UConn, School of Computing
Fall 2024
CSE 3400/CSE 5850: Introduction to Computer and Network Security / Introduction to Cybersecurity

Assignment 3

Instructor: Prof. Ghada Almashaqbeh
Student: Isaac Piegat
Posted: 9/28/2024
Submitted: 10/6/2024
Submission deadline: 10/5/2024, 11:59 pm

Notes:

- Solutions **must be typed** (using latex or any other text editor) and must be submitted as a pdf (not word or source latex files).
- This homework has a **shorter late days allowance** than usual. It will be **only 2 days** to allow us to post the key solution before midterm test 1. Thus, if you still have free late days (2 or more), you can use up to 2 days, and if not, there will be a deduction for the late day. After 2 days from the deadline no late submissions will be accepted (and the key solution will be posted).
- In all of the below, if the scheme is insecure then provide an attack against it and analyze its success probability/attacker's advantage. If the scheme is secure, just provide a convincing argument along with the attacker's advantage (formal security proofs are not required).

Problem 1 [60 points]
This problem is about encryption modes.

1. Rafa claims that if we modify the ECB mode as follows, it will become CPA secure: As before, a message is divided into blocks, so a message m would be $m = m_0 \| m_1 \| \cdots \| m_w$ (where w is an integer). For each message, we generate a fresh random string r of length n bits, and we encrypt the message as $E_k(m) = (E_k(m_1 \| r), E_k(m_2 \| r), \ldots, E_k(m_w \| r))$. Is Rafa's claim true? why?

   Rafa is correct. Concatonating each message block, each with their own unique random string of length n, means the encryption mode will become non-deterministic because even if a block like m_1 is reused, the random string concatonated onto the end would generate different outputs in the encryption function.

2. Coco wants to output one of the intermediary pads (say pad_1) in the OFB mode as part of the ciphertext of the messages she sends to Rafa. Does this modification preserve the CPA security of the OFB mode? why?

This modification does not preserve the security of the OFB mode because once Rafa knows the pad, Rafa can then derive the plaintext by xoring the ciphertext and now known pad. This information compromises the CPA security of the OFB mode.

3. In the CTR mode, Alice and Bob decided to sync their counters via two step increments instead of one as in the original CTR mode we studied in class. That is, for a message $m = m_1 \parallel m_2 \parallel \cdots \parallel m_w$ (where w is an integer), for m_1 the counter value would be s (the random initial value of the counter), for m_2 the counter value would be $s + 2$, for m_3 the counter value would be $s + 4$ and so on. Does this modification impact the CPA security and correctness of the CTR encryption mode? why?

   As long as the value of s is unique and not reused, the security and correctness is not impacted. Two step increments does nothing to change the uniqueness of s, and seeing how it is not reused because s is unique, the modification does not impact security or correctness.

4. What is the effect of the following ciphertext reordering/dropping/corruption on correctness of decryption for each of the OFB, CBC and CTR modes (note that for CTR mode $c_0 = s$ is the initial value of the counter, while it is IV for CBC and CTR modes). In all cases you have to justify your answers:

   (a) Alice sent Bob ciphertext $c_0, c_1, c_2, c_3, c_4, \ldots, c_w$, which was received by Bob as $c_0, c_1, c_3, c_4, \ldots, c_w$ (so c_2 was dropped).

   OFB - Only m_2 is effected as it is lost. All other plaintexts are accurate because the keystream is only generated the IV.

   CTR - Only m_2 is effected as it is lost. All other ciphertexts are accurate because the counters for $c_3, c_4, ...$ are independent.

   CBC - Only m_1 is not effected. Each block of ciphertext is dependent on the previous block, thus all blocks after c_2 are corrupted.

   (b) Alice sent Bob ciphertext $c_0, c_1, c_2, c_3, c_4, \ldots, c_w$, which was received by Bob as $c_0, c_2, c_4, c_3, c_1, \ldots, c_w$ (so there is a reordering of the ciphertext over the channel that Bob does not know about).

