The University of Newcastle
School of Electrical Engineering and Computer Science

COMP3260/COMP6360 Data Security
Week 11 Workshop – 22 & 23 May 2019

Sample Exam Questions

NOTE: This are just sample exam questions, to give you an idea about type of questions and what kind of solutions you need to provide; there will be only 8 questions in the final exam and some/all of them will be DIFFERENT. It is not enough to study these solutions; you need to study all the lecture notes and questions from tutorials, assignments and tests.

1. Give definitions of perfect secrecy, unconditional security and computational security.

Solution

Perfect secrecy is achieved when nothing is learned about the plaintext no matter how much ciphertext is intercepted. For example, if an attacker knows that the plaintext may be message A or message B with a probability of 0.5, then after intercepting the ciphertext, the attacker still only knows that the probability of the plaintext being message A or message B is still 0.5. This occurs when any ciphertext can be obtained from any plaintext using some key.

Unconditional security is achieved when a cipher cannot be broken even with unlimited computational power because there is insufficient information required to uniquely determine the plaintext. This occurs when the Unicity distance never approaches zero, regardless of how much ciphertext is intercepted.

Computational security is achieved when a cipher cannot be broken with limited computing resources. The amount of computing resources necessary to consider something computationally secure is usually determined by the value of the message - if the cost of breaking the cipher is greater than the value of the message, then a cipher may be considered computationally secure.

2. For each of the following ciphers, state if they achieve perfect secrecy, unconditional security, computational security, or none of the above. Justify your answer.

a. One-time pad
b. Homophonic cipher where each homophone appears in the ciphertext at most once
c. Higher order homophonic cipher
d. Caesar cipher
e. DES
f. AES
g. RSA

3. Suppose that M is a 4-digit integer enciphered digit by digit, using a circular Caesar-type substitution cipher with key K, $0 \leq K \leq 9$, and that all possible 4-digit integers are equally likely. For example, if the plaintext M=1234 is enciphered with key K=7, then the ciphertext is C = 8901 and if the plaintext M'=0098 is enciphered with the same key, the corresponding ciphertext is C'=7765. How much ciphertext is needed to break this cipher? Explain your answer.

Solution
The cipher cannot be broken, as there is no redundancy in the plaintext – each possible plaintext is equally likely, so there is no way of determining the plaintext through cryptanalysis. The unicity distance is infinite, as redundancy is 0.

4. The Playfair cipher uses a 5 × 5 matrix of 25 letters as a key (letter J is not used), and enciphers a block of two letters at the time. Find the unicity distance for the Playfair cipher.

Solution
There are 25! Possible ways of arranging the matrix which acts as a key. The first position in the matrix has 25 possible letters that can take that position, the second position in the matrix has 24 possible letters that can take that position (because the first position has already been chosen), and so on, which gives us 25!

We then estimate the unicity distance as the entropy of the key over the redundancy of the plaintext.

$$U = \frac{H(K)}{D}$$
$$= \frac{\log_2 25!}{3.2}$$

5. What is Kerckhoff's principle?

Kerckhoff's principle may be stated as follows: "A cryptosystem should be secure even if everything about the system, except the key, is public knowledge".

This stands in opposition to "security though obscurity" and steganography approaches.

The advantage of following Kerckhoff's principle is that the system itself does not need to be kept secret, only the keys themselves. This makes it easier to have many users using the system, as there is not problem with the details of the system being leaked – the only secret that must be kept is the value of the key.

Another advantage of a non-secret system is that an open system may be analysed and commented on by more people than a secret system – this may lead to flaws not discovered by the original author being discovered and fixed by others, giving a system which is stronger than the original design.

6. What is the difference between a stream and a block cipher?

Solution
Stream ciphers convert plaintext into ciphertext one bit/byte at a time. Examples are Vigenere ciphers and Rotor machines. They are fast to implement, but only use confusion, and provide no diffusion.

Block ciphers convert plaintext into ciphertext one block at a time. 1 block = 128 bits in current block ciphers, the larger the block the more diffusion can be applied, at the cost of computation time. An example is the Playfair cipher, which has a block size of 2 characters.

7. S-boxes are commonly used in symmetric encryption systems to provide substitution and non-linearity. Explain in detail how S-boxes are designed in the following cryptosystems.

a. ***AES***

b. ***DES***

Solution
In AES, the S-Box is used as part of the substitute bytes step which occurs at the start of the encryption round, and the inverse S-Box is used as part of the inverse sub-bytes step which happens second in the decryption round. There is one S-Box and one inverse S-Box, unlike DES which has 8 S-Boxes. The values in the S-Box are chosen such that there is a low correlation between input bits and output bits. The values are generated using a transformation of the values in $GF(2^8)$, with multiplicative inverse providing most of the non-linearity.

