MAS433 Cryptography: Tutorial 3
Block Cipher, Stream Cipher, Hash Function, MAC
15.10.2010

Problem 1. Modes of operation of block cipher
1.1. Consider those five modes of operation: ECB, CBC, CFB, OFB, CTR. If there is error in a ciphertext block (it is not the last ciphertext block), how many plaintext blocks would be decrypted wrongly?
1.2. A mode of operation operates as follows: $C_0 = IV$, $C_i = E_K(P_i) \oplus C_{i-1}$ for $i \geq 1$. Is this mode of operation better than ECB mode?
1.3. A mode of operation operates as follows: $P_0 = IV_1$, $C_0 = IV_2$, $C_i = E_K(P_i \oplus C_{i-1}) \oplus P_{i-1}$ for $i \geq 1$. How to decrypt the ciphertext? If there is error in a ciphertext block (it is not the last ciphertext block), how many plaintext blocks would be decrypted wrongly?

Problem 2. Meet-in-the-middle attack on block cipher
Apply the meet-in-the-middle attack to three-key Triple-DES. How to reduce the complexity of the attack to 2^{112} DES operations, and 2^{56} memory (we consider each unit of memory as 128 bits here) ?

Problem 3. Solve over-defined algebraic equations
3.1. In a system of algebraic equations over GF(2), if there are n binary variables, and the highest degree of monomials is d, then what is the maximal number of different monomials in this system of equations? How many such equations are needed in order to solve this system of equations through linearization? (Hint: for a monomial over GF(2), $x^2y^3z^8 = xyz$, i.e., the degree is 3)
3.2. There are several nonlinear equations over GF(7),

$$\begin{aligned}
x_1 + x_2 + x_1x_2 + x_2x_3 &= 1\\
x_1 + x_3 + x_1x_2 &= 0\\
x_1 + x_2x_3 &= 6\\
x_2 + x_3 + x_1x_2 &= 1\\
x_2 + x_3 &= 3
\end{aligned}$$

How to linearize the above over-defined equations?

Problem 4. Stream cipher
4.1. The A5/1 stream cipher consists of a 64-bit state. If we generate a long keystream from A5/1 using a key and an IV, what would be the estimated period of the keystream? (Hint: Is the state of A5/1 updated in an invertible way? Why? Can we apply the birthday paradox here?)
4.2. For the RC4 stream cipher, will two elements in the table S become identical? Why? How many possible values a table in RC4 can take?

Problem 5. Birthday Attack
5.1. Randomly select four persons. What is the probability that two of these four persons have the same birthday?
5.2. Randomly select 32 persons. What is the probability that two of these 32 persons have the same birthday?

Problem 6. Hash Function

6.1. For a hash function based on a modified Merkle-Damgard construction in which only '0' bits are padded to the end of the message, how to find a collision with low complexity?

6.2. For SHA-1, the message length is formatted as a 64-bit word, and it is padded to the message. If the message length is 200 bits, how many compression function operations are needed to compress this message? How many compression function operations are needed if the message lengthens are 0 bit, 1 bit, 447 bits, 448 bits, 511 bits, 512 bits, 960 bits?

6.3. Here is a method to protect the login passwords on a computer: when we create a user account in a computer system (Windows, Linux or UNIX), the login password of a user is hashed, and the hashed value (instead of the password) is stored on the computer. When a user login to the system, the computer hashes the password, and compares the hashed value with the stored hash value. With this approach, it becomes difficult to recover the passwords if an attacker (or even administrator) gains access to the computer.
6.3.1. What property of hash function is used in this password protection scheme?
6.3.2. Suppose that SHA-1 is used in this password protection scheme, what is the complexity to recover an equivalent password if the hashed value of an password is known to an attacker? (Here the equivalent password means that this password is hashed to the same value as the hash value being stored on a computer, but this password may or may not be the same as the original password.)

