_1^a \times C_2^b \bmod n \\
&= (m^{e_1})^a \times (m^{e_2})^b \bmod n \\
&= m^{a \cdot e_1 + b \cdot e_2} \bmod n \\
&= m^1 \bmod n \\
&= m \quad //
\end{aligned}
$$

Problem 3. RSA: $\lambda(n)$

In RSA, d can be computed as $\underline{e{\cdot}d \equiv 1(\mathrm{mod}\lambda(n))}$, where

$$\lambda(n) = \frac{(p-1)(q-1)}{gcd(p-1, q-1)}.$$

$ed = \alpha\lambda(n)+1$

(a) Prove that encryption and decryption are inverse operations.

Let $g = gcd(p-1, q-1)$

Then $p-1 = gp'$

$q-1 = gq'$

$\lambda(n) = gp'q'$

Let the encryption be: $c = m^e \bmod n$.

$c^d \bmod p = m^{ed} \bmod p$

$= m^{\alpha\cdot\lambda(n)+1} \bmod p$ (since $ed \equiv 1 \bmod \lambda(n)$)

$= m^{\alpha\cdot g\cdot p'\cdot q'+1} \bmod p$

$= m^{\alpha(p-1)\cdot q'+1} \bmod p$

$= m$ (1)

Similarly: $c^d \bmod q = m \bmod q$ (2)

From (1), $p \mid c^d - m$ (3)

From (2), $q \mid c^d - m$ (4)

Since p and q are coprime, from (3) and (4): $pq \mid c^d - m$

$\Rightarrow c^d \bmod n = m$ //

(b) Let $n = 209$, $e = 7$. Find the value of the private key d. Decrypt the ciphertext $c = 3$.

Solution:

$n = 11 \times 19$

$\lambda(n) = 90$

Since $e \cdot d \bmod \lambda(n) = 1$, we apply the extended Euclidean algorithm to $(e, \lambda(n))$

90

7

$6 = 90 - 12 \times 7$

$1 = 7 - 6$

$\therefore\ 1 = 7 - 6 = 7 - (90 - 12 \times 7)$

$= -90 + 13 \times 7$

$\therefore\ d = 13$

$m = c^d \bmod n$

$= 3^{13} \bmod 209$

$= 71$ //

Problem 4. RSA: small difference between p and q

The p and q in RSA should be randomly generated, and they are the same size. The difference between p and q should not be small.

(a) Suppose that p and q are 1024-bit prime numbers, but the difference between p and q is small, say, $u = |p - q| < 2^{32}$. How to factorize the product of p and q?

Solution: Let $u = 2v$. Suppose that $p > q$.

$$p - q = 2v$$

$$p^2 - pq = 2pv$$

$$p^2 - 2pv = n$$

$$p^2 - 2pv + v^2 = n + v^2$$

$$(p-v)^2 = n + v^2$$

$$p - v = \sqrt{n+v^2} \quad (p \gg v)$$

$$p = v + \sqrt{n+v^2}$$

Then we try all the possible values of v.

If the value of v is guessed correctly, $\sqrt{n+v^2}$ should be an integer.

(b) Suppose that $u = |p-q| < 20$, and $p \times q =$ 2189284635403183. Find the values of p and q.

Solution: $u = 2v$

$p = v + \sqrt{n+v^2}$

Try $v = 1, 2, 3 \ldots\ldots 9$

When $v = 9$,

$p = 9 + \sqrt{n+v^2}$

$= 9 + 46789792$

$= 46789801$ //

$q = n/p = 46789783$ //

Problem 5.

Dixon's Random Squares Algorithm

Factorize 256961 using Dixon's Random Squares Algorithm. The factor base $\{-1,2,3,5,7,11,13,17,19,23,29,31\}$ may be used.

Solution.

$$m = \lfloor \sqrt{256961} \rfloor = 506$$

$$Q(x_i) = (m + x_i)^2 - n$$

If $x_i = -3$, $Q(-3) = 503^2 \bmod n = (-1) \times 2^4 \times 13 \times 19$

If $x_i = -2$, $Q(-2) = 504^2 \bmod n = (-1) \times 5 \times 19 \times 31$

If $x_i = -1$, $Q(-1) = 505^2 \bmod n = (-1) \times 2^4 \times 11^2$

If $x_i = 1$, $Q(1) = 507^2 \bmod n = 2^3 \times 11$

If $x_i = 5$, $Q(5) = 511^2 \bmod n = 2^6 \times 5 \times 13$

If $x_i = 10$, $Q(10) = 516^2 \bmod n = \cancel{2^6} \times 5 \times 11 \times 13^2$

If $x_i = 13$, $Q(13) = 519^2 \bmod n = 2^4 \times 5^2 \times 31$

For $x_i = -3, -2, 5, 13$, we obtain:

$$(503 \times 504 \times 511 \times 519)^2 \equiv (2^7 \times 5^2 \times 13 \times 19 \times 31)^2 \pmod{n}$$

$$75319^2 \equiv 91105^2 \bmod n$$

$\gcd(75319 + 91105, 256961) = 293$

$\gcd(75319 - 91105, 256961) = 877$

$\Rightarrow 256961 = 293 \times 877$

Problem 6. Toy ElGamal Encryption

In a toy ElGamal encryption scheme, $p = 227$, $g = 2$, and $x = 15$. Decrypt the ciphertext (10, 159).

