CS 166 ASSIGNMENT 1 SOLUTION

Overview

Marks on question:

- No mark: 1-part short answer question (the key point(s) for explanation is underlined)
- !: 1 part auto-graded question (answer in bold)
- +: 2-parts question with fill-in-blanks part and explanation part. The answer for fill-in-blanks part is in bold, the key point(s) for explanation is underlined.

Basic rules of grading: Full points if your explanation/equation was correct, even if the answer is wrong (because points already deducted in the fill-in-blank part). And you get 0.5 (on whole question, not each sub-question) as long as you wrote something.

Week 1 - Classical Ciphers & Stream Ciphers (Q1 – Q8)

!Q1: Find the key for a parameterized Caesar cipher (L1 P15)

Key should be the number of shifts. A quick way is to count the number of shifts based on plaintext 'a'.

In this problem, 'a' is encrypted to 'l' and 'l' is 8 letters away from 'a' so key = 8

!Q2: Calculate the expected time for exhaustive key search to break simple substitution (L1 P18)

I. Worst case: Trudy needs to check all 2^{88} keys.
So time (in years) = $2^{88}/2^{30} = 2^{58}$ seconds = $2^{58}/60/60/24/365 = \mathbf{9.1} * \mathbf{10^9}$ years

II. In average: Trudy needs to check about half of the 2^{88} keys (= 2^{87} keys) .
So time (in years) is also half of the worst case = $4.6 * 10^9$ years
Or $2^{87}/2^{30} = 2^{57}$ seconds = $2^{57}/60/60/24/365 = \mathbf{4.6} * \mathbf{10^9}$ years

+Q3: "Secure" vs. "Insecure" (L1 P20)

It is true that an "insecure" cipher might be harder to break than a "secure" cipher if based on the definition given in class.

I. Definition for "secure": A cipher system is secure if best known attack is to try all keys (i.e., no shortcut)

II. Based on the definition, parameterized Caesar cipher is considered as "secure" since the easiest way to break it is to try all 25 keys. While the general simple substitution cipher is considered as "insecure" since there's a shortcut to break it. However, obviously simple substitution cipher is harder to break than parameterized Caesar. This is an example of "insecure" cipher being harder to break than a "secure" cipher.
Apply to a general situation, a "secure" cipher with a small key space is easier to break compared to an "insecure" cipher with a shortcut that requires some extra analysis.

Yan Chen
Computer Science Department

+Q4: Decrypt a message that's encrypted using double transposition (L1 P21 & P22)

1. Put the ciphertext into a 7 * 10 matrix.
2. Find the row that contains all 5 letters for the first word "there". Which is row 3.
3. Move row 3 to the top.
4. Try all the column permutations to find one that will put "there" at first and result in meaningful words in each row.
5. Rearrange the rows to determine the row permutation.
6. Plaintext all in lowercases, no space or punctuations:
   therearesomewhosaythatcommunismisthewaveofthefutureletthemcometoberlin

FYI, the message is quote from President John F. Kennedy: There are some who say that communism is the wave of the future. Let them come to Berlin.

Q5: Using XOR properties in onetime pad (L1 P22 & 23)

I. To get K', Trudy computes $K' = P' \oplus C$.

II. Prove that if $K' = P' \oplus C$, then $P' = C \oplus K'$:

Since $K' = P' \oplus C$, then

$K' \oplus C = P' \oplus C \oplus C$ ($\oplus$ C on both sides)

$= P' \oplus (C \oplus C)$ (Associative)

$= P' \oplus 0$ (Self-Inverse)

$= P'$ (Identity)

That is, $P' = C \oplus K'$, Q.E.D.

!Q6: Analyze majority function used in A5/1 (L1 P14)

As the hint states, you can easily find the answer using a truth table. Actually, II, III, and IV are also easy to analyze without looking into the truth table. Let maj(X_8, Y_{10}, Z_{10}) = m

I. X steps means X_8 = m. As you can see in the truth table, in 6 out of the 8 rows X_8 = m. So, X steps 6/8 = 0.75 of the time.

