DCU

DUBLIN CITY UNIVERSITY

SEMESTER 1 SOLUTIONS 2016/2017

MODULE: CA4005 - Cryptography and Security Protocols

PROGRAMME(S):

CASE - BSc in Computer Applications (Sft.Eng.)
CPSSD - BSc in ComputationalProblem SolvandSW Dev.
ECSAO - Study Abroad (Engineering and Computing)
ECSA - Study Abroad (Engineering and Computing)

YEAR OF STUDY: 4,O,X

EXAMINERS: Geoffrey Hamilton (Ph:5017)
Prof. David Bustard
Dr. Ian Pitt

TIME ALLOWED: 3 hours

INSTRUCTIONS: Answer all 5 questions. All questions carry equal marks.

PLEASE DO NOT TURN OVER THIS PAGE UNTIL INSTRUCTED TO DO SO

The use of programmable or text storing calculators is expressly forbidden.
Please note that where a candidate answers more than the required number of questions, the examiner will mark all questions attempted and then select the highest scoring ones.

Requirements for this paper (Please mark (X) as appropriate)

□ *Log Tables*
□ *Graph Paper*
□ *Dictionaries*
□ *Statistical Tables*
□ *Thermodynamic Tables*
□ *Actuarial Tables*
□ *MCQ Only - Do not publish*
□ *Attached Answer Sheet*

QUESTION 1 ***[Total marks: 20]***

1(a) [6 Marks]

What are the minimum recommended block size and key size which should be used for a block cipher? What attacks could be mounted against the block cipher if either of these sizes is less than recommended? What are the implications of this for DES?

Solution:
The block size and key size for a block cipher should be at least 128 bits. If the block size is less than recommended, then blocks will be repeated much more often, and attackers will be able to detect patterns in the output which leak secret information. If the key size is less than recommended, then the block cipher will be susceptible to a brute force attack over the smaller keyspace. This means that DES is susceptible to both of these attacks since it has a block size of 64 bits and key size of 56 bits.

1(b) [8 Marks]

Describe the *Feistel structure* which is used in many block ciphers in conjunction with a round function. Show how the effect of the Feistel structure can be reversed using the same round function. Using these answers, describe how encryption and decryption are performed in DES.

Solution:
Using a Feistel structure with round function f and input with left half L_{i-1} and right half R_{i-1}, the output is calculated as follows:

- $L_i = R_{i-1}$
- $R_i = L_{i-1} \oplus f(k_i, R_{i-1})$

The same round function f can be used for decryption:

- $L_{i-1} = R_i \oplus f(k_i, R_{i-1})$
- $R_{i-1} = L_i$

Encryption is performed in DES as follows:

- Initial permutation of bits.
- Split into left and right half.
- 16 rounds of identical operations, depending on round key.
- Inverse initial permutation.

The round transformation is as follows:

- Linear expansion: 32 bits → 48 bits.
- XOR with subkey of 48 bits (key schedule selects 48 bits of key k).
- 8 parallel non-linear S-boxes (6 input bits, 4 output bits).
- Permutation of the 32 bits.

Decryption in DES essentially follows these steps in reverse.

1(c) [6 Marks]

If a decision were taken to increase the security of DES by encrypting using the entire cipher twice, describe an attack which could be mounted against this modified cipher. How much more security would be provided by encrypting three times rather than two?

Solution:
Encrypting using DES twice would not give much more security than DES due to the meet-in-the-middle attack. Encrypting using DES three times would give roughly twice the level of security as DES since one side of a meet-in-the-middle attack would involve two applications of DES.

[End Question 1]

QUESTION 2 *[Total marks: 20]*

2(a) [6 Marks]

A cryptographically secure hash function should be *pre-image resistant* and *collision-free*. Define these properties and why they are important for hash functions.

Solution:
A hash function is pre-image resistant if it is computationally infeasible to recover data from its digest. This is important because the original data may need to be kept secret.

A hash function is weakly collision-free or second pre-image resistant if, given M, it is computationally infeasible to find a different M' such that $H(M) = H(M')$. It is strongly collision-free if it is computationally infeasible to find different messages M and M' such that $H(M) = H(M')$. This is important because being able to find collisions relatively easily allows an attacker to replace one message with another which they have found to have the same digest.

2(b) [7 Marks]

Explain the *birthday paradox* by determining for k people the probability that one pair has a birthday in common. For a n-bit digest, how many messages would we need to hash on average before we find a collision? What would therefore be an appropriate digest size?

