Total points: 55. Total time: 75 minutes. 6 problems over 6 pages. No book, notes, or calculator

1. [10 points]
a. Are n=247 and e=11 valid numbers for RSA. Explain. If you answer yes, obtain the corresponding d.
b. Are n=247 and e=9 valid numbers for RSA. Explain. If you answer yes, obtain the corresponding d.

Solution

Part a

There are two requirements:

- n must be a product of two primes
- e must be relatively prime to $\phi(n)$ (so that d, which equals e^{-1} mod-n, exists)

First requirement **[1 points]**
n = 247 = 13·19. 13 and 19 are primes. So this holds.

Second requirement **[2 points]**
If n =p·q where p and q are distinct primes, then $\phi(p \cdot q) = (p-1)\cdot(q-1)$
So $\phi(247) = (13-1)\cdot(19-1) = 12 \cdot 18 = 216$

gcd(11, 192) = 1
[11 is prime and does not divide 216 exactly]
So e=11 is valid.
So $d = 11^{-1} \bmod 216$ **[1 points]**

Obtaining d **[4 points]**
We want integers a and b such that $1 = a \cdot 216 + b \cdot 11$ (then b will be d).
We can do trial and error or use Euclid's algorithm, as shown below.
[Below, rows n = −2 and n = −1 are initialization.
$r_n \leftarrow$ remainder (r_{n-2}/r_{n-1});
$q_n \leftarrow$ quotient (r_{n-2}/r_{n-1});
$u_n \leftarrow u_{n-2} - q_n \cdot u_{n-1}$;
$v_n \leftarrow v_{n-2} - q_n \cdot v_{n-1}$;
]

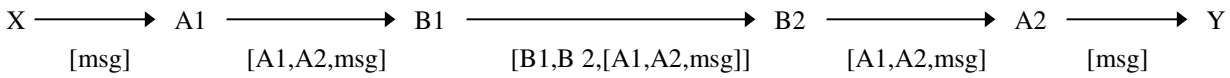
n	q_n	r_n	u_n	v_n
−2		216	1	0
−1		11	0	1
0	19	7	1	−19
1	1	4	−1	20
2	1	3	2	−39
3	1	1	−3	59

From row n=3, we have
$r_n = \gcd(216, 11) = 1$ (which we already knew), and
$1 = (-3)\cdot 216 + (59)\cdot 11$ [= −648 + 649]
So $d = 59 \bmod 216 = 59$.

Part b

$\gcd(9, 192) \neq 1$
[both are divisible by 3]
So e = 9 is not valid. **[2 points]**

2. [6 points]



Every day X talks to Y via nodes A1, B1, B2, A2, as shown above: X sends a msg of 64 octets; A1 attaches a header of "A1,A2"; B1 puts the entire packet in another packet with header "B1,B2"; B2 undoes B1's wrapping; A2 undoes A1's wrapping. Addresses A1, A2, B1, B2 are each 64 bits.

One day, X and Y decide to encrypt their communication with a secret key J (i.e., X and Y share J), and B1 and B2 decide to encrypt their communication with a secret key K (i.e., B1 and B2 share K). X and Y use AES with 128-bit encryption block size and a 256-bit key in CBC mode. B1 and B2 use AES with 64-bit encryption block size and a 128-bit key in CBC mode. Give the size of A1-B1 packet and the size of the B1-B2 packet. Explain your answers briefly.

Solution

CBC requires an IV of the encryption block size:

- For A1-A2, this is 128 bits, which is 16 octets.
- For B1-B2, this is 64 bits, which is 8 octets.

A1, A2, B1, B2 are each 64 bits, which is 8 octets.