6.4. In a hash function based on Merkle-Damgard construction, the com-

pression function is given as $H_i = E_{m_i}(H_{i-1})$, where $E_k()$ denotes the encryption of an ideal block cipher with n-bit block size.
6.4.1. What is the complexity to find a preimage of this hash function? (Hint: meet-in-the-middle attack)
6.4.2. What is the complexity to find a second-preimage of this hash function?
6.4.3. What is the complexity to find a collision of this hash function?

6.5. In a hash function based on a modified Merkle-Damgard construction in which the IV is not fixed (i.e., an user is allowed to set the value of IV arbitrarily), the compression function is given as $H_i = E_{m_i}(H_{i-1})$, where $E_k()$ represents the encryption of an ideal block cipher with n-bit block size.
6.5.1. What is the complexity to find a preimage of this hash function?
6.5.2. What is the complexity to find a second-preimage of this hash function?
6.5.3. What is the complexity to find a collision of this hash function?

6.6. In a hash function based on a modified Merkle-Damgard construction in which the IV is not fixed (i.e., an user is allowed set the value of IV arbitrarily), the compression function uses the Davies-Meyer structure with an ideal block cipher with n-bit block size.
6.6.1. What is the complexity to find a preimage of this hash function?
6.6.2. What is the complexity to find a second-preimage of this hash function?
6.6.3. What is the complexity to find a collision of this hash function?

Problem 7. Message Authentication Code

7.1. After the CBC-encryption, denote the last ciphertext block as C_n. An MAC algorithm generates the authentication tag as: $t = E_{K_A}(C_n)$, where K_A is the secret key for MAC algorithm, and it is different from the encryption key. How to attack this MAC algorithm?

7.2. Denote the message as $M = m_1 \| m_2 \| m_3 \| \cdots \| m_n$. An MAC algorithm generates the authentication tag as: $t = E_K(m_1) \oplus E_K(m_2) \oplus \cdots \oplus E_K(m_n)$. How to attack this MAC algorithm?

7.3. CMAC
7.3.1. What are the differences between CBC encryption and CBC-MAC?
7.3.2. Why the CBC-MAC is insecure?
7.3.3. What are the differences between CMAC and CBC-MAC?
7.3.4. In CMAC, why the value of $K1$ is not simply set as $E_k(0)$?

7.4. HMAC

7.4.1. Consider an MAC algorithm constructed from a hash function using the key-prefix method. How to attack this MAC algorithm?

7.4.2. HMAC-SHA-1 is the HMAC constructed from SHA-1. It is now widely used for secure internet communication. For HMAC-SHA-1, how many compression function operations are needed to generate the authentication tag of an 800-bit message? How many compression function operations are needed if the message lengths are 0 bit, 960 bits?

7.5. Unconditionally secure MAC

In an unconditionally secure MAC, the message is given as $m_1, m_2, m_3, ...m_n$, the one-time key is given as $k_0, k_1, k_2, ...k_n$, where $m_i \in GF(97)$, and $k_i \in GF(97)$. The message authentication tag is generated as

$$t = (k_0 + \sum_{i=1}^{n} m_i \times k_i) \bmod 97\,.$$

7.5.1. The above unconditionally secure MAC is used for authenticating message. What is the probability that a message can be modified without being detected?

7.5.2. If k_0 is not randomly generated, and its value is '12' or '67' with equal probability. For a short message with only one block m_1, what is the probability that a message can be modified without being detected?

7.5.3. If the message is given as $m_1, m_2, m_3, ...m_n$, the one-time key is given as $k_0, k_1, k_2, ...k_n$, where $0 \leq m_i < 194$, and $0 \leq k_i < 194$. The message authentication tag is generated as $t = (k_0 + \sum_{i=1}^{n} m_i \times k_i) \bmod 194$. Is there efficient attack against this authentication scheme?

Problem 8. Initialization Vector

8.1 How to choose the initialization vectors for CBC mode, CFB mode, OFB mode, CTR mode, synchronous stream cipher, asynchronous stream cipher, hash function, CBC-MAC and CMAC?

8.2 How would the security be affected if identical IVs are used with the same secret key in the cryptosystems in Problem 8.1?