Solution:
If there are k people, there are $\frac{k(k-1)}{2}$ pairs.

- The probability that one pair has a common birthday is $\frac{k(k-1)}{2\times 365}$.
- If $k \geq \sqrt{365}$ then this probability is more than half.

In general, if there are n possibilities then on average $\sqrt{n}$ trials are required to find a collision. So for a n-bit digest, we need to try an average of $2^{n/2}$ messages to find two with the same digest.

- For a 64-bit digest, this requires 2^{32} trials (feasible)
- For a 128-bit digest, this requires 2^{64} trials (not feasible)

128-bit is therefore an appropriate digest size.

2(c) [7 Marks]

Describe how hash functions can be used to store passwords in conjunction with a *salt*, and how this makes them less susceptible to attack.

Solution:
Rather than storing passwords unencrypted (which is obviously insecure), we can store instead the digest of passwords. This would still be susceptible to a dictionary attack in which digests are pre-computed and compared for more commonly used passwords. A salt is a random value that is generated for each password, combined with it before applying the hash and stored alongside the digest in the password file. By using a salt, constructing a table of possible digests will be difficult, since there will be many possible keys for each password. An attacker will thus be limited to searching through a table of passwords and computing the key for the salt that has been used.

[End Question 2]

QUESTION 3 ***[Total marks: 20]***

Consider a toy RSA example in which the public key is $(N = 33, e = 17)$.
3(a) [6 Marks]

Determine the value of the private key.

Solution:
The private exponent $d = e^{-1} \pmod{\phi(N)}$ i.e. $17^{-1} \pmod{20}$. This can be calculated using the extended Euclidean GCD algorithm:

$$\begin{array}{rcl} 20 & = & 17 \times 1 + 3 \\ 17 & = & 5 \times 3 + 2 \\ 3 & = & (1 \times 2) + 1 \end{array}$$

So:

$$\begin{array}{rcl} 3 & = & 20 - (17 \times 1) \\ 2 & = & 17 - (5 \times 3) = 17 - (5 \times 20) + (5 \times 17) = (6 \times 17) - (5 \times 20) \\ 1 & = & 3 - (1 \times 2) = 20 - (17 \times 1) - (6 \times 17) + (5 \times 20) = (6 \times 20) - (7 \times 17) \end{array}$$

So $17^{-1} \pmod{20} = -7 = 13 \pmod{20}$

The private key is therefore $(N = 33, d = 13)$.

3(b) [7 Marks]

Describe how encryption is done in RSA. Give an efficient algorithm which can be used to implement this encryption, and use this algorithm to encrypt the message 27.

Solution:
Encryption is RSA is done by calculating $c = m^e \pmod{N}$. An efficient algorithm for this modular exponentiation is the square and multiply algorithm; this can be computed bit by bit left-to-right or right-to-left. The left-to-right variant for computing $m^e \pmod{N}$ where e has n bits $e_{n-1} \ldots e_0$ is as follows:

```
y = 1
for i = n-1 downto 0 do
   y = (y*y) mod N
   if $e_i$ = 1 then
      y = (y*m) mod N
   end
end
```

To encrypt 27, we need to compute $27^{17} \pmod{33}$. Using the described algorithm, this is computed as follows:

i	x_i	y
4	1	$1 \times 1 \times 27 \pmod{33} = 27$
3	0	$27 \times 27 \pmod{33} = 3$
2	0	$3 \times 3 \pmod{33} = 9$
1	0	$9 \times 9 \pmod{33} = 15$
0	1	$15 \times 15 \times 27 \pmod{33} = 3$

So the encrypted value is 3.

3(c) [7 Marks]

Describe how decryption is done in RSA. Describe a technique which can be used to implement this decryption more efficiently using the prime factors of the modulus, and use this technique to decrypt the ciphertext generated above.

Solution:
We want to calculate $c^d \pmod{pq}$ and can calculate this more efficiently using $c^d \pmod{p}$ and $c^d \pmod{q}$ and the Chinese Remainder Theorem.

To calculate $3^{13} \pmod{33}$, we calculate $3^{13} \pmod{3}$ and $3^{13} \pmod{11}$ and combine using the Chinese Remainder Theorem.

$3^{13} \pmod{3} = 0$ and $3^{13} \pmod{11} = 3^3 \pmod{11} = 5$, so $3^{13} \pmod{33} = 27$

So the decrypted value is 27.

