ECE 518 Homework 3

Alan Palayil

Due Date: 10/05/2022

A1: Problem 11.2 from Understanding Cryptography

1. Attack A: Given the hash value, we are unable to locate the input string because of the hash's one-way property. Although it is possible to calculate the hash using the input string, it is not possible to calculate the input string using the hash. There is no way to determine the exact input string if the hash function is constructed correctly to have a one-way property. Since the hash function's one-way property cannot be broken if it is created correctly, this attack will fail.

Attack B: Assume that the hash function is h().Additionally, h(x) = m, where m is the hash and x is the string. Finding the second pre-image of the hash entails attempting to locate a second-string y with h(y) = m. A second preimage can be obtained through brute force, even though we are unable to determine the exact initial string with certainty. It will take a very long time for this attack. The complexity of its time is 2n.The attacker needs to know what kinds of passwords might be used and then use brute force to find a string with the same hash. There will be a 50% chance that each attempt will succeed. For such a brute-force attack, the rainbow attack with the rainbow table is frequently used. A rainbow table with passwords and their pre-hashed values is included in this.

As a result, the second preimages of h cannot be easily determined.

Attack C: Finding out m and m' without knowing any of them is referred to as a collision. Preimages are harder to find than collisions. This is because, following the discovery of 2n pairs of input and output, It becomes extremely likely that two of them will produce the same hash or output. We are unable to select which user's hash to break, which is a drawback. However, this approach is the most feasible for me if I do not care about a specific user but want to obtain as many passwords as possible. The complexity of its time is 2n/2.

Hence, this is the attack which has the most success rate in this scenario.

2. A rainbow attack is typically used in the brute force method of obtaining the password. It consists of a rainbow table that already contains the hashes of billions of passwords. Passwords can be recovered by matching the table to the database. Before the hash, some random text is added to the password. Usually, the text (salt) is long strings. Typically, the salt is added to short passwords to make it computationally challenging to integrate them with the rainbow table. To make brute force more difficult, salt should be added to passwords before their hash value is calculated.

3. There can be 2^{80} distinct hash values for an 80-byte hash output. Because 2^{80} values are used to accommodate all possible strings, there can at least be one collision in hashed 280 + 1 random strings. Using the current computation, it is not difficult to match against these many strings. This makes it more likely that a possible password will be revealed. As a result, 80 is not a very secure hash length.

A2: Problem 12.3 from Understanding Cryptography

1. When an attacker has access to both the plaintext and the cyphertext, stream cyphers are not secure. The attacker will also know the h(x) when they know the plaintext x. The attacker can then use arbitrary x' and h(x') to recalculate c after learning the keystream. Without knowing whether the sender is authenticated or not, the receiver will correct the message. OTP can also be used in this attack. OTP is completely secure against adversaries who are passive, but it does not protect against adversaries who are active.

2. When a keyed hash function like MAC is used to compute the checksum, the attack is ineffective. The equation says that the MAC uses the second key, k2.so that the generated verification tag cannot be broken. Since the attacker already knows x, even if he cracks the stream cipher, he won't be able to figure out the MAC tag without knowing k2.so that the recipient will immediately be informed of any changes he makes to x.

A3:

p= 11 and q= 19.

n = p x q = 11 x 19 = 209

$\Phi (n) = (p – 1) (q – 1) = (11 – 1) (19 – 1) = 180$

A. For e = 5. So,

$g (d (\Phi (n),e) =1; 1 < e < \Phi (n)$ –(i)

If e = 5. $g (d (\Phi (180),5) =5$ –(ii)

From (i) & (ii) e = 5 does not hold the condition.

Thus, it is not a valid choice.

B. For e = 7.

g (d (Φ (n),e) = g (d (Φ 180,7) = 7,1

which come under the condition 1 < 7 < 180. So, it is a valid choice.

Computing private key (d) by demod Φ (n) = 1. So, d = 103.

(103 x 7) mod 180 = (721) mod 180 = 1. So, d = 103.

C. Bob wants to send message x = 10. So, the cipher text $C = x^e \pmod n$

$y = (10)^7 \bmod 209 = 186$.

Decryption of y.

$x = y^d \bmod n$

$x = (186)^{103} \bmod 209$

So, $(186)^1 \bmod 209 = 186$

$(186)^2 \bmod 209 = 111$

$(186)^4 \bmod 209 = 199$

$(186)^8 \bmod 209 = 100$

$(186)^{16} \bmod 209 = 177$

$(186)^{32} \bmod 209 = 188$

$(186)^{64} \bmod 209 = 23$

$(186)^{103} \bmod 209 = (23)(188)(199)(111)(186) \bmod 209 = 10$ [103 = 64+32+4+2+1]

So, x = 10.

A4:

p = 13, q = 17, and e = 5.

n = p x q = 13 x 17 = 221

Φ (n) = (p – 1) (q – 1) = (13 – 1) (17 – 1) = 192

A. For computing private key (d)

demod Φ (n) = 1

d (5) mod 192 = 1

(77)(5) mod 192 = 1

So, d = 77. Thus, public key = { e, n } = { 5, 221 } and private key = { d, n } = { 77, 221 }

B. Bob want sign message x = 10.

So, cipher text $y = x^e \bmod n$

$y = 10^5 \bmod 221 = 108$ So, the cipher text is 108.

Alice verification i.e., decryption of y

$x = y^d \bmod n$

$x = 108^{77} \bmod 221$

$(108)^1 \bmod 221 = 108$

$(108)^2 \bmod 221 = 172$

$(108)^4 \bmod 221 = 191$

$(108)^8 \bmod 221 = 16$

$(108)^{16} \bmod 221 = 35$

$(108)^{32} \bmod 221 = 120$

$(108)^{64} \bmod 221 = 35$

$(108)^{77} \bmod 221 = (35)(16)(191)108) \bmod 221 = 10$ [77 = 64+8+4+1]

So, Alice verifies it as 10.