In DES, the S-Box is used as part of the round function (remember that DES is a Feistel cipher). The round function F works as follows: the round input is expanded, then XORed with the round key. The output from this is passed through the S-Box, then permuted. There are 8 S-Boxes, and each S-Box takes a 6-bit input and returns a 4-bit output. The reasoning behind the choices for the values in the S-Box has not been published.

8. For each of the following elements of DES, indicate the comparable element in AES or explain why it is not needed in AES.

a. XOR with subkey

b. XOR of the output of F-function with the left half of data block

c. F-function

d. Permutation P

e. Swapping of halves of the data block

Solution
a) Add Round Key Step
b) The Mix Columns Step
c) The Substitute Bytes Step
d) The Shift Rows step
e) Not needed because Mix Columns and Shift Rows operate on the entire block, and when combined should cause every byte to alter every other byte.

9. With the aid of diagram describe the following two modes of operation of DES: Cipher Feedback Mode and Output Feedback Mode.

Solution

CFB – Input is processed s bits at a time. Preceding ciphertext is used as input to the encryption algorithm to produce pseudorandom output, which is XORed with plaintext to produce next unit of ciphertext.

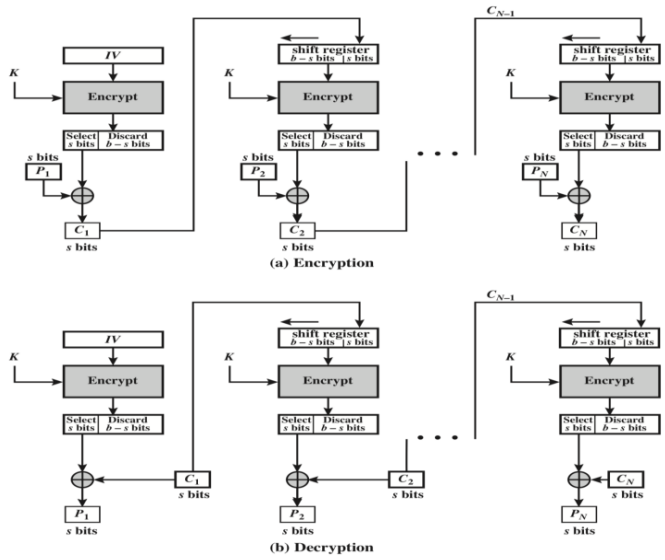


Figure 6.5 *s*-bit Cipher Feedback (CFB) Mode

OFB – Similar to CFB, except that the input to the encryption algorithm is the preceding encryption output, and full blocks are used.

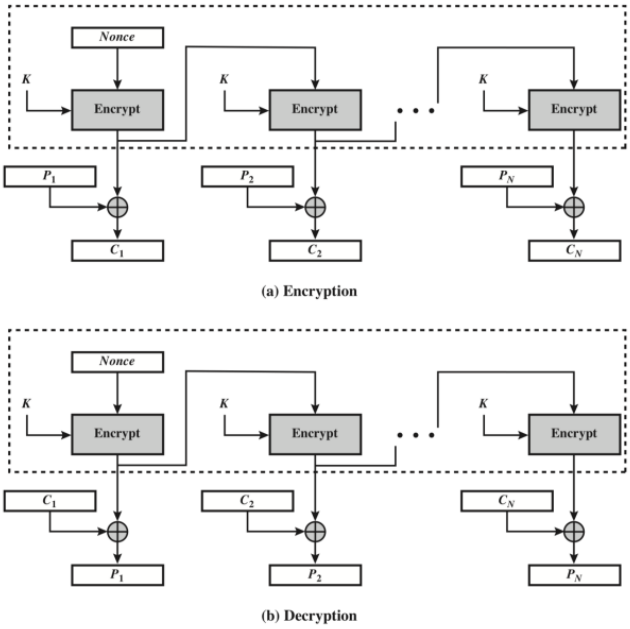


Figure 6.6 Output Feedback (OFB) Mode

10. What is Double DES and how is it vulnerable to Meet-in-the-Middle Attack?

Solution

Double DES encrypts with two keys K1 and K2 such that $C = E_{K2}(E_{K1}(P)))$ and $P = E_{K1}(E_{K2}(C)))$.

To explain Meet-in-the-Middle attack, consider X such that $X = E_{K1}(P) = D_{K2}(C)$.

Given a known plaintext-ciphertext pair (P, C), we can encrypt P with all possible keys K1, and store the results X in a table. We can then decrypt C with all possible keys K2, and compare the decryption of C with the stored X. If there is a match, then we have found a candidate combination of K1 and K2, which can be verified by testing another plaintext-ciphertext pair.

