Homework 02

ECE443/518

By: Boyang Wang

1, We now consider the relation between passwords and key size. For this purpose, we consider a cryptosystem where the user enters a key in the form of a password.

- Assume a password consisting of 8 letters, where each letter is encoded by the ASCII scheme (7 bits per character, i.e., 128 possible characters). What is the size of the key space which can be constructed by such passwords?
- What is the corresponding key length in bits?
- Assume that most users use only the 26 lowercase letters from the alphabet instead of the full 7 bits of the ASCII-encoding. What is the corresponding key length in bits in this case?
- At least how many characters are required for a password in order to generate a key length of 128 bits in case of letters consisting of
  - 7-bit characters?
  - 26 lowercase letters from the alphabet?

Ans:

A, 8 letters, each letter is 7 bit, which makes 8*7=56bit, there will be 2^56= 72057594037927936 different combination of keys.

B, The corresponding key length in bits is 56

C, if we only use the lowercase from alphabet, there will be 26 options per character. There are 8 letters, which makes it 26^8= 208827064576 different keys. Log_2 208827064576 = 37.6035 bits

D.1 Roundup(128/7) = 19 characters, we at least need 19 ASCII characters for password to ensure a 128 bits key length

D.2 Each character contains Log_2 26 = 4.7 bits of data Roundup(128/4.7) = 28 characters We at least need 28 characters to achieve 128 bits key length.

2. *(3 points)*

A. Calculate $2x$ mod 13 for x = 1, 2, …. 12.

B. Calculate $3x$ mod 13 for x = 1, 2, …. 12.

C. Argue that if p is a prime number and $1 \leq x < y \leq p - 1$ are two integers, then for any integer $1 \leq a \leq p - 1$, ax mod p and ay mod p cannot be the same.

Ans:

A.

x = [1 2 3 4 5 6 7 8 9 10 11 12]

2x = [2 4 6 8 10 12 14 16 18 20 22 24]

2x mod 13 = [2 4 6 8 10 12 1 3 5 7 9 11];

B.

x = [1 2 3 4 5 6 7 8 9 10 11 12]

3x = [3 6 9 12 15 18 21 24 27 30 33 36]

3x mod 13 = [3 6 9 12 2 5 8 11 1 4 7 10];

C. If p is a prime number, for integers x and y meets $1 \leq x < y \leq p - 1$, then for any integer $1 \leq a \leq p - 1$, ax mod p and ay mod p cannot be the same.

Obviously a!=0, y>x, so ax!=ay

Suppose ax mod p is equal to ay mod p and is equal to z, we can know that, z+p*n = ax, z+p*m=ay, (n and m are integers), we only need to prove that z+p*n = ax, z+p*m=ay, (n and m are integers) is impossible

Subtract two equations, we have p*n-p*m=ax-ay

So p(n-m)=a(x-y) , furthermore, we have n-m=a(x-y)/p

n-m is integer, x-y is integer and x-y<p, a<p

Since p is prime number, the product of two number smaller than a prime number divide by a prime number cannot equal to an integer.

So, this is impossible.

Topic is proved.

3, (3 points)
A. Calculate 2^x mod 13 for x = 1, 2, …. 12.
B. Calculate 3^x mod 13 for x = 1, 2, …. 12.
C. What do the infinite sequences 2x mod 13 and 3x mod 13 look like for x =1, 2, ….?
ANS:

A.
x = [1 2 3 4 5 6 7 8 9 10 11 12]
2^x = [24 8 16 32 64 128 256 512 1024 2048
4096]
2^x mod 13 = [2 4 8 3 6 12 11 9 5 10 7 1]

B.
x = [1 2 3 4 5 6 7 8 9 10 11 12]
3^x = [39 27 81 243 729 2187 6561 19683 59049 177147
531441]
3^x mod 13 = [3 9 1 3 9 1 3 9 1 3 9 1]

C.
The pattern of 2^x mod 13 is
2 4 8 3 6 12 11 9 5 10 7 1
The pattern of 3^x mod 13 is
3 9 1

4, At first glance it seems as though an exhaustive key search is possible against an OTP system. Given is a short message, let's say 5 ASCII characters represented by 40 bit, which was encrypted using a 40-bit OTP. Explain exactly why an exhaustive key search will not succeed even though sufficient computational resources are available. This is a paradox since we know that the OTP is unconditionally secure. That is, explain why a brute-force attack does not work.

Note: You have to resolve the paradox! That means answers such as "The OTP is unconditionally secure and therefore a brute-force attack does not work" are not valid.

Ans: (by Jia Wang)

Just consider an example where the plaintext is BBBBB, i.e. 42 42 42 42 42 in ASCII/hex.

For an OTP of 01 01 01 01 01, the ciphertext is 43 43 43 43 43, i.e. CCCCC.

However, with just the ciphertext 43 43 43 43 43, it is also possible that the plaintext is AAAAA,
i.e. 40 40 40 40 40, since the OTP key could be 03 03 03 03 03.

In other words, without additional information regarding the plaintext, exhaustive key search is useless since every plaintext is possible and every key is possible.

5, The minimum key length for the AES algorithm is 128 bit. Assume that a special-purpose hardware key-search machine can test one key in 10 ns on one processor. The processors can be parallelized. Assume further that one such processor costs $10, including overhead. (Note that both the processor speed and the prize are rather optimistic assumptions.) We assume also that Moore's Law holds, according to which processor performance doubles every 18 months.

How long do we have to wait until an AES key search machine can be built which breaks the algorithm on average in one week and which doesn't cost more than $1 million?

ANS:

AES key is 128 bit, which means there are 2^128 different pwd combinations.

These machines can test 1 key in 10e-9 second.

The task is to build a machine to crack the AES 128bit key in one week within 1e6 dollar.

We assume the Moore law is not continuous, which means the performance only increase every 18 months. Also, we assume that the price of a single unit will not reduce. It will remain at $10 per unit

The unit that we can build = 1000000/10 = 100000

The password combination = 2^128 = 3.402823669209384634633746074317 7e+38

Current time in second = 2^128 * 10e-9 second / nodes can build

= 2^128 * 10e-9 / 100000

= 3.4028e25

Current time in day = current time in second / 60/ 60 / 24

= 9.45228797002607e+21

7 = Current time in day / 2^iterations

2^iterations = current time in day / 7

Iterations = $\log_2$(current time in day / 7)

= $\log_2$(2.3631e22/7)

=65.6088 iterations

It will take Roundup (65.6088) = 66 iterations to be able to break the 128bit AES in a week

72 iteration is 66*18/12 = 99 years

So, we will at lease wait for 99 years

6, We are using AES in counter mode for encrypting a hard disk with 1 TB of capacity. What is the maximum length of the IV?

ANS:

Maximum length of the Initialization (IV)

1TB = 2^10 GB = 2^20 MB = 2^40 BYTE

If we don't want to change the IV in the middle of the encryption, then:

Each block that is encrypted is 128 bit, so we need the counter to be at least 2^43/2^7=2^36 which is 36 bits.

The maximum length of IV is 128-36 = 92 bits.