UNIVERSITY OF MARYLAND
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

ENEE 457
Computer Systems Security
Instructor: Charalampos Papamanthou

Homework 2

Out: 09/24/18 Due: 10/01/18 11:59pm

Instructions

1. Strictly adhere to the University of Maryland Code of Academic Integrity.
2. Submit your solutions as a pdf document at Canvas. Include your full name in the solutions document. Name the solutions document as x-hw2.pdf, where x is your last name.

Problem 1 (20 points)

Let $\mathcal{E}_k(\cdot)$ and $\mathcal{D}_k(\cdot)$ be the encryption and decryption algorithms of a symmetric key cryptosystem with ℓ-bit keys and n-bit plaintexts and ciphertexts. We derive another symmetric key cryptosystem with a 2ℓ-bit key by applying twice $\mathcal{E}_k(\cdot)$ and $\mathcal{D}_k(\cdot)$:

$$\mathcal{E}'_{(k_1,k_2)}(M) = \mathcal{E}_{k_2}(\mathcal{E}_{k_1}(M))$$

$$\mathcal{D}'_{(k_1,k_2)}(C) = \mathcal{D}_{k_1}(\mathcal{D}_{k_2}(C))\,.$$

Consider an adversary who wants to perform a brute-force attack on the above cryptosystem to recover keys k_1 and k_2 but knows only a single valid pair of plaintext M and ciphertext C.

1. Suppose that the adversary has only $O(n)$ bits of space. Describe the attack in pseudocode and estimate the number of encryption and decryption operations performed in addition to the overall runtime of the algorithm.
2. Suppose now that the adversary has $O(n \cdot 2^\ell)$ bits of space. Describe, again in pseudocode, a more efficient way to perform the attack and estimate the number of encryption and decryption operations performed and the overall runtime.
3. In response to increases in computational power, the Data Encryption Standard (DES) was extended to support longer keys without designing a new algorithm. This was accomplished by developing Triple DES (3DES), which encrypts the plaintext three times with three different keys, with decryption applying the same keys in reverse order (the same as previously described scheme, but with one more key). Given again $O(n \cdot 2^\ell)$ bits of space, describe how to modify the attack of (2) to break 3DES. Again, estimate the number of decryption and encryption operations performed along with the overall runtime.

All answers estimating the number of cryptographic operations should be in big-O notation in terms of n and ℓ, and all answers should have at least a short justification.

Problem 2 (20 points)

1. Describe how Merkle-Damgard construction works.
2. Suppose that you can find two different large files (e.g., movies) f_1 and f_2 for which the Merkle-Damgard construction outputs the same hash. Show that this implies that you can also find a collision in the small hash function $h(.)$ that is used at each step of the Merkle-Damgard computation. You should consider two cases, one where the files are of the same size and one when the files are not the same size.

Problem 3 (20 points)

1. Two of the security properties of cryptographic hash functions are collision resistance and second-preimage resistance. A hash function is **collision resistant** if it is hard to find two inputs m_1, m_2 that hash to the same output. A hash function is **second pre-image resistant** if for a given input m_1, it is difficult to find a second different input m_2 that hash to the same output as m_1. Prove that collision resistance implies second pre-image resistance.
2. State the security property of a message authentication code (MAC) that we mentioned in class.
3. Let $f_k(m)$ be a secure message authentication code, i.e., it satisfies the previous definition (k is the secret key). Consider the following MAC that is based on $f_k(.)$ and is consisted of n bits. On input a message $a||b$ with $|a| = |b| = n - 1$ and key $k \in \{0,1\}^n$, the MAC is computed as
$$f_k(0||a)||f_k(1||b)\,,$$
where $||$ denotes concatenation. Is the new MAC secure? If yes, prove it. If not, give an attack. How about the MAC
$$f_k(0||a)||f_k(f_k(1||b))\,?$$
Again, either prove security or give an attack.

