Symmetric Key Cryptography Exercises

Cryptology
FME (UPC) 2024
Jorge L. Villar

Last updated: Aug 8 20:10:15 2024

Linear Feedback Shift Registers (LFSR)

1. A LFSR of 3 cells produces a binary sequence starting with 110010. Compute its characteristic polynomial (or give the feedback function) and complete the previous output sequence. Which period has it?

2. Compute the minimal linear complexity of a binary sequence $x_1, x_2, ...$ starting with 0100111101101

   Hint: Iteratively, use values $m = 1, 2, ...$ and prove the non-existence of a F_2-linear map T_m such that $T_m(x_k, ..., x_{k+m-1}) = x_{k+m}$ for all k, until you reach a value such that the map exists. For instance, m cannot be 2 because $T_2(x_3, x_4) = T_2(0, 0) = 0 \neq 1 = x_5$. Gaussian elimination (in F_2) can help.

3. Describe a binary sequence that cannot be generated by any LFSR.

   Hint: A m-cells LFSR cannot produce a nontrivial sequence with more than $m - 1$ consecutive zeros. Why?

Practical Stream Ciphers

4. Using the specification of RC4, and assuming that a 3-bytes long IV is preppended to the long-term secret key k, show that with a noticeable probability the first byte of the keystream reveals the first byte of k, assuming that IV $= (3, 255, 0)$. Indeed, show that the most likely value for the permutation S after initialization fulfils $S[0] = 3$, $S[1] = 0$, $S[2] = 5$ and $S[3] = 6 + k[0]$, and then the first byte of the produced keystream is $S[S[1] + S[S[1]]] = 6 + k[0]$.

   Similarly, show that for IV $= (3, 255, 251)$ the most likely value of the first produced keystream byte is 3, and then it cannot be considered as pseudorandom.

Feistel Networks

5. Show that at least three iterations in a Feistel Network are necessary to achieve the diffusion property. To do that, assuming that each iteration can be written as $(L, R) \mapsto (R, L \oplus F_i(K_i, R))$, where $F_i()$ and K_i are the round function and the round key, show that for two iterations flipping a single bit of the L-part of the the plaintext results in flipping only the one bit in the L-part of the resulting ciphertext, and it is located at the same position.

Modes of Operation

6. In CBC and CFB modes the ciphertext block c_i is computed from the corresponding message block m_i and the previous ciphertext block c_{i-1} using XOR and the block cipher E_k. Show why the similar combination $c_i = E_k(m_i) \oplus c_{i-1}$ is not more secure than the basic ECB mode.

7. Show that no padding is necessary when using CFB, OFB or CTR modes of operation, because you can instead discard the unnecessary bits when masking the last incomplete message block. Why this trick cannot be applied to CBC or ECB modes?

8. Assuming that the probability that at least two of k independently chosen random values from a set of n values are equal (Birthday paradox) is about $k^2/(2n)$, give an estimation of the amount of information that can be encrypted with AES in CBC mode under the same key, such that the previous collision probability remains below 2^{-80}. What happens if AES is replaced by DES?

(Generated by lineprocx v2.97)

Plaintext Padding

A particular padding scheme is defined as $\text{Pad}(m) = (m, p, l)$, where $p = 10...0$ has exactly a 1-bit followed by zero or more 0-bits, and l is a 64-bit string containing the binary representation of the length of m (in bits). The string p has the minimal possible length so that the resulting length of $\text{Pad}(m)$ is an exact multiple of the block length (say 128 bits). We assume that the length of m is less than 2^{64} bits.

9. Compute the maximum difference between the lengths of $\text{Pad}(m)$ and m.

10. Show that Pad fulfils the three properties:
    - m is always a prefix of Pad(m).
    - If m and m' have the same length, then so do $\text{Pad}(m)$ and $\text{Pad}(m')$.
    - If m and m' have different lengths, then the last blocks of $\text{Pad}(m)$ and $\text{Pad}(m')$ differ.

Information Theoretic MAC

11. Consider the polynomial based MAC mentioned in section 2.1.1, defined as $\text{MAC}(k, m) = k_0 + k_1 m + ... + k_n m^n$, where $m, k_0, ..., k_n \in F_q$. Show that even if the attacker knows n valid pairs message/tag, he can only forge a new valid pair with probability $1/q$.

    Hint: Use polynomial interpolation to show that for any new message all possible values of the tag are equally likely.

CBC-MAC

12. Find a more general forgery against CBC-MAC for messages of arbitrary length, using the ideas in Proposition 6. Namely, given two valid message/tag pairs (m, t) and (m', t'), forge a new valid pair (m'', t'') where m'' is almost the concatenation of m and m' (you might have to change the first block of m') and $t'' = t'$.

13. One can try to define a CTR-mode based MAC (say CTR-MAC) by encrypting a message $(m_1, ..., m_n)$ using a block cipher E_k operating in CTR mode with a fixed IV (say IV=0), and then XORing all the ciphertext blocks to obtain the tag, $t = c_1 \oplus ... \oplus c_n$. Explain why it is insecure by showing a tag forgery for a two-block message.

Merkle-Damgård Construction

14. If H is a hash function using Merkle-Damgård construction using the length padding described in section 3.2.1, then show that given $H(m)$ and the length of m (but not m itself) an attacker can find some nonempty string x such that he can compute $H(m, x)$.

    Hint: From the length of m you can know the bits appended to m by the padding function. Then, you can append some extra blocks to $\text{Pad}(m)$ and use $H(m)$ to compute $H(m, x)$ iterating only for the new blocks.

Merkle Trees

15. How can the Merkle Tree construction be generalized to ternary trees? How many hash values have to be provided in a proof that an object belongs to the collection? Give an explicit construction for a set of 7 objects (documents), and a proof for the 4-th object.

Key reusing

16. Show the insecurity of using the same key k for encryption in CBC mode and for authentication with CBC-MAC. To do that, from a given pair message/ciphertext (m, c), show how to find a valid fresh message/tag pair m', t' where $m' \neq m$ for CBC-MAC without using the key k. Observe that CBC-MAC always uses IV = 0 in the CBC chain, while the value of IV used in the pair (m, c) will be nonzero.