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

1. [10 points]
Are n=221 and d=35 valid numbers for RSA. Explain. If you answer yes, obtain the corresponding e.

Solution

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 **[2 points]**
n = 221 = 13·17. 13 and 17 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 $\phi(221) = (13-1) \cdot (17-1) = 12 \cdot 16 = 192$

gcd(35, 192) = 1
[because 35 = 7·5 and 192 = 2^6.3, so they have no factors in common]
So e=35 is valid.
So d = 35^{-1} mod 192 **[2 points]**

Obtaining d **[4 points]**
We want integers a and b such that 1 = a·192 + b·35 (then b will be e).
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		192	1	0
–1		35	0	1
0	5	17	1	–5
1	2	1	–2	11
2	17	0		

From row n=1, we have
r_n = gcd(35, 192) = 1 (which we already knew), and
1 = (–2)·(192) + (11)·35 [= -384 + 385]
So d = 11 mod 192 = 11.

2. [6 points]

X → A1 → B1 → B2 → A2 → Y

[msg] [A1,A2,msg] [B1,B 2,[A1,A2,msg]] [A1,A2,msg] [msg]

Every day X talks to Y via nodes A1, A2, B2, B1, as shown above: X sends a msg of 56 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 32 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 *integrity-protect* their communication with a secret key K (i.e., B1 and B2 share K). Both pairs use DES in CBC mode. Give the size of A1-B1 packet and the size of the B1-B2 packet. Explain your answers briefly.

Solution

DES operates on 8-octet (64-bit) data blocks.
CBC requires an IV of the encryption block size, so this too is 8 octets.
A1, A2, B1, B2 are each 32 bits, which is 4 octets.

- X-A1 pkt = J{msg} **[2 points]**
  pkt size = IV + msg.size
  = 8 + 56 octets = 64 octets
- A1-B1 pkt = [A1,A2, J{msg}] **[1 point]**
  pkt size = 4 + 4 + 64 = 72 octets
- MAC{[A1-A2 pkt]} = IV + CBC residue **[2 points]**
  mac size = 8 + 8 octets
- B1-B2 pkt = [B1, B2, [A1-A2 pkt], MAC{[A1-A2 pkt}] **[1 points]**
  pkt size = 4 + 4 + 72 + 8 + 8
  = 96 octets

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

[−1 point for each missing IV]
[−1 point for missing residue]

---

3. [14 points]
An organization has a PKI (public-key infrastructure) for its employees consisting of a single CA (certification authority) and a single directory server (DS). Answer the following questions. Be brief and precise.

a. Describe the steps taken by a new employee A upon joining the organization.
b. Describe the steps employee A takes to email a message confidentially to an employee B (who may not be online).
c. Describe the steps employee A takes to send a message confidentially to an employee B (who may not be online) such that B can be assured from the contents of the message that it was sent by A (without doing any further interactions).

Solution

Part a. [4 points]

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

Part b. [5 points]

- A contacts DS and gets certificate for B ($\mathbf{cert_B}$) and latest **CRL** **[3 points]**
- A verifies $\mathbf{cert_B}$ using $\mathbf{pub_{CA}}$ **[2 points]**
  encrypts msg using $\mathbf{pub_B}$
  and emails encrypted msg to B

Part c. [5 points]

- A contacts DS and gets certificates for A and B ($\mathbf{cert_A}$, $\mathbf{cert_B}$) and latest **CRL** **[3 points]**
- A verifies $\mathbf{cert_B}$ using $\mathbf{put_{CA}}$ **[2 points]**
  encrypts msg using $\mathbf{pub_B}$
  signs result with its private key $\mathbf{pri_A}$
  and emails encrypted msg and signature to B

Parts b and c.
Roughly zero points for involving CA.
Roughly zero points for doing an authentication with B
−1 point for missing CRL.
−1 point for missing a certificate.