II. X, Y, Z all steps means $X_8 = Y_{10} = Z_{10}$ = m.
That is, only 2 possibilities: either they all = 0 or they all = 1.
So, X, Y, and Z all step 2/8 = 0.25 of the time.

III. We only have 3 bits to consider, so majority means 2/3 or 3/3.
As we found in II, 3/3 happens 0.25 of the time,
then 2/3 happens 1 - 0.25 = 0.75 of the time.
You can also count the rows in the truth table to get the same answer.

We are considering the majority, so it is impossible that 0 or 1 bit to be the majority of the 3 bits.
So, at most 1 (0 or 1) of the registers step never happens, the answer is 0.

X_8	Y_{10}	Z_{10}	m
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Q7: Upper bound on the size of the RC4 state space. (L2 P18 & P19)

I. To compute the upper bound:
   - S has 256 elements, so we have 256P256 possibilities (permutate all 256 elements). And 256P256 = 256!
   - We have 256 different i's and 256 different j's. So, there are 256 * 256 = 2^{16} distinct indices i and j

   So, the final answer is 256! * 2^{16}.
   Note that other for the second part, 256P2 or 256C2 also will be considered as the correct answer.

II. We want to know the upper bound because we care about when the keystream will be repeated. And if the keystream generated repeats, then it will cause security problem, similar to using a one-time pad more than 1 time.

Q8: **Kerckhoffs' principle (L1 P11** & L2)

I. Kerckhoffs' principle: the strength of a cryptosystem depends ONLY on the key (or ONLY key should be hidden).

II. Since secret never remains secret, and we want the good guys to be able to find weaknesses before bad guys!

III. For example, A5/1 violates the principle, but then the secret got revealed and now it's broken although the algorithm is complex; Compared to RC4, which follows the principle, is still alive even though the algorithm is simple.
   (Other examples also ok, as long as show those not following the principle are broken (or secret is not secret anymore), while those follow the principle are still alive.)

Week 2 - Block Ciphers & Symmetric Summary (Q9 – Q19)

!Q9: Using S-Box in DES (L3 P13)

Among the 6 bits of the input, bits 0 and 5 for row; bits 1 through 4 for column.

Find the "intersection" of row and column, you can get the answer: 1110

	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
00	1110	0100	1101	0001	0010	1111	1011	1000	0011	1010	0110	1100	0101	1001	0000	0111
01	0000	1111	0111	0100	1110	0010	1101	0001	1010	0110	1100	1011	1001	0101	0011	1000
10	0100	0001	1110	1000	1101	0110	0010	1011	1111	1100	1001	0111	0011	1010	0101	0000
11	1111	1100	1000	0010	0100	1001	0001	0111	0101	1011	0011	1110	1010	0000	0110	1101

!Q10: AES, functions confusion or diffusion (L3 P18 - P21)

Only ShiftRow is primarily for diffusion
(easy to remember from the name: shift = permutate --> diffusion);
others (ByteSub, MixColumn, and AddRoundKey) are for confusion.

Q11: Feistel cipher (L3 P9 & P11 & P17 & P23)

I. In TEA algorithm, each block is divided to left and right parts, and the algorithm looks like a Feistel cipher. But there's no swap and each part doesn't $\oplus$ with the result of the round function. So, it is almost a Feistel cipher.

II. No, round function does not need to be invertible for DES.
Since DES is a Feistel cipher, that is, each round involving swapping left and right parts, and $\oplus$ right part with the result of round function. Both swap and $\oplus$ are invertible, the round function doesn't need to be invertible in order for the whole algorithm to be invertible.

III. Yes, round functions need to be invertible for AES.
AES is not a Feistel cipher, the round functions need to be invertible so the whole algorithm is invertible.

