ECE 518 Homework 2

Alan Palayil

Due Date: 9/21/2022

A1: Problem 1.4 from Understanding Cryptography

1. There are 2^7= 128 characters possible and since the password is 8-characters long. The possible number of 8-character passwords is:

   2^{7*8}= 7.21x10^{16}

2. The key length in bits is the exponential power of the possible passwords calculated. Thus, the key length is 56.
3. If the password is restricted to the lower-case letters, then the characters decrease from 127 to 26. Thus, the possible number of 8-letter passwords:
   26^8 keys.

   The length of key in bits is equivalent to:

   $\log_2(26^8)$ = 37.60 bits.

4. A: To calculate the password length which corresponds to 128-bits, substitute a variable x required to be equal to 2^{128} bit key:
   $128^i=2^{7x}=2^{128}$
   7x=128
   x=18.29 ASCII characters
   B: To calculate the ASCII characters for lower-case letters (26) isn't represented with the power of 2.
   $26^x=2^{128}$
   x= $\log_{26}(2^{128})$
   x= 27.23 ASCII characters in lower-case letters

A2 A: Calculate 2x mod 13 for x = 1, 2, . . ., 12. With the formula is (A*B) mod C = (A mod C* B mod C) mod C

Since A is 2 and C is 13 with the value of B being a variable for 1, 2, …, 12. Thus, the equation is (2 mod 13*B mod 13) mod 13

Value of B	Value of equation
1	(2 mod 13*1 mod 13) mod 13 = (2*1) mod 13 = 2
2	(2 mod 13*2 mod 13) mod 13 = (2*2) mod 13 = 4
3	(2 mod 13*3 mod 13) mod 13 = (2*3) mod 13 = 6
4	(2 mod 13*4 mod 13) mod 13 = (2*4) mod 13 = 8
5	(2 mod 13*5 mod 13) mod 13 = (2*5) mod 13 = 10
6	(2 mod 13*6 mod 13) mod 13 = (2*6) mod 13 = 12
7	(2 mod 13*7 mod 13) mod 13 = (2*7) mod 13 = 1
8	(2 mod 13*8 mod 13) mod 13 = (2*8) mod 13 = 3
9	(2 mod 13*9 mod 13) mod 13 = (2*9) mod 13 = 5
10	(2 mod 13*10 mod 13) mod 13 = (2*10) mod 13 = 7
11	(2 mod 13*11 mod 13) mod 13 = (2*11) mod 13 = 9
12	(2 mod 13*12 mod 13) mod 13 = (2*12) mod 13 = 11

ECE 518 Homework 2

A2 B: Calculate 3x mod 13 for x = 1, 2, . . ., 12. With the formula is (A*B) mod C = (A mod C* B mod C) mod C

Since A is 3 and C is 13 with the value of B being a variable for 1, 2, …, 12. Thus, the equation is (2 mod 13*B mod 13) mod 13

Value of B	Value of equation
1	(3 mod 13*1 mod 13) mod 13 = (3*1) mod 13 = 3
2	(3 mod 13*2 mod 13) mod 13 = (3*2) mod 13 = 6
3	(3 mod 13*3 mod 13) mod 13 = (3*3) mod 13 = 9
4	(3 mod 13*4 mod 13) mod 13 = (3*4) mod 13 = 12
5	(3 mod 13*5 mod 13) mod 13 = (3*5) mod 13 = 2
6	(3 mod 13*6 mod 13) mod 13 = (3*6) mod 13 = 5
7	(3 mod 13*7 mod 13) mod 13 = (3*7) mod 13 = 8
8	(3 mod 13*8 mod 13) mod 13 = (3*8) mod 13 = 11
9	(3 mod 13*9 mod 13) mod 13 = (3*9) mod 13 = 1
10	(3 mod 13*10 mod 13) mod 13 = (3*10) mod 13 = 4
11	(3 mod 13*11 mod 13) mod 13 = (3*11) mod 13 = 7
12	(3 mod 13*12 mod 13) mod 13 = (3*12) mod 13 = 10

A2 C: Argue that if p is a prime number and 1 ≤ x < y ≤ p − 1 are two integers, then for any integer 1 ≤ a ≤ p − 1, ax mod p and ay mod p cannot be the same.

Using the contrary that ax mod p = ay mod p, implies that ax – ay mod p = a (x – y) mod p = 0 mod p which is p | a (x – y) –(i)

But Euclid's lemma, since a prime p divides the product ab of two integers a and b, then p must divide at least one of those integers a or b (p | a or p | b).

But from (i) p | a (x – y), p is either divides a or (x – y) and from the given condition from the statements 1 ≤ x < y ≤ p – 1 and 1 ≤ a ≤ p – 1. It is deduced that $|a| \leq p-1$ *and* $|x-y| \leq p-1$ which means that p does not divide a or (x – y).

Thus, by contradiction, $ax \ mod \ p \neq ay \ mod \ p$

A3 A: Calculate 2^x mod 13 for x = 1, 2, . . ., 12. With the formula is (A^B mod C = ((A mod C) ^B) mod C.