The Meet-in-the-Middle attack has an effort of 2^{56}, which is not much more than an attack on standard DES which has an effort of 2^{55}. We have doubled the amount of computation power required to encrypt/decrypt without substantially increasing the effort of attacking the system.

11. Consider the RSA scheme.

a. If the public key is *(e, n) = (3, 33),* encipher the plaintext *M = 7*. Break the cipher by finding p, q and d. Decipher the ciphertext *C = 2*. (You don't need a calculator – use fast exponentiation!)

b. Prove that $M^{ed} \bmod n = M$ for all values of M, including those where gcd $(M, n) \neq 1$.

Solution

a.

$$\begin{aligned} C &= 7^3 \bmod 33 \\ &= 13 \end{aligned}$$

$p = 3, q = 11, \phi(n) = 20, 3d \bmod 20 = 1, d = 7$

$$\begin{aligned} M &= 2^7 \bmod 33 \\ &= 29 \end{aligned}$$

b.

Recall that for the RSA algorithm, the public key is a pair of numbers (e, n) and the and the private key is a pair of numbers (d, n) where $n = p \cdot q$ for some distinct prime numbers p and q. A message M, is an integer between 0 and $n - 1$. We will start by taking the following as true (as shown in lectures):

$$\begin{aligned} \phi(n) &= (p-1)(q-1) \\ \gcd\big(d, \phi(n)\big) &= 1 \\ (e \cdot d) \bmod \big(\phi(n)\big) &= 1 \\ E(M) &= M^e \bmod n \\ D(M) &= M^d \bmod n \\ E\big(D(M)\big) &= M^{e \cdot d} \bmod n \\ D(E(M)) &= M^{e \cdot d} \bmod n \end{aligned}$$

From this point, to prove that the RSA system works correctly, we need to prove that $E\big(D(M)\big) = M$ and $D\big(E(M)\big) = M$. This is equivalent to showing $M^{e \cdot d} \bmod n = M$.

We will approach this by first showing

$$\begin{aligned} M^{e \cdot d} \bmod p &= M \bmod p \\ M^{e \cdot d} \bmod q &= M \bmod q \end{aligned}$$

And since $\gcd(p, q) = 1$, then by the Chinese Remainder Theorem we know

$$\begin{aligned} M^{e \cdot d} \bmod (p \cdot q) &= M \bmod (p.q) \\ M^{e \cdot d} \bmod (n) &= M \bmod (n) \\ &= M \end{aligned}$$

Since M is an integer between 0 and $n - 1$.

To prove $M^{e \cdot d} \bmod p = M \bmod p$, we need to work through two cases. Case 1 is when $\gcd(M, p) = 1$, and Case 2 is when Case 2 is when $\gcd(M, p) \neq 1$.

For Case 1: If $\gcd(M, p) = 1$, then $M^{\phi(p)} \bmod p = 1$ by Euler's generalisation of Fermat's little theorem. Observe that $(e \cdot d) \bmod \big(\phi(n)\big) = 1 \Rightarrow (e \cdot d) = k\phi(n) + 1$. Then

$$\begin{aligned} M^{e \cdot d} \bmod p &= M^{k\phi(n)+1} \bmod p \\ &= M^{k((p-1)(q-1))+1} \bmod p \\ &= \big(M \cdot M^{k(p-1)(q-1)}\big) \bmod p \\ &= \Big(M \cdot \big(M^{(p-1)}\big)^{k(q-1)}\Big) \bmod p \\ &= \Big(M \cdot \big(M^{(p-1)} \bmod p\big)^{k(q-1)}\Big) \bmod p \\ &= \Big(M \cdot \big(M^{\phi(p)} \bmod p\big)^{k(q-1)}\Big) \bmod p \\ &= \big(M \cdot (1)^{k(q-1)}\big) \bmod p \\ &= (M) \bmod p \end{aligned}$$

Which is what we wanted to show.

For Case 2: If $\gcd(M, p) \neq 1$, then $M = (k \cdot p)$ (i.e. M is a multiple of p, since p is prime). Thus, we know that $M \bmod p = 0$, and

$$\begin{aligned} M^{e \cdot d} \bmod p &= (M \bmod p)^{e \cdot d} \bmod p \\ &= (0)^{e \cdot d} \bmod p \\ &= 0 \\ &= M \bmod p \text{ (because } M \bmod p = 0) \end{aligned}$$

Which is what we wanted to show.