Q12: Stream ciphers vs. Block ciphers (L2 P8 & L3 P4 & L4 P16)

I. Stream ciphers rely on one-time pad while block ciphers rely on codebook.

II. For stream ciphers, each key generates a keystream which is used as one-time pad while in block ciphers, each key generates a different codebook.

!Q13: Block cipher modes overview (L4 P6)

False, block cipher modes are for all block ciphers in general, so they are not related to a particular cipher. That is, there's no relationship between how each block is encrypted and what mode is used to encrypt multiple blocks.

Q14: ECB weakness (L4 P8)

I. In ECB, the same plaintext block will result in the same ciphertext block. Therefore, If the ciphertext is large enough that there are repeated blocks, Trudy can get statistics.

II. Because the problem only occurs when there are same plaintext blocks and it's less likely to have repeated blocks when we have a small number of blocks.

+Q15: CBC encryption vs. CBC decryption (L4 P9 & P10)

I. The encryption rule for CBC is $C_n = E(C_{n-1} \oplus P_n, K)$, i.e., each ciphertext block depends on the previous ciphertext block. Therefore, once one of the ciphertext block is wrong, all ciphertext blocks after will be affected.
In this question, P_0 is wrong, so C_0 (the first ciphertext block) will be different wrong, and all blocks after will be affected.

II. The decryption rule for CBC is $P_n = C_{n-1} \oplus D(C_n, K)$, that is, each plaintext block is based on two ciphertext blocks, not another plaintext block. Therefore, once one of the ciphertext block is wrong, it will only affect 2 plaintext blocks at most.
So if C_0 is wrong (= X), some of the plaintext blocks will be affected, the plaintext blocks being affected are:

- $\mathbf{P_0} = IV \oplus D(X, K)$
- $\mathbf{P_1} = X \oplus D(C_1, K)$

+Q16: Decryption in CTR (L4 P11)

CTR is similar to stream cipher. That is, E(IV + n, K) can be viewed as the "keystream", then for each block, encryption is
C = P ⊕ keystream, and decryption is P = C ⊕ keystream.

+Q17: Integrity in symmetric key crypto (L4 P12 - P15)

I. "Provide integrity" means detecting/preventing any unauthorized writing.

II. For symmetric key ciphers encryption and MAC ...

- Encryption using symmetric key ciphers cannot provide integrity.
For example, if using a one-time pad, it is easy for Trudy to change the plaintext to a desired text P' by making a fake key = P' ⊕ C (showed in Q5).
Another example is, Trudy can change the order of the ciphertext blocks that are encrypted using ECB and the receiver won't be able to notice it.

- MAC (Message Authentication Code) can provide integrity.
Error propagates into MAC. That is, if Trudy changes any one of the plaintext (or ciphertext) blocks, the result MAC will be a different MAC. And Trudy cannot make the same MAC without knowing K.

+Q18: Always use the same IV instead of choosing IVs at random (L4 P9 & P11)

For CBC, the same initial plaintext block(s) will result in the same initial ciphertext block(s).

For CTR, the same plaintext will result in the same ciphertext.

Therefore, CTR is worse, since it then is just the same as ECB. For CBC, once the plaintext starts to differ, then the ciphertext will also starts to be different, even if there are repeated plaintext blocks.

Q19: 3DESs (L3 P14 & P15)

There is a possible attack to the "double-encrypt" $C = E(E(P, K_1), K_2)$ scheme based on the fact that D(C, K_2) = E(P, K_1).
If Trudy knows a plaintext P and the corresponding ciphertext C, she can first pre-compute all possible C_1 = E(P, K_1) and store the results in a table (C_1 and corresponding K_1). Then, she can try to decrypt C using all possible K_2, that is, compute D(C, K_2), until finding a match in the table. That is, the pair of K_1 and K_2 that makes E(P, K_1) = D(C, K_2) should be the correct keys.
Note that the table has 2^{56} entries, which means the work needed to break a "double-encrypt" DES is the same as to break a regular DES with 56-bit key. That is, the "double-encrypt" doesn't solve the key-too-short problem.