Problem 4 (20 points)

Let F be a pseudorandom permutation (PRP). Show that each of the following message authentication codes is insecure. (In each case the shared key is a random $k \in \{0,1\}^n$)

1. To authenticate a message $m = m_1||m_2||\cdots||m_l$, where $m_i \in \{0,1\}^n$, compute $t := F_k(m_1) \oplus \cdots \oplus F_k(m_l)$.
2. To authenticate a message $m = m_1||m_2||\cdots||m_l$, where $m_i \in \{0,1\}^n$, choose $r \leftarrow \{0,1\}^n$ at random, compute $t := F_k(r) \oplus F_k(m_1) \oplus \cdots \oplus F_k(m_l)$, and send $\langle r, t\rangle$.
3. To authenticate a message $m = m_1||m_2||\cdots||m_l$, where $m_i \in \{0,1\}^{n/2}$, choose $r \leftarrow \{0,1\}^n$ at random, compute
$$t := F_k(r) \oplus F_k(\langle 1\rangle||m_1) \cdots \oplus F_k(\langle l\rangle||m_l),$$
(where $\langle i\rangle$ is the $n/2$-bit encoding of the integer i), and send $\langle r, t\rangle$.

Problem 5 (20 points) DES algorithm is a cryptographic scheme that is used for symmetric encryption (besides AES). The output of a single round of the DES algorithm with input (L_i, R_i) (where L_i and R_i are 32-bit blocks) is given by the Feistel Network
$$L_{i+1} = R_i$$

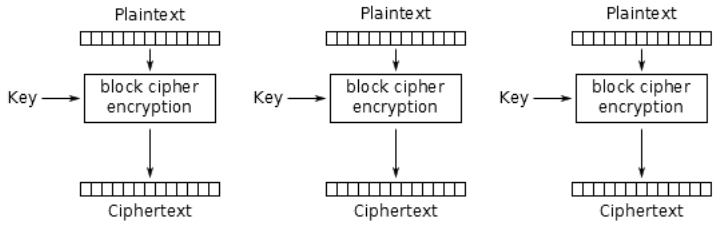
$$R_{i+1} = L_i \oplus f_{k_i}(R_i)\,,$$

where k_i is the 48-bit round key. Recall also that the output of the round function $f_{k_i}(.)$ is computed with the S-boxes, that map a 48-bit string (which equals $k_i \oplus e_i$ where e_i is the expansion of R_i into 48 bits) to a 32-bit string.

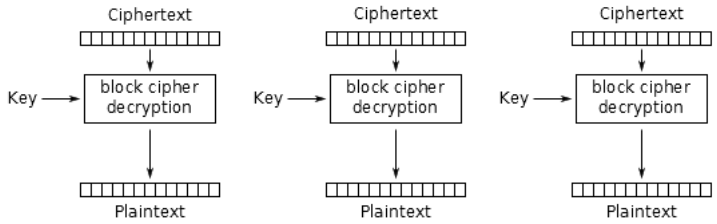
1. Why does DES use eight S-boxes, each mapping 6 bits to 4 bits, and not a single S-box that directly maps 48 bits to 32 bits? How many bits would be required to store such an S-box?
2. Devise an algorithm that breaks a single-round DES and retrieves the round key k_1. Assume you are given *one pair* of valid input/output (i.e., L_1, R_1 and L_2, R_2). How many operations does the algorithm require?
3. Devise an algorithm that breaks a double-round DES when you are given one pair of valid input/output (i.e., L_1, R_1 and L_3, R_3). Can this attack be extended to a triple-round DES (in this case you are given L_1, R_1 and L_4, R_4)? Justify your answer.
4. In class we talked about different modes of block cipher encryption. Below are diagrams detailing ECB, CBC, CFB, and OFB modes of operation. For each mode of operation, explain what happens, in terms of which plaintext block(s) or plaintext bit(s) get corrupted, when you flip a single bit in the ciphertext and then try to decrypt it using the same mode.

   Hint: In all these mode of operations there are either whole affected plaintext blocks or affected plaintext bits (not the whole block). In order to investigate the different modes, it is better if you present in detail the decryption formula for a plaintext block m_i. In addition, notice that XOR operation is bitwise.