Part c
−1 point for not sending $cert_A$ and CRL to B (without them, B has to interact with DS)

4. [10 points]

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

Attacker can do off-line password guessing: **[5 points]**

- get N_A, S_A (from msgs 1,2)
- run following password-guessing algorithm

```
for candidate password cpw do {
   obtain candidate key cJ from cpw;
   cC ← decrypt N_A with cJ;
   cR ← decrypt S_A with cJ;
   if cC + 1 = cR then {cJ is J; exit}
}
```

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

Part b.

Attacker can do off-line password guessing: **[5 points]**

- generate any N_A
  send [A, B, conn, N_A] // msg 1
  receive [B, A, S_A N_B] // msg 2
- run password-guessing algorithm in part a

---

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

The same protocol as in problem 4 except that J is now a high-quality key, B can handle muliple clients at a time, and the different instances of B do not communicate with each other.

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

Consider an attacker who can only **spoof A**. Can this attacker impersonate A to B. If you answer no, explain briefly. If you answer yes, describe the attack.

Solution

To impersonate A to B, the attacker must deliver a suitable msg 3 to B, i.e., one that has S_B equal to the correct response for N_B **[1 points]**

Because B can handle multiple clients at the same time,
the attacker obtain J{N_B} via a reflection attack:
- request another connection to B with msg 1 set to [A, B, conn, N_B]
- the msg 2 response from this instance of B will have S_A equal to J{N_B}

[4 points]

So the attacker can impersonate A to B.

0 points for password-guessing attack (not possible because J is high-quality key)
0 points if no explanation provided

---

6. [10 points]
Server B, which supports many clients, is attached to the Internet at a well-known (not secret) <TCP port, IP addr> y.
Each client shares a password-dervied key with B. So B has for, each client, an entry consisting of the client id and key.
The clients and server also share Diffie-Hellman parameters g and p (not secret).
B has so many clients that it can decrypt ciphertext encrypted with a client key only if it already knows the client id;
i.e., it is not feasible for B to try all the client keys until it finds one that results in sensible plaintext.

Write down an authentication protocol so that a client A attached at an Internet <TCP port, IP addr> x can connect to B without disclosing its id (i.e., "A") to 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). Cliearly identify the operations done at each side and the messages exchanged.

Solution

1. A attaches to x and requests TCP connnection to y **[3 points]**
2. After connection is established, A initiates DH exchange with B **[3 points]**
3. After DH exchange, A sends its id encrypted with DH key and authentication nonce, etc **[4 points]**

A at x (has g, p and secret key K)	**B at y** (has g, p and a [client id, key] entry for each client)
Part 1 (x establishes TCP connection with y)	
attach to x; request TCP connection to y	
	accept connection request
become open to x	
	become open to y
Part 2 (A and B establish DH key)	
gen a $T_A \leftarrow g^a \bmod p$ send [x, y, T_A] (i.e., send T_A as data on TCP connection)	
	gen b $T_B \leftarrow g^b \bmod p$ send [x, y, T_B] $J_B \leftarrow (T_A)^b \bmod p$ // DH key
$J_A \leftarrow (T_B)^a \bmod p$ // DH key	
Part 3 (A initiates authentication with B using K)	
gen N_A send [x, y, J_A{"A", K{N_A}}]	
	extract "A", K{N_A} using J_B $R_A \leftarrow$ 1 + decrypt K{N_A} using K gen N_B send[y, x, J_B{R_A, K{N_B}}]
extract R_A, K{N_B} using J_A if $R_A = N_A + 1$ then B authenticated $R_B \leftarrow$ 1 + decrypt K{N_B} using K send[x, y, J_B{R_B}]	
	extract R_A, K{N_B} using J_A if $R_B = N_B + 1$ then A authenticated

At most 1 point if part 1 missing. (Without part 1, A and B cannot authenticate without exposing A's id.)
0 points if A or B sends messages with "A" exposed in part 2 (e.g., send [A, B, T_A]).
