CPSC 6200: COMPUTER SECURITY PRINCIPLES

Homework 2

Saurabh Sharma
10/01/2023

1. Complexities of Brute-Force Attacks (20 points)

1 Ans.

Problem statement: Each character can be any lower or upper case letters, a digit from 0 through 9, or one of 10 symbols. We need to find how many possible passcodes we can get to crack that PC.

Passcode to access the PC = five characters
(a-z) = 26
(A-Z)= 26
(0-9) = 10
Other Symbols =10

By adding the above three, we get 72.
Then we need to do
= 26+26+10+10
= 72C5
= 72!/(72-5)! 5!
= 72!/67! 5!
If repeated characters allowed=1,39,91,544.

2 Ans.

Problem statement: Alice sends an encrypted document to bob. The characters are also the same as in the First question. The encryption scheme is a Substitution, and then Eve is also called Malori, and we need to find how many trails Eve to decrypt the ciphertext in the worst case.

(a-z)= 26
(A-Z)= 26
(0-9)= 10
other Symbols = 10
Space= 1
Now, adding this 26+26+10+10+1=73
No. of Characters in Cipher text = No. Of Characters in Plain text in the Substitution method.
Total number of attempts in worst case = 73Cn
= 73!/ ((73-n)!*n!)

3 Ans.

Total no. of Characters in Alice Password= 8
(a-z) = 26
(A-Z) = 26

(0-9)= 10
other Symbols = 10
No. of Characters can choose from per input = 26+26+10+10
= 72.

Total number of attempts in worst case = 72C8
= 11,96,90,16,345

Total time for all attempts in worst case = 72C8*2
= 23,93,80,32,690 sec

To Covert, in days, we need to divide it by 3600- sec in an hour& 24 hours per day.
= 23938032690/3600*24
= 277060.5635 days.

(A). The server allows at most 5 login attempts within one minute.
One attempt will take 12 secs
72C8 attempts will take : 12*72C8/3600*24= 1662363.38125 days

(B). The server allows at most 15 login attempts within 30 minutes.
One attempt will take 120 secs
72C8 attempts will take : 120*72C8/3600*24= 16623633.8125 days

2. Bail-Out Money (20 points)

Given

Let Pb and Pt be the Public Keys of Barack and Tim. A message m sent by Barack to Tim is Transmitted as Ept(m), and the reply r from the Tim to Barack is transmitted as Epb(r).

1. The attacker employs a "Man-in-the-Middle" attack to discover how much each bank's bail is worth. A man-in-the-middle attack is a form of active listening in which the assailant sends messages between two parties communicating while giving the impression that they are speaking directly to one another.
2. In this attack, the attacker only requires the public keys, which are widely accessible, rather than the private keys of the two communicating parties.

1Ans :
Problem Statement: Describing how the attacker can learn the bailout amount for each bank even if he cannot derive the private keys.

● The adversary carries out a dictionary attack. The attacker encrypts the 900B candidate answers from Tim using public key pT and the 10 candidate messages from Barack using public key pB since the message format is fixed and there are 10 potential banks and 900B possible bailout amounts.

● The attacker then compares the ciphertexts that Barack and Tim exchanged with the ones that had already been calculated to discover the matching plaintexts. Keep in mind that the attacker does not require access to the private keys that Tim and Barack utilize.

2Ans :

As a result of the above attack, Barack decides to modify the protocol for exchanging messages. Two simple protocol modifications will make the above attack much more difficult. The first should use random numbers, and the second one should use symmetric encryption.

An attacker can interfere with the communication between Barack and Tim as follows:

● Barack asks Tim for his public key. When Tim transmits his public key to Barack, the attacker intercepts the communication and sends a message to Barack that is falsely identified as coming from Tim but really contains the attacker's public key (pr) (PA).

● Barack thinks Tim is the rightful owner of the public key PA. The attacker has the same ability to convince Tim that Barack is the rightful owner of the public key PA. Now, Barack uses PA (E, A (m)) to encrypt his message before sending it to Tim.

● Another transmission is intercepted by the attacker, who uses his private key SA to interpret the message (Ep (m)) and records the name of the bank that Barack sent. In the following phase, the adversary uses Tim's public key to encrypt the same message before sending his essential general PA as Barack's public key.

● Tim encrypts the bail amount in his response using the attacker's public key (pA). The assailant will send the message to Barack by encrypting it with his public key after first decrypting it with his private key.

Finally, by using this Simple modification like random variables and Symmetric encryption, we can increase the difficulty of attacks by adversaries.

3. RSA Cryptosystem (30 Points)

1Ans :

1. The original encrypted message is Memodn, and the signature file for the message is Me mod n to Eve. The signature algorithm is Memodn, which means the signature is encrypted using Bob's private key.

2. The original message M originated from Bob for Eve with its public key and modulus value (e,n), meaning the original encrypted message is Me mod n. Eve calculates the size of the signature-encrypted message after it has been decrypted. Eve is aware of the signature algorithm but is unaware of the value of d. (Bob's private key).

• Now, if Eve obtains any encrypted communication or cipher message C from Alice (intended for Bob), she will attempt to decipher it to get the value of P.