Solution:

$$m = C_1^{-x} \cdot C_2 \bmod p, \quad \text{where } C_1 = g^k \bmod p, \; C_2 = y^k \cdot m \bmod p$$

$$= (10^{15})^{-1} \times 159 \bmod 227$$

$$10^{15} \bmod 227 = 76$$

$$76^{-1} \bmod 227 = 3$$

$$\therefore m = 159 \times 3 \bmod 227$$

$$= 23 \quad //$$

Problem 7. Index Calculus Algorithm
Let $p = 227$. The element $g = 2$ is a generator of the multiplicative group Z_p^*.

(a) Compute g^{32}, g^{40}, g^{59} and g^{156} modulo p, and factorize them.

Solution:
(a)

$$2^{32} \bmod 227 = 176 = 2^4 \times 11 \quad (1)$$

$$2^{40} \bmod 227 = 110 = 2 \times 5 \times 11 \quad (2)$$

$$2^{59} \bmod 227 = 60 = 2^2 \times 3 \times 5 \quad (3)$$

$$2^{156} \bmod 227 = 28 = 2^2 \times 7 \quad (4)$$

Solution:
(b)

From (1), (2), (3) and (4), we obtain:

$$4 \times \log_2 2 + \log_2 11 \equiv 32 \bmod 226 \quad (5)$$

$$\log_2 2 + \log_2 5 + \log_2 11 \equiv 40 \bmod 226 \quad (6)$$

$$2 \times \log_2 2 + \log_2 3 + \log_2 5 \equiv 59 \bmod 226 \quad (7)$$

$$2 \times \log_2 2 + \log_2 7 \equiv 156 \bmod 226 \quad (8)$$

Solve (5), (6), (7) and (8), we obtain

$$\log_2 3 = 46$$

$$\log_2 5 = 11$$

$$\log_2 7 = 154$$

$$\log_2 11 = 28$$

(b) Find the values of $\log_g 2 \mod p$, $\log_g 3 \mod p$, $\log_g 5 \mod p$, $\log_g 7 \mod p$, and $\log_g 11 \mod p$.

Solution on the previous page.

(c) Suppose that we wish to compute $\log_g 173 \mod p$. Multiply 173 by $g^{177} \mod p$, and factorize the result. What is the value of $\log_g 173 \mod p$?

Solution:

$$173 \times 2^{177} \equiv 168 \equiv 2^3 \times 3 \times 7 \pmod{227}$$

$$\therefore \log_2 173 + 177 \equiv 3 \times \log_2 2 + \log_2 3 + \log_2 7 \pmod{226}$$

$$\log_2 173 \equiv 3 + \log_2 3 + \log_2 7 - 177 \pmod{226}$$

$$\equiv 3 + 46 + 154 - 177 \pmod{226}$$

$$\equiv 26 \pmod{226}$$

//

Problem 8. RSA Signature Scheme

Suppose that hash function is not used in RSA digital signature scheme. The signature is generated as $s = m^d \bmod n$. Given a message m and its signature s, how to modify m without being detected?

Solution:

Let $m' = m + \alpha \cdot n$ for any arbitrary α,

the signature of m' is the same as that of m.

Problem 9. ElGamal Signature Scheme

(a) In the Elgamal signature scheme, why should the signature with $s = 0$ be deleted?

Solution:

In Elgamal Signature Scheme, s is generated as:

$$S = (H(m) - xr) \cdot k^{-1} \bmod p-1$$

If $s = 0$, it means that

$$(H(m) - x \cdot r) \cdot k^{-1} \bmod p-1 = 0 \qquad (1)$$

k is chosen so that $\gcd(k, p-1) = 1$, so $k \neq 0$ (2)

From (1) & (2),

$$H(m) - x \cdot r \bmod p-1 = 0 \qquad (3)$$

If $\gcd(r, p-1) = 1$, (3) can be solved directly;

Otherwise, there are more than one solutions for x, and we need to test those solutions to determine the value of x.

(b) In the Elgamal signature scheme, each per-message secret integer k should be used only once. If the per-message secret integer k is reused, how to attack this digital signature algorithm?