A5: Problem 8.5 from Understanding Cryptography

1. P = 467, $\alpha = 2$, a = 3, and b = 5

ALICE sends a key to the BOB. The key is computed as:

$A = \alpha^a \bmod P$
$= 2^3 \bmod 467$
$=8 \bmod 467$
$=8$

BOB sends a key to the ALICE. The key is computed as:

$B = \alpha^b \bmod P$
$= 2^5 \bmod 467$
$=32 \bmod 467$
$=32$

ALICE sends 8 to the BOB. BOB sends 32 to the ALICE.

The shared key (common key) for ALICE and BOB is:

$K = \alpha^{ab} \bmod P$
$= 2^{3x5} \bmod 467$
$= 2^{15} \bmod 467$
$=32768 \bmod 467$
$= 78$

ALICE computes the shared key with the known key 32 that is received from the BOB as:

$K_a = B^a \bmod P$
$=32^3 \bmod 467$
$=32768 \bmod 467$
$= 78$

BOB computes the shared key with the known key 8 that is received from the ALICE as:

$K_b = A^b \bmod P$
$=8^5 \bmod 467$
$=32768 \bmod 467$
$= 78$

Thus, since $K_a = K_b$. 78 is the shared secret key between ALICE and BOB.

2. P = 467, $\alpha = 2$, a = 400, and b=134

ALICE sends a key to the BOB. The key is computed as:

$A=\alpha^a \bmod P$
$= 2^{400} \bmod 467$
$=137$

BOB sends a key to the ALICE. The key is computed as:

$B = \alpha^b \bmod P$
$= 2^{134} \bmod 467$
$=84$

ALICE sends 137 to the BOB. BOB sends 84 to the ALICE.

The shared key for ALICE and BOB is:

$K=a^{ab} \bmod P$
$= 2^{400x134} \bmod 467$
$= 2^{53600} \bmod 467$
$= 90$

ALICE computes the shared key with the known key 84 that is received from the BOB as:

$K_a = B^a \bmod P$
$= 84^{400} \bmod 467$
$= 90$

BOB computes the shared key with the known key 137 that is received from the ALICE as:
It is computed as follows:

$K_b = A^b \bmod P$
$=137^{134} \bmod 467$
$=90$

Thus, since $K_a = K_b$. 90 is the shared secret key between ALICE and BOB.

3. P=467, α =2, a =228, and b=57

ALICE sends a key to the BOB. The key is computed as:

$A=\alpha^a \bmod P$
$= 2^{228} \bmod 467$
$=394$

BOB sends a key to the ALICE. The key is computed as:

$B = \alpha^b \bmod P$
$= 2^{57} \bmod 467$
$=313$

ALICE sends 394 to the BOB. BOB sends 313 to the ALICE.
The shared key for ALICE and BOB is:

$K=\alpha^{ab} \bmod P$
$= 2^{228x57} \bmod 467$
$= 2^{12996} \bmod 467$
$= 206$

ALICE computes the shared key with the known key 84 that is received from the BOB as:

$K_a = B^a \bmod P$
$= 313^{228} \bmod 467$
$= 206$

BOB computes the shared key with the known key 137 that is received from the ALICE as:
It is computed as follows:

$K_b = A^b \bmod P$
$=394^{57} \bmod 467$
$=206$

It is proved that $K_a = K_b$. 206 is the shared secret key between ALICE and BOB.

A6: Problem 13.11 from Understanding Cryptography

P = 467, α=2, a = 228, b = 57, and o = 16

i. o for K_{Ao} and K_{Bo}

$A = \alpha^o \bmod P$

$= 2^{16} \bmod 467$

$= 156$

$B = \alpha^o \bmod P$

$= 2^{16} \bmod 467$

$= 156$

A send to B and B sends to A.

$K_{Ao} = A^o \bmod P$

$= 156^{16} \bmod 467$

$= 113$

$K_{Bo} = B^o \bmod P$

$= 156^{16} \bmod 467$

$= 113$

Oscar uses the value o = 16

Bob calculated as $K_{Bo} = A^b \bmod P$

$= 156^{57} \bmod 467$

$= 438$

ii. Computing between Alice and Bob

$K_{Ao} = B^a \bmod P$

$= 156^{228} \bmod 467$

$= 243$

Bonus Problem 7.13 from Understanding Cryptography

1. The same number will be encrypted to the same cyphertext using RSA encryption, which is deterministic. As a result, Oscar can precompute and store the ciphertexts of given plaintexts in a code book. Oscar then refers to that code book to decrypt any new messages that come in.

2. Using the python code:

```
import numpy as np
start_val = 65
for plaintext in range (65,91):
    cyphertext = plaintext**11%3763
    print(chr(start_val),cyphertext)
    start_val = start_val + 1
```

The output is:

A 3288

B 705

C 3003

D 3335

E 153

F 2555

G 2698

H 2912

I 1125

J 1635

K 2464

L 1567

M 333

N 3368

O 2929

P 3696

Q 1720

R 1204

S 2514

T 3313

U 2499

V 2850

W 1222

X 1224

Y 3222

Z 2762

By looking up the code book, we can easily know that the text Alice wants to send is SIMPSONS.

3. Due to the increased key length, OAEP will help to prevent such an attack in practice. Oscar can no longer make the code book because n has a bit length that is reasonable.