جامعة بيرزيت
. BIRZEIT UNIVERSITY

Faculty of Engineering and Technology
Electrical and Computer Engineering Department

ENCS4320, Applied Cryptography

Midterm Exam (V2)

Tuesady, 3/01/2023

Name:________________________________ **ID**:_________________________

1. (**2 pts**) For the following encryption scheme, state whether the scheme is perfectly secret. Justify your answer.
The message space is $\mathcal{M} = \{m \in \{0,1\}^{\ell} \mid$ the last bit of m is 0 $\}$. Gen chooses a uniform key from $\{0,1\}^{\ell-1}$. $\mathrm{Enc}_k(m)$ returns ciphertext $m \oplus (k \parallel 0)$, and $\mathrm{Dec}_k(c)$ returns $c \oplus (k \parallel 0)$.

One can prove that this is perfectly secret by analogy with the onetime pad.

Essentially the final bit of the message is being ignored here, since it is always 0.)

2. (**3 pts**) In a Feistel cipher, how does the encryption in one round look like? Can any function be used in this construction? How does decryption work?

Let us describe one round of a Feistel cipher which takes m and produces $R_k(m)$. Here, k is the round key.

- Split the plaintext m into two halves (L_0, R_0)
- $L_1 = R_0$
- $R_1 = L_0 \oplus f_k(R_0)$

- Then, $R_k(m)$ is (L_1, R_1).

The function $f_k(x)$ is referred to as the round function. It can be any PRF function (taking the appropriate amount of input bits, and producing the same number of output bits).

To obtain $m = (L_0, R_0)$ from $R_k(m) = (L_1, R_1)$, we set

$R_0 = L_1$ and then compute

$L_0 = R_1 \oplus f_k(R_0)$

3. (**5 pts**) Consider a block cipher with 5-bit block size and 5-bit key size such that

$$E_k(b_1b_2b_3b_4b_5) = (b_1b_2b_3b_4b_5)\oplus k$$

Encrypt $m = (010101010101010)_2$ using k=$(10001)_2$ and **CBC**-mode (IV=$(10011)_2$).

Solution. $m = m_1m_2m_3$ with $m_1 = 01010, m_2 = 10101, m_3 = 01010.$

$c_0 = \mathbf{10011}$

$c_1 = E_k(m_1 \oplus c_0) = E_k(01010 \oplus 10011) = E_k(11001) = 11001 \oplus 10001 = \mathbf{01000}$

$c_2 = E_k(m_2 \oplus c_1) = E_k(10101 \oplus 01000) = E_k(11101) = 11101 \oplus 10001 = \mathbf{01100}$

$c_3 = E_k(m_3 \oplus c_2) = E_k(01010 \oplus 01100) = E_k(00110) = 00110 \oplus 10001 = \mathbf{10111}$

Hence, the ciphertext is $c = c_1c_2c_3 = (10011\ 01000\ 01010\ 10001\ )$.

4. (**3 pts**) What is the block size of AES? What is the key size? How is it possible that AES uses less rounds than DES?

The block size of AES is 128 bits. Its key size is 128/192/256 bits. It consists of 10/12/14 rounds.

Unlike DES, AES is not a Feistel network. While for a Feistel network, each round only encrypts half of the bits, all bits are being encrypted during each round of AES. That's one indication why AES requires less rounds than DES.

5. (**4 pts**) What is the effect of a single-bit error in the ciphertext when using the CBC-mode of operation?

Solution: Say a message $m_1, m_2, \ldots$ is encrypted to give a ciphertext $c_0, c_1, c_2, \ldots$, and then a single bit is flipped somewhere in the ciphertext. We look at the effect of decrypting the resulting (modified) ciphertext using each of the stated modes to obtain a message $m_1', m_2', \ldots$

CBC mode. Say a bit is flipped in c_i to give modified block c_i'. When decrypting, m_i is computed as $m_i = F_k^{-1}(c_i') \oplus c_{i-1}$ and $F_k^{-1}(c_i')$ will, in general, be completely unrelated to $F_k^{-1}(c_i)$.

Thus,

m_i' will have no relation to m_i.

The next message block, m_{i+1}', is computed as m_{i+1}'=$F_k^{-1}(c_{i+1})\oplus c_i'$ and so

m_{i+1}' will be equal to m_{i+1}' but with a single bit flipped. The rest of the message blocks will be unchanged.

6. (**4 pts**) What is the effect of a dropped ciphertext block (e.g., if the transmitted ciphertext $c_1, c_2, c_3, \ldots$ is received as $c_1, c_3, \ldots$ when using the CBC-mode of operation?