This shows that $M^{e \cdot d} \bmod p = M \bmod p$. To show $M^{e \cdot d} \bmod q = M \bmod q$, we apply the same argument, but replace p with q. We can then apply the Chinese Remainder Theorem as described above to show $M^{e \cdot d} \bmod n = M$, and thus $E\big(D(M)\big) = M$ and $D\big(E(M)\big) = M$, proving that RSA works correctly.

12. Explain how a public-key cryptosystem can provide both privacy and authenticity.

Solution

Privacy can be achieved by encrypting with the receiver's public key, only the receiver can decrypt, as only the receiver has their private key.

Authenticity can be achieved by encrypting with the sender's private key, only the sender could have produced the message as only they have their private key, and anyone can verify the message by decrypting it with the sender's public key.

13. Outline Diffie-Hellman key exchange scheme and show how it can be used for 3 or more parties.

Diffie-Hellman key exchange is based on discrete logarithms. Users choose a global q and α generate private keys, calculate public keys based on the private keys, and send

each other their public keys in plaintext. These public keys enable the users to generate a shared key K.

With three parties, the scheme works as follows:

Alice, Bob and Carol want to obtain a shared key $K = \alpha^{xyz} \bmod q$. They start by choosing a global q and α.

Alice chooses key x, and sends Bob $X = \alpha^x \bmod q$.
Bob chooses key y, and sends Carol $Y = \alpha^y \bmod q$.
Carol chooses key z, and sends Alice $Z = \alpha^z \bmod q$.
If there were more than three parties, Carol would send Z to the fourth member instead of Alice, and so on until each of the members has sent their public key to the next member along.

After the initial round of messages, Alice sends Bob $Z' = Z^x \bmod q$.
Bob sends Carol $X' = X^y \bmod q$.
Carol sends Alice $Y' = Y^z \bmod q$.
If there were more than three parties, Carol would be sending to the fourth member, rather than Alice, similar to the initial round of messages.

Now Alice can compute $K = Y'^x \bmod q$.
Bob can compute $K = Z'^y \bmod q$.
Carol can compute $K = X'^z \bmod q$.

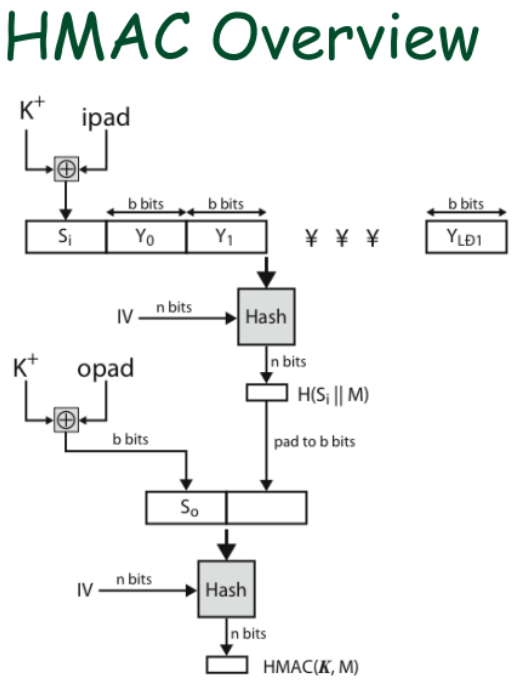
And $K = \alpha^{xyz} \bmod q$.

14. What is a one-way hash function? What is a difference between a one-way hash function and a message authentication code (MAC)?

Solution
A message authentication code (MAC) is used to verify that the message has not been altered, that it comes from the sender, and that the sequence of the message is correct. A message authentication code is generated by a secret key shared by sender and receiver. The message authentication code may be produced by using a one-way hash function, but it could also be produced using an encryption function.

A one-way hash function takes a variable-size message M and produces a fixed-sized output H(M). Because it is not a one-to-one function, it is not reversable, and multiple inputs can produce the same output. Hash functions are faster than encryption, due to them not needing to be reversable.

15. With the aid of diagrams describe in detail HMAC.



1. Create K+ by appending zeros to K
2. XOR K+ with ipad, producing S_i
3. Append M to S_i
4. Apply Hash function H to the output of step 3
5. XOR K+ with opad to produce S_o
6. Prepend S_o to the output of step 4
7. Apply H to the output of step 6

16. What are the main issues that digital signatures address?

Solution
Digital signatures address the issue of trust between two communicating parties. Digital signatures enable a third party to verify the sender of a message in the case of a dispute, preventing forgery and repudiation by the sender.

17. What are the differences between a direct digital signature system and an arbitrated digital signature system?

Solution
A direct digital signature system operates directly between the two communicating parties, whereas an arbitrated digital signature system sends communications though a trusted third party. An arbitrated digital signature system is able to operate using symmetric key encryption system, but a direct digital signature system requires some kind of public key system to be operating.