Total points: 71. Total time: 75 minutes. 9 problems over 7 pages. No book, notes, or calculator

1. [14 points]
a. Are n=221 and e=3 valid numbers for RSA. Explain. If you answer yes, obtain the corresponding d.
b. Are n=221 and e=5 valid numbers for RSA. Explain. If you answer yes, obtain the corresponding d.

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 **[4 points]**
If n =p·q where p and q are distinct primes, then $\phi(p{\cdot}q) = (p-1){\cdot}(q-1)$
So $\phi(221) = (13-1){\cdot}(17-1) = 12{\cdot}16 = 192$

Part (a)
gcd(3, 192) > 1 [because 3 divides 192 exactly (with quotient 64)].
So e=3 is not valid. **[2 points]**

Part(b)
gcd(5, 192) = 1 [because 5 is prime and does not divide 192 exactly].
So e=5 is valid.
So $d = 5^{-1}$ mod 192 **[2 points]**

Obtaining d **[4 points]**
We want integers a and b such that 1 = a·192 + b·5 (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		192	1	0
–1		5	0	1
0	38	2	1	–38
1	2	1	–2	77
2	2	0	5	–192

From row n=1, we have
r_n = gcd(5, 192) = 1 (which we already knew), and
1 = (–2)·(192) + (77)·5 [= -384 + 385]
So d = 77 mod 192 = 77.

---

2. [6 points]
Sensor X periodically sends a 32-octet measurement to a receiver Y (1 octet = 8 bits). One day the administrator decides that X should protect the measurement data by adding a MAC obtained using DES in CBC mode (in the standard way). How many octets does X now send for each measurement? Explain your answer.

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.
The MAC consists of the IV and the residue (last cipherblock) of DES-CBC encryption.
So X now sends 32-octet measurement plus 8-octet IV plus 8-octet residue: 48 octets total

[Roughly 3 points for the IV part and 3 points for the residue part.]

[3 points if you you don't say anything wrong, you say that a residue and IV is sent and your numbers are off.]

[3 points if you correctly solved for encryption instead of MAC.]

---

3. [10 points]
An organization wants you to implement a PKI (public key infrastructure) for its employees. It has a large number of employees, divided into class-A employees and class-B employees. Class-A employees stay with the organization for several years on an average. When a class-A employee leaves, his/her access privileges must be revoked within an hour. Class-B employess stay with the organization for six or seven days. When a class-B employee leaves, his/her access privileges must be revoked within a day.

Identify the documents of the PKI (e.g., certificates) and their structure (e.g., fields).
Impose constraints, if any, that would improve performance by exploiting the nature of the employees.

Solution

Generic solution: **[5 points]**

- The documents are **certificates** and **CRLs**.
- Each certificate must have the following fields: [3 points]
  - serial number
  - employee id/name
  - public key
  - expiration time
  - CA's signature (computed over all the above fields)
- Each CRL must have the following fields: [2 points]
  - issue time
  - list of serial numbers of revoked unexpired certificates.
  - CA's signature (computed over all the above fields)
- A CRL must be issued every hour.

Optimized solution: **[5 points]**

- For class A employees, set the expiration time to a year (or so) from issue [1 point]
- For class B employees, set the expiration time to seven days from issue [1 point]
- Do not include class B certificates in CRLs,
  since their certificates will expire by the revocation deadline [2 point]
- Add a "employee classification field" to the certificate (set to A or B),
  so users attempting to contact a class B employee can skip the CRL check. [1 point]

[Note: the problem does not ask how one user determines that another user is currently valid.]

---

4. [15 points]

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

c. Consider an attacker that can **only spoof B**. 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:

- get N_A, S_A (from msgs 1,2) or N_A, S_A (from msgs 2,3). **[2 points]**
- run following password-guessing algorithm: **[3 points]**

  for candidate password cpw do {
  obtain candidate key cJ from cpw;
  $cS_A \leftarrow$ encrypt N_A with cJ;
  if $cS_A = S_A$ then {cJ is J; exit}
  }

Part b.

Attacker can do off-line password guessing:

- generate any N_A
send [A, B, conn, N_A] // msg 1
receive [B, A, S_A N_B] // msg 2 **[3 points]**
- run password-guessing algorithm in part a

Part c.

Attacker cannot do off-line password guessing
because it cannot get a <plaintext, ciphertext> pair. **[2 points]**

At most it can get N_A (from receiving msg 1)
and send msg 2 with a known N_B. and garbage for S_A.
But A will not respond with msg 3 because S_A won't match. **[3 points]**

---

5. [5 points]
The same protocol as in problem 4 except that J is now a high-quality key; B still handles at most one client at a time.

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

Consider an attacker who can **eavesdrop, intercept messages, spoof A, and spoof B**. 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 J{N_B} (the encryption of N_A with J). **[2 points]**

Because the attacker does not have J, it must do one of the following:
1. pass N_B to A in msg 2 and get A to return J{N_A} in msg 3, or
2. pass N_B to B in msg 1 and get B to return J{N_A} in msg 2.

Option 1 is not possible because A sends msg 3 only if msg 2 also has J{N_A}, which the attacker cannot do.
Option 2 is not possible because each time B response to msg 1, it chooses a new N_B. **[3 points]**

So the attacker cannot impersonate A to B.

0 points for any of the following answers:

- Attacker impersonates A to B because it can (observe / relay) messages between A and B. Any intermediate node in the path between A and B would be an attacker by this standard.
- Password-guessing attacks: these are not possible because J is a high-quality secret.