• In the RSA cryptosystem, this assault is known as a "plaintext attack," in which the user delivers the cipher value twice to recover the original message value.
• For this method, Eve uses the random integer value r. With Alice's cipher value, create a message value that looks like ($C*r^e \bmod n$).

Bob receives this value in order to verify the signature, and if Bob responds, he sends back $K=(C*r^e \bmod n) \bmod n$. The S further validates the K's value. Eve can now compute the value using the cipher values C and random r.

Therefore,

$$K = (C*r^e \bmod n)^d$$
$$= C^d \bmod n * r^{ed} \bmod n$$
$$=(c^d * r) \bmod n$$

Where r is a random value $(P*r) \bmod n \,||\, C^d \bmod n = P$ as per the RSA rule.

Now that K's value has been saved in a file, P may be calculated using modular arithmetic.$P= K*r^{-1} \bmod n$ Eve can now determine the value of P using the known values of K and r.

• Hence, the modular arithmetic technique is utilized to make Alice's plaintext P and signature S available to Eve.

2Ans : Bob prevents this attack. This only pertains to RSA taught in textbooks. When RSA is used correctly, padding schemes are used to prevent the cipher - text from being malleable in this way, thwarting the attack.

4. RSA Public-Key Encryption Lab (30 points)

1Ans. Deriving a private key

```
BIGNUM *p = BN_new();
BIGNUM *q = BN_new();
BIGNUM *e = BN_new();

// Assign the first large prime
BN_hex2bn(&p, "F7E75FDC469067FFDC4E847C51F452DF");

// Assign the second large prime
BN_hex2bn(&q, "E85CED54AF57E53E092113E62F436F4F");

// Assign the Modulus
BN_hex2bn(&e, "0D88C3");

BIGNUM* priv_key1 = get_rsa_priv_key(p, q, e);
printBN("the private key for task1 is:", priv_key1);

BIGNUM* enc = BN_new();
BIGNUM* dec = BN_new();
```

```
BIGNUM* get_rsa_priv_key(BIGNUM* p, BIGNUM* q, BIGNUM* e)
{
        /*
                given two large prime numbers, compute a private key
                using the modulo inverse of the totatives of the product p*q
        */
        BN_CTX *ctx = BN_CTX_new();
        BIGNUM* p_minus_one = BN_new();
        BIGNUM* q_minus_one = BN_new();
        BIGNUM* one = BN_new();
        BIGNUM* tt = BN_new();

        BN_dec2bn(&one, "1");
        BN_sub(p_minus_one, p, one);
        BN_sub(q_minus_one, q, one);
        BN_mul(tt, p_minus_one, q_minus_one, ctx);

        BIGNUM* res = BN_new();
        BN_mod_inverse(res, e, tt, ctx);
        BN_CTX_free(ctx);
        return res;
}
```

Result:

```
d =  3587A24598E5F2A21DB007D89D18CC50ABA5075BA19A33890FE7C28A9B496AEB
Verify: e*d mod (p-1)*(q-1)= 01
```

2Ans. Encrypting a Message

```
// Assign the private key
BIGNUM* priv_key = BN_new();
BN_hex2bn(&priv_key, "74D806F9F3A62BAE331FFE3F0A68AFE35B3D2E4794148AACBC26AA381CD7D30D");

// Assign the public key
BIGNUM* pub_key = BN_new();
BN_hex2bn(&pub_key, "DCBFFE3E51F62E09CE7032E2677A78946A849DC4CDDE3A4D0CB81629242FB1A5");
printBN("the public key is: ", pub_key);

// Assign the Modulus
BIGNUM* mod = BN_new();
BN_hex2bn(&mod, "010001");

// We are going to encrypt the message 'A top secret!'.
// In order to use RSA, first we need to convert this message into hex.
// Then we can convert the hex into a BIGNUM for the computations.
BIGNUM* message = BN_new();
BN_hex2bn(&message, "4120746f702073656372657421");

printBN("the plaintext message for task2 is: ", message);
enc = rsa_encrypt(message, mod, pub_key);
printBN("the encrypted message for task2 is: ", enc);
dec = rsa_decrypt(enc, priv_key, pub_key);
printf("the decrypted message for task2 is: ");
printHX(BN_bn2hex(dec));
printf("\n");
```

Result:

```
Encrypted Message=  A142B0BFA3DD2C1A5B4859B06D7CA6C420F3205E9CA702E1F6B059FAD6EF3977
```

3Ans. Decrypting a Message

```
BIGNUM* task3_enc = BN_new();
BN_hex2bn(&task3_enc, "8C0F971DF2F3672B28811407E2DABBE1DA0FEBBBDFC7DCB67396567EA1E2493F");

// We already have the public and private keys.
// We can decrypt using our rsa_decrypt function.
dec = rsa_decrypt(task3_enc, priv_key, pub_key);
printf("the decrypted message for task3 is: ");
printHX(BN_bn2hex(dec));
printf("\n");
```

Result:

```
Decrypted Message =  48656C6C6F20776F726C640D0A
```