   OFB and CTR - Unaffected. All plaintexts will be accurate but in the wrong order because the decryption process does not rely on the previous cipher text block.

   CBC - All plaintexts except m_1 will be corrupted. CBC requires the previous block in order to decrypt the ciphertext correctly. Messing up the order would then mess up the decryption, corrupting all plaintexts except m_0 in this case.

   (c) Alice sent Bob ciphertext $c_0, c_1, c_2, c_3, c_4, \ldots, c_w$. Bob received the ciphertext (all of it in the same order) but with the last two bits of c_0 flipped.

OFB and CTR - Only m_0 is corrupted. All ciphertexts do not rely on each other for decryption, thus the rest are correct.

CBC - The incorrect bits in c_0 would corrupt m_0 and m_1 as the decryption process relies on the ciphertext of the previous block. All other plaintexts are correct as $c_1, c_2, ...$ are correct.

(d) Alice sent Bob ciphertext $c_0, c_1, c_2, c_3, c_4, \ldots, c_w$. Bob received the ciphertext (all of it in the same order) but with the left half of c_3 is flipped.

OFB and CTR - Only m_3 is corrupted. All ciphertexts do not rely on each other for decryption, thus the rest are correct.

CBC - The incorrect bits in c_3 would corrupt m_3 and m_4 as the decryption process relies on the ciphertext of the previous block. All other plaintexts are correct as $c_1, c_2, c_4, ...$ are correct.

Note: This problem has a bonus of 10 points.

Problem 2 [60 points]
Let $G : \{0,1\}^{n/2} \to \{0,1\}^n$ be a secure PRG, and $F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ be a secure PRF. For each of the following MAC constructions, state whether it is a secure MAC and justify your answers.

1. Given message $m \in \{0,1\}^{n/2}$, compute the tag as $MAC_k(m) = F_k(G(m)) \,\|\, LS2B(m)$, where $LS2B(m)$ is the last 2 bits of m.

   The MAC is secure because even though the attacker can identify the last two bits of the tag, this gives the attacker a non-neligible advantage in forging the tag of $1/2^{n-2}$. This number is still 0.

2. Given message $m \in \{0,1\}^n$, compute $y = F_k(m)$, parse $y = y_0 \,\|\, y_1$ such that $|y_0| = |y_1| = n/2$, then compute the tag as $G(y_0) \oplus G(y_1)$.

   This MAC is secure. The output would be of length $2n$ as you would pass both halves through the secure PRG (G) effectively doubling and adding two halves ($1/2 * 2 + 1/2 * 2 = 2$). The randomness of the PRG would then give the attacker no advantage, thus $(1/2^n)$ 0.

3. Given message $m \in \{0,1\}^{3n}$, parse m as $m = m_0 \,\|\, m_1 \,\|\, m_2$ such that $|m_0| = |m_1| = |m_2| = n$. Compute the tag as $MAC_k(m) = F_k(m_0) \,\|\, F_k(m_1 \oplus m_2)$.

   This MAC construction is secure because the attacker does not gain any significant advantage in causing a collision or forging a valid tag. The tag is indistinguishable from random, and the probability of causing a collision remains $1/2^n$.

4. A variation of the CMAC construction: Assume the message m to be of an even number of blocks (so it is a VIL whose length can be any even number of blocks). We compute a tag as $CMAC_k(m) = CBC - MAC_k(\frac{L(m)}{2} \,\|\, m_1 \,\|\, \cdots \,\|\, m_{L(m)/2}) \,\|\, CBC -$

$MAC_k(\frac{L(m)}{2} \parallel m_{(L(m)/2)+1} \parallel \cdots \parallel m_{L(m)})$, where $\frac{L(m)}{2}$ is a block representing half the length of the message m.

Provided that both halves of the message are fixed-length and the CBC-MAC is applied correctly to each half, it is secure. There is no significant advantage for the attacker because recombination or mixing halves from different messages would not yield a valid tag due to the security properties of CBC-MAC on fixed-length messages.

Note: This problem has a bonus of 10 points.