---

6. [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 N_A send [A, B, conn, N_A] // msg 1	
	receive [A, B, conn, N_A] $S_A \leftarrow$ encrypt N_A with key J generate random N_B 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 $T_A = N_A$ then B is authenticated else abort $S_B \leftarrow$ encrypt N_B 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 $T_B = N_B$ 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 J{N_B} (the encryption of N_A with J). **[2 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}

[3 points]

So the attacker cannot impersonate A to B.

---

7. [5 points]
Human principal A uses an RSA public-key pair {<*e*, *n*>, <*d*, *n*> for signature purposes. However, A does not remember the public-key pair. Instead A remembers a password *pw* and obtains its public key pair from a directory server D. D's entry for A consists of three pieces of information: *e*, *n*, and *L*, where *L* is *d* encrypted with a key *J* obtained from *pw*. Here is the protocol A uses to obtain its public-key pair and send a signed message to B.

A (has *pw*)	**D** (has <A, *e*, *n*, *L*>)	**B**
send [A, D, gimme] // msg 1		
	receive [A, D, gimme] send [D, A, *e*, *n*, *L*] to A // msg 2	
receive [D, A, *e*, *n*, *L*] compute key *J* from *pw* *d* ← decrypt *L* with key *J*		
send [A, B, msg, signature on msg] // msg 3		
		receive [A, B, msg, signature on msg] ….

Can an attacker who can only eavesdrop (i.e., hear messages but not intercept messages or spoof messages) obtain *d* by off-line password guessing. If you answer no, explain briefly. If you answer yes, describe the attack..

Solution

Yes, the attacker can obtain d by off-line password guessing.

From msg 2, the attacker gets *e*, *n*, *L*. **[2 points]**
From msg 3, the attacker gets *msg* and *sig* (the signature on *msg*).

It executes the following: **[3 points]**

```
for candidate password cpw do {
   cd ← decrypt L with cpw;
   csig ← signature on msg;
   if csig = sig then {<cd, n> is A's private key; exit}
}
```

Note:
Attacker does not need msg 3. It can chose any plaintext *x* and check whether $x^{e \cdot cd}$ mod *n* equals *x*.

---

8. [10 points]
Principals A and B use the following authentication protocol involving a shared high-quality secret key K and Diffie-Hellman parameters g and p (not secret).

A (has K, g, p)	**B** (has K, g, p)
generate random N_A and S_A $T_A \leftarrow g^{SA}$ mod p send [A, B, $K\{N_A\}$, T_A]	
	receive [A, B, $K\{N_A\}$, T_A] $M_A \leftarrow$ decrypt $K\{N_A\}$ using K generate random N_B and S_B $T_B \leftarrow g^{SB}$ mod p send [B, A, M_A, $K\{N_B\}$, T_B] session key $S \leftarrow T_A^{SB}$ mod p
receive [B, A, M_A, $K\{N_B\}$, T_B] if $M_A = N_A$ then B authenticated else abort $M_B \leftarrow$ decrypt $K\{N_B\}$ using K session key $S \leftarrow T_B^{SA}$ mod p send [A, B, M_B]	
	receive [A, B, M_B] if $M_B = N_B$ then A authenticated else abort
<-- ---- A and B use session key S for data exchange ------->	

Consider an attacker C that can eavesdrop, intercept messages, and send messages with another's sender id. Can this attacker decrypt the data exchange between A and B? If you answer no, explain briefly. If you answer yes, describe the attack.

Solution to problem 8

[Note: the problem and solution are taken directly from my F07 exam 1 solution, problem 5, solution attempt 3]

Because K and DH are not used in conjunction, the data exchanged can be decrypted by a man-in-the-middle attack.

	A (has K, g, p)		**Attacker C**		**B** (has K, g, p)
1	generate random N_A and S_A $T_A \leftarrow g^{SA}$ mod p send [A, B, K{N_A}, T_A] //msg 1	→	intercept msg 1 generate random S_C $T_C \leftarrow g^{SC}$ mod p session key $S_{AC} = T_A^{SC}$ mod p forward msg 1 with $T_A \rightarrow T_C$	→	
2		←	intercept msg 2 session key $S_{BC} = T_B^{SC}$ mod p forward msg 2 with $T_B \rightarrow T_C$	←	receive [A, B, K{N_A}, T_C] $M_A \leftarrow$ decrypt K{N_A} using K generate random N_B and S_B $T_B \leftarrow g^{SB}$ mod p send [B, A, M_A, K{N_B}, T_B] //msg 2 session key S $\leftarrow T_A^{SB}$ mod p
3	receive [B, A, M_A, K{N_B}, T_C] $M_A = N_A$ so B is authenticated $M_B \leftarrow$ decrypt K{N_B} using K session key $S_A \leftarrow T_C^{SA}$ mod p send [A, B, M_B] // msg 3	→	no need to modify msg 3	→	
					receive [A, B, M_B] $M_B = N_B$ so A authenticated
	<---- A shares session key S_A with C --> A thinks it shares it with B				<---- B shares session key S_B with C --> B thinks it shares it with A
			C does following for every msg that A sends to B (including the disconnection handshake messages): • intercept the message, • decrypt encrypted fields with S_{AC} and re-encrypt with S_{BC}, • forward modified msg to B C does the same for every msg that B sends to A (with the roles of S_{AC} and S_{BC} interchanged).		

Identifying that the secret key K and the DH parameters are not used in conjunction. **[2 points]**

Identifying / attempting a man-in-the-middle attack. **[3 points]**

Details of the man-in-the-middle attack. **[5 points]**

---

9. [1 point]
Deterministic methods can be used to find out if a 10-digit numbers is prime. One helpful fact is that a number is divisible by 9 if and only if the sum of its digits is divisible by 9; for example, 12834 (and 84312) is divisible by 9 because 1+2+8+3+4 equals 18 which is divisible by 9. Prove this fact.

Solution

Figure it out. It's easy.