- X-A1 pkt = J{msg} **[2 points]**
  pkt size = IV + msg.size
  = 16 + 64 octets = 80 octets
- A1-B1 pkt = [A1,A2, J{msg}] **[2 point]**
  pkt size = 8 + 8 + 80 = 96 octets
- B1-B2 pkt = [B1, B2, K{[A1-A2 pkt]}] **[2 points]**
  pkt size = 8 + 8 + 8 (for IV) + 96
  = 120 octets

[3 points for the A1-B1 pkt and 3 points for the B1-B2 pkt.]

In each part:
−1 point for missing IV
−1 point for incorrect IV size
−2 points for sending key instead of or with IV

---

3. [14 points]
An organization has a PKI (public-key infrastructure) for its principals consisting of a single CA (certification authority) and a single directory server (DS). Certificates have an expiry time of 1 year. CRLs are issued hourly. Answer the following questions. Be brief and precise.

a. Describe the steps taken when a principal A joins the organization.
b. Attacker C steals principal A's private key and A does not realize this. How long after this can C impersonate A.
c. Attacker C steals principal A's private key and A realizes this. Describe the steps A takes.
How long after this can C impersonate A to a principal B.

Solution

Part a. [4 points]

- A interacts with CA offline **[1 points]**
- A generates its public key pair $< \mathbf{pub_A}, \mathbf{pri_A} >$ **[2 points]**
  and gives CA its $\mathbf{pub_A}$
- A gets CA's public key $\mathbf{pub_{CA}}$ **[2 points]**
  and (optionally) certificate for A issued by CA $\mathbf{cert_A}$

Part b. [5 points]

- C can impersonate A to B until A's certificate's expires (1 year at worst) **[3 points]**

Part c. [5 points]

- A interacts offline with CA
- A generates a new public key pair.(as in part a) **[2 points]**
- CA adds A's old certificate's serial number in the next CRL it issues **[2 points]**
- Assume B uses latest CRL. Then C can impersonate A to B until
  A's old certificate's expiry time
  or until next CRL is issued, which is within 1 hour of contacting CA
  whichever is earlier. **[2 points]**

Part a
−2 point for missing CA's public key.
−1 point for missing certificate.

Part b
−2 point for not referring explicitly to expiry time

Part c
−3 point for not using CRL.

4. [10 points]

client A (has J)	**server B** (has password file with entry [A: J])
generate random N_A $C_A \leftarrow$ encrypt N_A with key J send [A, B, conn, C_A] // msg 1	
	receive [A, B, conn, C_A] $N_A \leftarrow$ decrypt C_A with key J $R_A \leftarrow N_A + 1$ generate random N_B $C_B \leftarrow$ encrypt [N_B, R_A] with key J send [B, A, C_B] // msg 2
receive [B, A, C_B] [N_B, R_A] $\leftarrow$ decrypt C_B with key J if $R_A = N_A+1$ then B is authenticated else abort $R_B \leftarrow N_B + 1$ $S_B \leftarrow$ encrypt R_B with key J send [A, B, S_B] // msg 3	
	receive [A, B, S_B] $R_B \leftarrow$ decrypt S_B with key J if $R_B = N_B+1$ then A is authenticated else abort

Client A and server B use the above authentication protocol. J is a key obtained from a password. B handles at most one client at a time. All encryptions are in CBC mode. Answer the following; each part below is independent.

a. Consider an attacker that can **only eavesdrop** (i.e., hear messages in transit but cannot intercept messages or send messages with somebody else's sender id). Can this attacker obtain J by off-line password guessing. If you answer no, explain briefly. If you answer yes, describe the attack.

b. Consider an attacker that can **only spoof A** (i.e., send messages with sender id A and receive messages with destination id A, but not eavesdrop or intercept messages). Can this attacker obtain J by off-line password guessing. If you answer no, explain briefly. If you answer yes, describe the attack.