   (a) ECB mode:

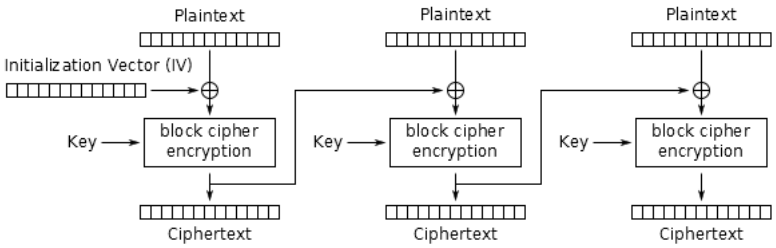


Electronic Codebook (ECB) mode encryption

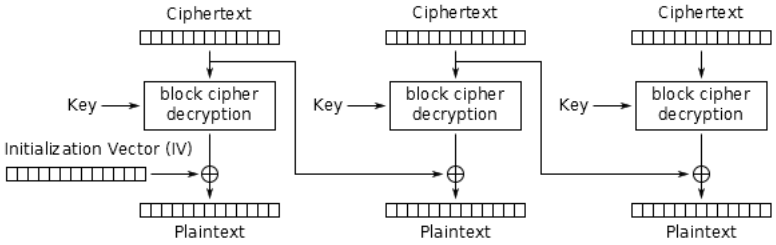


Electronic Codebook (ECB) mode decryption

(b) CBC mode:

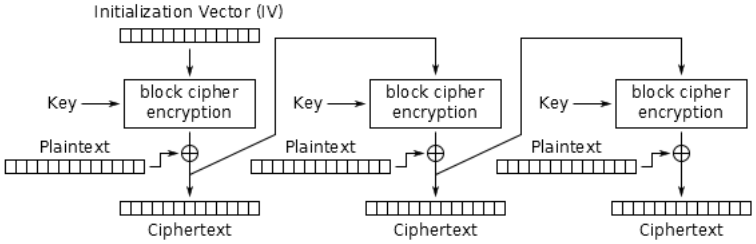


Cipher Block Chaining (CBC) mode encryption

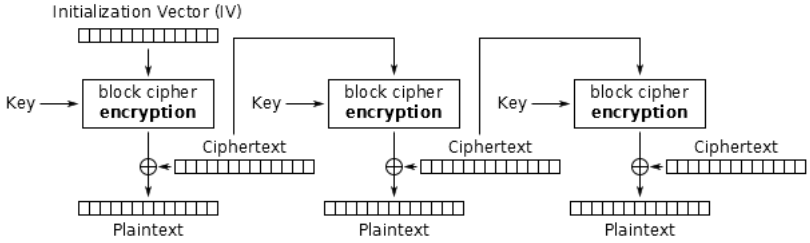


Cipher Block Chaining (CBC) mode decryption

(c) CFB mode:

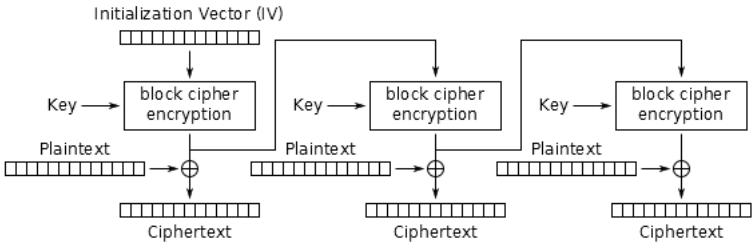


Cipher Feedback (CFB) mode encryption

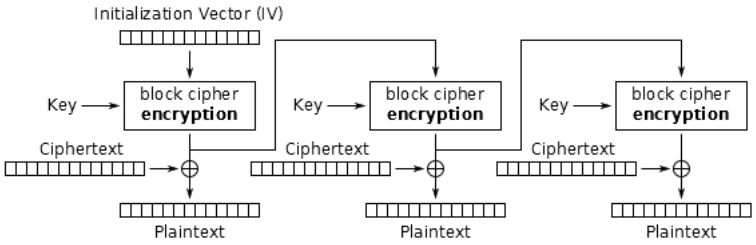


Cipher Feedback (CFB) mode decryption

(d) OFB mode:



Output Feedback (OFB) mode encryption



Output Feedback (OFB) mode decryption