[End Question 3]

QUESTION 4 ***[Total marks: 20]***

4(a) [8 Marks]

Describe the Needham-Schroeder Public-Key (NSPK) protocol and Lowe's attack on the protocol.

Solution:
The objective of NPSK is for two entities A and B to mutually authenticate each other using their public keys K_A^+ and K_B^+.

The steps are as follows:

1. $A \rightarrow B : \{N_A, A\}_{K_B^+}$
2. $B \rightarrow A : \{N_A, N_B\}_{K_A^+}$

3. $A \to B : \{N_B\}_{K_B^+}$

Lowe's attack involves an intruder I interleaving two runs of the protocol and transferring authentication i.e. it can impersonate A to B:

$$
\begin{array}{llllll}
1. & A \to I : & \{N_A, A\}_{K_I^+} & & & \\
 & & & 1'. & I \to B : & \{N_A, A\}_{K_B^+} \\
 & & & 2'. & B \to I : & \{N_A, N_B\}_{K_A^+} \\
2. & I \to A : & \{N_A, N_B\}_{K_A^+} & & & \\
3. & A \to I : & \{N_B\}_{K_I^+} & & & \\
 & & & 3'. & I \to B : & \{N_B\}_{K_B^+}
\end{array}
$$

At the end of the attack, B believes it is communicating with A and that N_A and N_B are known only to A and B, but is in fact communicating with I who also knows N_A and N_B.

4(b) [6 Marks]

Describe the Diffie-Hellman (DH) key agreement protocol and explain why it is subject to a man-in-the-middle attack.

Solution:
Given a prime modulus p and a generator g, Diffie-Hellman key exchange works as follows:

1. A chooses a random x such that $1 < x < p - 1$.
2. $A \to B : g^x \pmod{p}$
3. B chooses a random y such that $1 < y < p - 1$.
4. $B \to A : g^y \pmod{p}$
5. A computes $K = (g^y)^x \pmod{p}$.
6. B computes $K = (g^x)^y \pmod{p}$.
7. A and B now share the secret K.

The man-in-the-middle attack with attacker I is as follows:

1. A chooses a random x such that $1 < x < p - 1$.
2. $A \to I : g^x \pmod{p}$
3. I chooses a random z such that $1 < z < p - 1$.
4. $I \to B : g^z \pmod{p}$
5. B chooses a random y such that $1 < y < p - 1$.
6. $B \to I : g^y \pmod{p}$
7. $I \to A : g^z \pmod{p}$
8. A computes $K = (g^z)^x \pmod{p}$.
9. B computes $K' = (g^z)^y \pmod{p}$.
10. I shares K with A and K' with B.

4(c) [6 Marks]

Explain how the DH protocol can be "embedded" into the NSPK protocol and give the protocol steps of the resulting modified protocol that achieves entity-authentication and key-exchange. Is the resulting protocol subject to Lowe's attack or a man-in-the-middle attack? Justify your answers.

Solution:
The DH protocol can be "embedded" into the NSPK protocol as follows:

1. $A \to B : \{g^x \pmod p, A\}_{K_B^+}$
2. $B \to A : \{g^x \pmod p, g^y \pmod p\}_{K_A^+}$
3. $A \to B : \{g^y \pmod p\}_{K_B^+}$

The resulting protocol is subject to Lowe's attack but not to a man-in-the-middle attack.

[End Question 4]

QUESTION 5 *[Total marks: 20]*

5(a) [7 Marks]

Describe SSL with particular reference to the protocols involved, how authenticity, confidentiality and integrity are achieved and the level of security provided.

Solution:
This is just bookwork.

5(b) [7 Marks]

Describe IPSec with particular reference to the protocols involved, how authenticity, confidentiality and integrity are achieved and the level of security provided.

Solution:
This is just bookwork.

5(c) [6 Marks]

Compare and contrast SSL and IPSEC with respect to the above.

Solution:
IPSec:

- Lives at the network layer (part of the OS)
- OS must be aware, but not apps
- Encryption, integrity, authentication, etc.
- Is overly complex, has some security "issues"
- Often used in VPNs

SSL:

- Lives at socket layer (part of user space)
- Apps must be aware, but not OS
- Encryption, integrity, authentication, etc.
- Relatively simple and elegant specification
- Built into web early on (Netscape)

SSL session is like IKE phase 1; SSL connection is like IKE phase 2.

[End Question 5]

[END OF EXAM]