Solution Outline:

$$r = g^k$$

$$S_1 = ((H(m_1) - x \cdot r) \cdot k^{-1} \bmod p-1 \qquad (1)$$

$$S_2 = ((H(m_2) - x \cdot r) \cdot k^{-1} \bmod p-1 \qquad (2)$$

From (1):

$$kS_1 \equiv H(m_1) - xr \pmod{p-1} \qquad (3)$$

From (2):

$$kS_2 \equiv H(m_2) - xr \pmod{p-1} \qquad (4)$$

$(3) \times S_2 - (4) \times S_1$:

$$0 \equiv S_2 (H(m_1) - xr) - S_1 (H(m_2) - xr) \pmod{p-1}$$

(c) In the Elgamal signature scheme, each per-message secret integer k should be randomly generated. If the per-message secret integer k is generated as follows: k_0 is randomly generated, $k_{i+1} = k_i + a$, where a is a known constant. Develop an attack to recover the private key.

Solution Outline:

$$r_0 = g^{k_0} \bmod p$$

$$r_1 = g^{k_0+a} \bmod p = \alpha \cdot r_0 \bmod p \quad (\alpha = g^a)$$

$$s_0 = (H(m_0) - x \cdot r_0) \cdot k_0^{-1} \bmod p-1 \qquad (1)$$

$$s_1 = (H(m_1) - x r_1) \cdot k_1^{-1} \bmod p-1$$

$$= (H(m_1) - \alpha \cdot x \cdot r_0) \cdot (k_0 + a)^{-1} \bmod p-1 \qquad (2)$$

Solve (1) & (2) for x.

(d) In the ElGamal signature scheme, each per-message secret integer k should be randomly generated. If the per-message secret integer k is generated as follows: k_0 is randomly generated, $k_{i+1} = k_i + a$, where a is a large unknown constant. Develop an attack to recover the private key .

Solution Outline:

$$s_0 = (H(m_0) - x r_0)\, k_0^{-1} \bmod p-1 \qquad (1)$$

$$s_1 = (H(m_1) - x r_1)\, k_1^{-1} \bmod p-1 \qquad (2)$$

$$s_2 = (H(m_2) - x r_2)\, k_2^{-1} \bmod p-1 \qquad (3)$$

Solve (1), (2) & (3) for x.

Problem 10. Digital Signature Algorithm (DSA)

(a) In the Digital Signature Algorithm, if the per-message secret integer k is reused, how to attack this digital signature algorithm?

Solution Outline:

In Digital Signature Algorithm, $r = (g^k \bmod p) \bmod q$,

$$s = (H(m) + xr)k^{-1} \bmod q$$

If s is reused,

$$s_1 = (H(m_1) + xr) \cdot k^{-1} \bmod q \qquad (1)$$

$$s_2 = (H(m_2) + xr) \cdot k^{-1} \bmod q \qquad (2)$$

Solve (1) & (2) for x.

(b) In a modified DSA, s is generated as $s = k^{-1}(H(m) - xr) \mod q$. What is the signature verification algorithm for this modified DSA?

Solution:

In DSA, $u_1 = H(m) s^{-1} \mod q$

$u_2 = r \cdot s^{-1} \mod q$

$v = (g^{u_1} y^{u_2} \mod p) \mod q$

If $s = k^{-1} (H(m) - xr) \mod q$, then

$$r = (g^k \mod p) \mod q$$

$$= (g^{s^{-1}(H(m)-xr)} \mod p) \mod q$$

$$= (g^{H(m) \cdot s^{-1}} \cdot y^{-rs^{-1}} \mod p) \mod q$$

$$= (g^{u_1} y^{-u_2} \mod p) \mod q$$

Thus we need to compute v' as

$$v' = (g^{u_1} y^{-u_2} \mod p) \mod q,$$

then check whether $r \stackrel{?}{=} v'$

(c) Let p and q be prime numbers and q is a divisor of $p-1$. Show that for any integer t, if $g = h^{(p-1)/q} \bmod p$, then $g^t \bmod p = g^{t \bmod q} \bmod p$.

Solution:

Let $t = \alpha q + \beta$, where $\beta = t \bmod q$.

$$
\begin{aligned}
& g^t \bmod p \\
&= (h^{(p-1)/q} \bmod p)^t \bmod p \\
&= h^{(\frac{p-1}{q})t} \bmod p \\
&= h^{(\frac{p-1}{q}) \times (\alpha q + \beta)} \bmod p \\
&= h^{(p-1) + (\frac{p-1}{q}) \cdot \beta} \bmod p \\
&= h^{(\frac{p-1}{q}) \times \beta} \bmod p \\
&= g^{\beta} \bmod p
\end{aligned}
$$

$$\therefore \quad g^t \bmod p = (g^{t \bmod q}) \bmod p;$$

Problem 11. Unconditionally secure and computationally secure

Is there unconditionally secure public key cryptosystem? Briefly explain why.

There is no unconditionally secure public key cryptosystem.

In public key cryptosystem, the public key is known to everyone. With unlimitied computing power, an attacker can always recover the private key from the public key.