Say a message $m_1, m_2, \ldots$ is encrypted to give a ciphertext $c_0, c_1, c_2, \ldots$, and then a single block is dropped. We look at the effect of decrypting the resulting (modified) ciphertext using each of the stated modes to obtain a message $m'_1, m'_2, \ldots$

***CBC mode**. Say block c_i is dropped. The result of decrypting the modified ciphertext is $m_1, \ldots, m_{i-1}, m'_{i+1}, m_{i+2}, \ldots$. I.e., message blocks before position i are recovered correctly, message block i is lost entirely, message block $i+1$ is garbled, and message blocks after position $i+1$ are recovered correctly (though shifted over by one block).*

7. (**4 pts**) Consider a variant of CBC-mode encryption where the sender simply increments the IV by 1 each time a message is encrypted (rather than choosing $IV = r$ at random each time). Show that the resulting scheme is *not* CPA-secure.

Solution: Define $\mathcal{A}$ as follows:

(a) Query the encryption oracle with the message $m = 0^{n-1} \parallel 1$. Receive in return a ciphertext $\langle IV, c \rangle$.

(b) If IV is odd (i.e., has low-order bit 1), then output a random bit.

(c) If IV is even, then output a pair of messages m_0, m_1 where $m_0 = 0^n$and m_1 is any other message. Receive in return a challenge ciphertext Query the encryption oracle with the message $m = 0^{n-1} \parallel 1$. Receive in return a ciphertext $\langle IV{+}1, c' \rangle$.

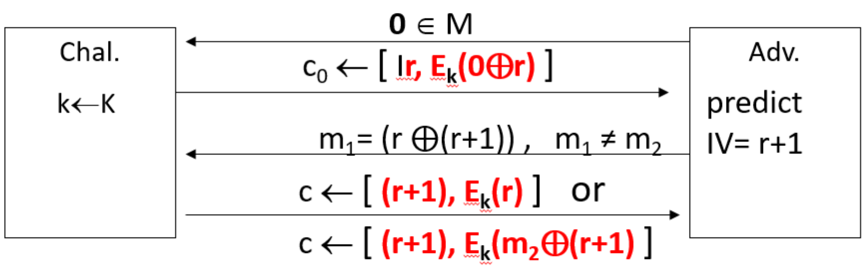
(d) If $c' = c$ output 0; else output 1.

We analyze the success probability of $\mathcal{A}$. If IV is odd, then $\mathcal{A}$ succeeds exactly half the time. For any even, though, it holds that $IV + 1 = IV \oplus (0^{n-1} \parallel 1)$. Thus, for any even IV we have

$$c = F_k(IV \oplus m) = F_k(IV \oplus 0^{n-1}1 \oplus m \oplus 0^{n-1}1 = F_k(IV + 1) \oplus m_0)$$

So if m_0 is encrypted then $c' = c$ and $\mathcal{A}$ outputs 0, while if m_1 is encrypted then $c' \neq c$ and $\mathcal{A}$ outputs 1. The overall success probability of $\mathcal{A}$ is therefore 3/4, and this modified CBC mode is not CPA-secure.

If the attacker can predict future IVs in AES-CBC, then AES-CBC is not IND-CPA secure


$\mathbf{0} \in M$
Chal.
$k \leftarrow K$
Adv.
$c_0 \leftarrow$ [Ir, $E_k(0 \oplus r)$]
predict
IV= r+1
$m_1 = (r \oplus (r+1))$, $m_1 \neq m_2$
$c \leftarrow$ [$(r+1)$, $E_k(r)$] or
$c \leftarrow$ [$(r+1)$, $E_k(m_2 \oplus (r+1))$]

If c= c_0 them the challenge cipher for m_1 otherwise for m_2

8. (**3 pts**) Let F be a PRF. Show that the following construction of MAC is **insecure**. Let $\mathcal{K} = \{0,1\}^n$ and $m = m_1 \parallel \cdots \parallel m_\ell$ with $m_i \in \{0,1\}^n$ for $i \in [1, \ell]$.
Pick $r \overset{U}{\leftarrow} \{0,1\}^n$, compute $t = F_k(r) \oplus F_k(m_1) \oplus \ldots \oplus F_k(m_\ell)$ and send (r, t).

This is **insecure** MAC.

If t is the tag for $m_1 \| m_2 \| \cdots \| m_\ell$, t would be a valid forgery for $m_2 \| m_1 \| \cdots \| m_\ell$, since changing the order of message blocks does not change the value of the tag.

$(r;\ t)$ remains a valid tag for any permutation of $m_1, m_2, \cdots, m_\ell$

9. (**2 pts**) Let (E, D) be an encryption system with key space $\mathcal{K}$, message space $\{0,1\}^n$ and ciphertext space $\{0,1\}^s$. Suppose (E, D) provides authenticated encryption. Does the following system provide authenticated encryption? Explain you answer

$E'(k, m) = (E(k, m) \oplus 1^s)$ and
$D'(k, c) = D(k, c \oplus 1^s)$

(E', D') provides authenticated encryption because an attacker on (E', D') directly give an attack on (E, D).

10. (**2 pts**) Does using a hash function provide authenticity? We have learned about the birthday paradox. What is its implication for hash functions?

No, everybody can use the same hash function. To provide authenticity, a digital signature or a MAC can be used.

For collision-resistance, the output size of a hash function needs to have twice the number of bits that would be necessary to prevent a brute-force attack.