Cryptography Engineering Midterm

April 2022

Problem 1 (20 pt)

In the following let p be a prime. The set $Z_p = \{\, x \text{ integer, such that } 0 \leq x < p\}$ is a group with respect to addition modulo p (i.e. every element x in Z_p has an inverse $-x \in Z_p$ such that $x + (-x) = 0 \; mod \; p$. The set $Z_p^* = \{\, x \text{ integer, such that } 0 < x < p\}$ is a group with respect to multiplication modulo p (i.e. every element x in Z_p^* has an inverse $x^{-1} \in Z_p^*$ such that $xx^{-1} = 1 \; mod \; p$.

Another cipher with perfect secrecy. Consider the following cipher.

Let Z_p^* be the message space, the key space and the ciphertext space.
Alice and Bob share a key $k \in Z_p^*$ uniformly chosen at random. To send a message $m \in Z_p^*$ to Bob, Alice computes the ciphertext $c = mk \; mod \; p$.

1. Prove that this cipher provides perfect secrecy using the criterium we proved in class.
2. Why one-time pad is a perfect secrecy?
3. Is the use of one-time pads susceptible to statistical analysis (especially if it is known that the plaintext is in American English?
4. Did public-key encryption scheme provide perfect secrecy? We assume there is a public-key encryption scheme (KeyGen,Enc,Dec) with perfect correctness (i.e., for all messages M and valid key-pairs (**PK,SK**), we have $Dec_{sk}(Enc_{pk}(M))=M$

Problem 2 (20 pt)

Predicting generators. Consider the following *congruential generator*. It uses constants $a, b \in Z_p^*$. The seed is a value $x_0 \in Z_p^*$. The i^{th} value generated is computed as

$$x_i = ax_{i-1} + b \; mod \; p$$

The sequence output by the generator is $S = x_0, x_1, x_2, \ldots$ Assume that an attacker knows p and witness the sequence.

1. Prove that after a short prefix (i.e. a few of the values x_i's) the attacker is able to predict the rest of the sequence (i.e. the rest of the x_j's).
2. What does this say about the security of using the congruential generator as the keystream generator for a stream cipher?
3. If an attacker knows constants $a, b \in Z_p^*$ and p. How many output bits $S = x_0, x_1, x_2, \ldots$ did the attacker to know to rest sequences.
4. However, If an attacker knows p but know nothing about constants $a, b \in Z_p^*$. In this case, How many output bits $S = x_0, x_1, x_2, \ldots$ did the attacker need to know to recover the rest sequence?

Problem 3 (10 pt)

Non-linear Composition of LFSRs. Consider the follow LFSR-based generator G. It is composed by three LFSR's R_1, R_2, R_3. Let $x_i(t)$ be the output of register R_i at clock pulse t. Then

$$G(t) = \left(x_1(t) AND x_2(t)\right) \oplus \left(\bar{x}_1(t) AND x_3(t)\right)$$

where $\bar{x}$ denotes the complement of bit x. Prove that

$$Prob[G(t) = x_2(t)] = \frac{3}{4}$$

Problem 4 (10 pt)

Passwords with insecure keyboard

Consider the following scenario:

Alice wants to login on a computer system. In order to gain access, she has to communicate her password to the system.

However, the keyboard (and the cable connection between the keyboard and the system) cannot be trusted since there may be a passive adversary recording the key strokes or tapping the line.

Conversely, let us assume that the display and computer used by Alice (and the connection between the system and the display) is secure, meaning it cannot be monitored by an adversary.

Devise a password protocol that will allow Alice to access the system without disclosing her password to the adversary.

Hint: You may apply Zero Knowledge Protocol.

Problem 5 (15 pt)

LFSR encryption algorithms to encrypt plaintext are theoretically breakable but if it is possible that you could add some improvement practical scheme to make a LSFR unbreakable. Please explain your methods.

Hint:https://crypto.stackexchange.com/questions/12754/can-a-lfsr-be-cryptographically-secure

Problem 6 (10 pt)

1.We have an Affine Ciphers $f(x) = (ax + b) mod\ 26$ How many keys are possible i.e. the key space

Hint: a *and* 26 *are relatively prime therefore GCD* $(a, 26) = 1$*, and we can always take* $1 \leq b \leq 26$

2.What is the key space for a Substitution Cipher $mod\ 26$

3.What is the key space for a Caesar Cipher $mod\ 26$.

Problem 7 (25 pt)

Alice and Bob share a secret key k. Bob wants an assurance that the message he receives are really coming from Alice. In order to obtain this Alice sends the message m and a MAC along with it $MAC_k(m)$.

We say that a MAC is secure against chosen message attack if an attacker Eve, after requesting the MAC $MAC_k(m_i)$ of n messages of her choice $m_1, \dots, m_n$ will not be able to produce a new message $m \neq m_i$ and a valid $MAC_k(m)$.

Recall CBC-MAC, a way of implementing MACs using symmetric encryption scheme. Let

$$f_k : \{0,1\}^l \rightarrow \{0,1\}^l$$

Be a block cipher with key k for example with l = 64, f could be DES.) Let m be a message of length bl bits.

$$m = m_1 \circ \ldots \circ \mathrm{m_b}$$

Where ∘ denotes concatenation and $\mathrm{m_i}$ is a block of length l bits. The MAC of m is computed using the cipher f in CBC mode:

$$\mathrm{MAC}_k(m) = f_k(f_k(\ldots f_k(f_k(m_1) \oplus m_2) \oplus \ldots m_{b-1}) \oplus m_b)$$

(a) Show that the CBC-MAC does not support variable length input. That is show that if messages have variable length it is possible to device a chosen message attack.
Hint: The idea is to ask for the MACs of two messages of length l and construct from them the MAC of a message of length $2l$.

(b) Suppose the length of the message is always smaller that 2^l. An attempt to get around the above problem would be to append the length of the message at the end as an extra block, before computing the MAC. Show that it is still possible to mount an effective chosen message attack.

Problem 8 (20 pt)

Eve intercepted a 4,096 bits encrypted voice message she devised a method to decrypt this encrypted message.
Eve notice that the underline plaintext voice in between Alice and Bob was coding by continues variable slope delta modulation (CVSD).

As we know CVSD is a delta modulation with variable step size (i.e., special case of adaptive delta modulation). As we know CVSD encodes at 1 bit per sample, so when voice is silent the plaintext will be encoded as 010101010101010101010101010
On the contrary, when speech it was encoded as
000000000011111111111110000000000111111111100000000.

Please find out your best way to help Eve break this 4,096 bits ciphertext and save the recovery plaintext into a binary file.

Hint1: You may apply Berlekamp-Massay algorithm to decrypt this ciphertext
Hint2: The key is generated by a lfsr whose length is at most 32
Hint3: The initial seed of the lfsr is 1.

Length of lfsr	Seed (binary)
3	001
5	00001

Note: Your program should read the bitstream from a file named “ciphertext.bin” and save the plaintext into “plaintext.bin”.