Since A is 2 and C is 13 with the value of B being a variable for 1, 2, …, 12. Thus, the equation is (2 mod 13*B mod 13) mod 13

Value of B	Value of equation
1	((2 mod 13) ^1) mod 13 = 2
2	((2 mod 13) ^2) mod 13 = 4
3	((2 mod 13) ^3) mod 13 = 8
4	((2 mod 13) ^4) mod 13 = 3
5	((2 mod 13) ^5) mod 13 = 6
6	((2 mod 13) ^6) mod 13 = 12
7	((2 mod 13) ^7) mod 13 = 11

8	((2 mod 13) ^8) mod 13 = 9
9	((2 mod 13) ^9) mod 13 = 5
10	((2 mod 13) ^10) mod 13 = 10
11	((2 mod 13) ^11) mod 13 = 7
12	((2 mod 13) ^12) mod 13 = 1

A3 B: Calculate 3^x mod 13 for x = 1, 2, . . ., 12. With the formula is A^B mod C = ((A mod C) ^B) mod C.

Since A is 3 and C is 13 with the value of B being a variable for 1, 2, …, 12. Thus, the equation is (2 mod 13*B mod 13) mod 13

Value of B	Value of equation
1	((3 mod 13) ^1) mod 13 = 3
2	((3 mod 13) ^2) mod 13 = 9
3	((3 mod 13) ^3) mod 13 = 1
4	((3 mod 13) ^4) mod 13 = 3
5	((3 mod 13) ^5) mod 13 = 9
6	((3 mod 13) ^6) mod 13 = 1
7	((3 mod 13) ^7) mod 13 = 3
8	((3 mod 13) ^8) mod 13 = 9
9	((3 mod 13) ^9) mod 13 = 1
10	((3 mod 13) ^10) mod 13 = 3
11	((3 mod 13) ^11) mod 13 = 9
12	((3 mod 13) ^12) mod 13 = 1

A3 C: What do the infinite sequences 2^x mod 13 and 3^x mod 13 look like for x =

1, 2, . . .,?

Using the formula, A^B mod C = ((A mod C) ^B) mod C using the tables from prior questions as reference, for 3^x mod 13 looking at the sequence repeating itself after every 4th number the infinite sequence would be:

3^x mod 13 = 3, 9, 1, 3, 9, 1, 3, 9, 1, 3, 9, 1, 3, 9, 1, …

For the 2^x mod 13, substituting the value of x to be 13, 14, 15:

Value of B	Value of equation
13	((2 mod 13) ^13) mod 13 = 2
14	((2 mod 13) ^14) mod 13 = 4
15	((2 mod 13) ^15) mod 13 = 8

From this it can be deduced that the sequence repeats itself every 13th number. So, the infinite sequence would be:

2^x mod 13 = 2, 4, 8, 3, 6, 12, 11, 9, 5, 10, 7, 1, 2, 4, 8, 3, …

A4: Problem 2.4 from Understanding Cryptography

An OTP system is immune to brute-force attack to implement a dictionary attack. It will return a dictionary back. So, if each bit of the output as A0 = B0 XOR C0, where A is the output, C is the key, and B is the message with one-time pad. Thus consider C to be "ESBT", thus if B is EAST and the output A is TYRA. So, there's no way of finding the possible message with that character length. With one-time pad it's completely rand with no basis to a brute force attack because there's no key, just a random pre-made keystream. Even if the keystream is guessed and even if the garbage characters are ruled out, there is still a lot of 4-character words which are non-garbage. So a you wouldn't know which is the correct message.

A5: Problem 4.16 from Understanding Cryptography

1. For the AES with 192-bit key length, takes $2^{189.7}$ operations.
   If ASIC can check $3*10^7$ key per second and if 100,000 such ICs are used in parallel. There are 31536000 seconds in a year. So, the number of years it takes to search the key is:
   $$Number\ of\ years = \frac{2^{189.7}}{3 \times 107 \times 100000 \times 31536000}$$
   $= 1.3473*10^{37}$ years
   The age of universe is 10^{10} years.
   $$\frac{1.3473 * 10^{37}}{10^{10}} = 1.33 \times 10^{34}$$
   So, the age of universe to search the key is $1.33*10^{34}$ times.

2. The number of Moore's Law iterations by x, it equates to:
   $$\frac{(5.304 \times 10^{38} years \times 365.25)}{2^x} = 1\ day$$
   $$Thus, x = 133.2\ iterations$$

The number of years= 1.5 years x 133.2 iterations= 199.8 years

A6: Problem 5.9 from Understanding Cryptography

1 TB contains 2^{40} bytes, and 1 byte contains 8 bits.

Thus, 1 TB = $2^{40}*8$

$= 2^{40}*2^3$

$= 2^{43}$ bits

The blocks required are:

$$= \frac{2^{43} bits}{128\ bits/_{block}}$$

$$= \frac{2^{43}}{2^7} blocks$$

$$= 2^{43-7} blocks$$

$$= 2^{36} blocks$$

The number of bits needed to counter value for each block:

$$= \log_2(2^{36})$$

= 36 bits

So, the maximum length of IV:

= Total no. of bits available – no. of bits required for counter value of each block

= 128 – 36 bits

= 92 bits

Therefore, the maximum length of IV for AES counter mode is 92 bits.