Solution

Part a. **[5 points]**

Attacker can do off-line password guessing:

- get C_A, C_B (from msgs 1,2) **[2 points]**

- run following password-guessing algorithm **[3 points]**

```
for candidate password cpw do {
   obtain candidate key cJ from cpw;
   cN_A ← decrypt C_A with cJ;
   [cN_B,cR_A] ← decrypt C_B with cJ;
   if cR_A + 1 = cN_A then {cJ is J; exit}
}
```

- Can use C_B, S_B (from msgs 2,3) instead

Part b. **[5 points]**

Attacker can do off-line password guessing:

- generate any C_A
  send [A, B, conn, C_A] // msg 1
  receive [B, A, C_B] // msg 2 **[3 points]**

- run password-guessing algorithm in part a **[2 points]**

---

c. **5. [5 points]**

The same protocol as in problem 4 except that each side includes a random field when encrypting a response.

client A (has J)	**server B** (has password file with entry [A: J])
generate random N_A $C_A \leftarrow$ encrypt N_A with key J send [A, B, conn, C_A] // msg 1	
	receive [A, B, conn, C_A] $N_A \leftarrow$ decrypt C_A with key J $R_A \leftarrow N_A + 1$ generate random N_B generate random Y $C_B \leftarrow$ encrypt [Y, N_B, R_A] with key J send [B, A, C_B] // msg 2
receive [B, A, C_B] [Y, N_B, R_A] $\leftarrow$ decrypt C_B with key J if $R_A = N_A+1$ then B is authenticated else abort $R_B \leftarrow N_B + 1$ generate random Z $S_B \leftarrow$ encrypt [Z, R_B] with key J send [A, B, S_B] // msg 3	
	receive [A, B, S_B] [Z, R_B] $\leftarrow$ decrypt S_B with key J if $R_B = N_B+1$ then A is authenticated else abort

Consider an attacker who can only **eavesdrop**. Can this attacker obtain J by off-line password guessing. If you answer no, explain briefly. If you answer yes, describe the attack.

Solution

Attacker can do off-line password guessing exactly as in problem 4a.
Same grading as in 4a.
[2 points: identifying messages on which to do password-guessing.
3 points: for password guessing algorithm**]**

Presence of random fields does not change anything, because the attacker knows the structure of the messages (because that is not a secret and it cannot be random otherwise A would not know how to extract the fields).

6. [10 points]

client A (has J)	**server B** (has password file with entry [A: J])
generate random C_A send [A, B, conn, C_A] // msg 1	
	receive [A, B, conn, C_A] $R_A \leftarrow$ encrypt (C_A + 1) with key J generate random C_B send [B, A, C_B, R_A] // msg 2
receive [B, A, C_B, R_A] $S_A \leftarrow$ decrypt R_A with key J if $S_A = C_A+1$ then B is authenticated else abort $R_B \leftarrow$ encrypt(C_B + 1) with key J send [A, B, R_B] // msg 3	
	receive [A, B, R_B] $S_B \leftarrow$ decrypt R_B with key J if $S_B = C_B+1$ then A is authenticated else abort
A and B exchange data using J{$C_A \oplus C_B$) as the session key.	

Client A and server B use the above authentication protocol. J is a strong key. B handles at most one client at a time. All encryptions are in CBC mode.
Consider an attacker that can **only eavesdrop** and **spoof A** (i.e., send messages with sender id A and receive messages with destination id A). Can this attacker obtain the contents of the data exchanged in the above session. If you answer no, explain briefly. If you answer yes, describe the attack.

Solution

Attacker can obtain the session key (and hence data) as follows.

Eavesdrops and gets C_A and C_B **[2 points]**

Eavesdrops and records data messages. **[3 points]**

After A disconnects, attacker spoofs A as follows: **[3 points]**
- compute $C_A' \leftarrow (C_A \oplus C_B - 1)$
- send [A,B,conn, C_A']
- receive [B, A, C_B', R_A']

R_A' is the same as the session key **[2 points]**

+1 point for not getting attack but saying offiline password guessing not possible because J is a strong key.
0 points for using offiline password guessing attack.
