Assignment 2: COMP6453 2024 T2

This set consists of 3 questions with a total marks of 60.

Change log

- Question 2, Part 3, Compute $t := F_k(r) \oplus F_k(\langle 1 \rangle || m_1) \oplus \cdots \oplus F_k(\langle l \rangle || m_l)$

1 Block Ciphers modes of operation

Consider CBC and CTR modes of operation for block ciphers.

1. What is the effect on decryption in case of a single-bit error in the ciphertext when using the (i) CBC and (ii) CTR modes of operation?
2. Suppose a message is encrypted and the ciphertext blocks $c_1, c_2, \cdots, c_i, c_{i+1}, \cdots c_t$ are sent. One block c_i is dropped, such that the resulting ciphertext $c_1, c_2, \cdots, c_{i-1}, c_{i+1}, \cdots c_t$ are received. What is the effect on the decrypted message while using (i) CBC mode and (ii) CTR modes of operation?

Marks

- Part 1: 2 + 2 marks
- Part 2: 2 + 2 marks
- Total marks is 8.

2 Security of Message Authentication Codes

Let F be a pseudorandom function. Show that each of the following MACs is insecure even if used to authenticate fixed-length messages. In each case you should present a forgery that can be done efficiently. In each case, assume there exists a function $KeyGen$ that outputs a uniform $k \in \{0,1\}^n$. Let $\langle i \rangle$ denote an $n/2$-bit encoding of the integer i.

1. To authenticate a message $m = m_1, \cdots, m_l$, where $m_i \in \{0,1\}^n$, compute tag $t := F_k(m_1) \oplus \cdots \oplus F_k(m_l)$.

2. To authenticate a message $m = m_1, \cdots, m_l$, where $m_i \in \{0,1\}^{n/2}$, compute tag $t := F_k(\langle 1 \rangle || m_1) \oplus \cdots \oplus F_k(\langle l \rangle || m_l)$.

3. To authenticate a message $m = m_1, \cdots, m_l$, where $m_i \in \{0,1\}^{n/2}$ choose uniform $r \leftarrow \{0,1\}^n$, compute $t := F_k(r) \oplus F_k(\langle 1 \rangle || m_1) \oplus \cdots \oplus F_k(\langle l \rangle || m_l)$ and let the tag be $\langle r, t \rangle$.

Marks

- Part 1: 5 marks
- Part 2: 7 marks
- Part 3: 8 marks
- Total: 20 marks

3 Finding collisions on a 40-bit Hash

Write a program to find hash collisions on a 40-bit hash. The program should consist of a function hashCollision() that returns a tuple (m_1, m_2, n), where m_1 and m_2 are different ASCII strings whose SHA-1 hashes have the same high-order 40 bits (same 10 initial hex digits). The component n of the return value is the number of calls to SHA-1. You can generate random ASCII strings by converting random integers to hex.

1. What is the maximum number of calls to the SHA-1 function required?

2. Can you reduce the number of calls to the SHA-1 function? (Hint: Use a result from your class)

3. Write a pseudo code for hashCollision().

4. Implement the above in any programming language of your choice. You do not have to implement SHA-1 but use an appropriate library to generate SHA-1 digests. The output of the program should be two hash collisions obtained by generating two tuples (m_1, m_2, n) and (m_1', m_2', n'). While submitting the code, please add compilation instructions and execution instructions.

Marks

- Part 1: 2 marks
- Part 2: 2 marks
- Part 3: 10 marks. If your idea is correct but the pseudo-code is incorrectly written - 4 marks.

- Part 4: 18 marks. Inefficient code using brute force search - 8 marks.
- Total: 32 marks

Full Submission

1. Q1: Write answer in a file q1. You can use pdf or txt format.
2. Q2: Write answer in a file q2. You can use pdf or txt format.
3. Q3: Put the answers of the part (1), (2), (3) in a pdf or txt file, code and headers for part (4) all in a folder.
4. Upload a zip file with all three answers. $\langle zid \rangle \langle ass2 \rangle .zip$