1. We have a symmetric encryption algorithm $E_K(M)=C$. Here K is the secret key, M is the plaintext, and C is the ciphertext. We (and the attacker) know that the key length is 192 bits. The attacker eavesdrops on the communication line and gets a copy of the ciphertext C1. Now the attacker decides to conduct the brute force attack and try every possible key to get the plaintext M1. Let us assume that there is only one possible M1 and if the attacker sees it, he will know that this is the correct one. The attacker has 1,000,000 machines, with each machine having the capabilities to try 5,000,000 decryption of C1 with different keys per second. If one machine finds the right key, it will automatically notify the attacker. Now please answer, how many years (roughly) does the attacker need to try 50% of the keys?

Note that Google has around 2 million servers. Also, check the Internet and see what the expected life time of the Sun is. Can you crack the key before that?

Answer:

Given;

length of the key i.e. n is 192 bits
Attacker has machines = 1* 10^6 [i.e. 1,000,000]
Each machine's capacity = 5*10^6 decryption of C1 with different keys per second [5,000,000]
Attacker using brute force attack to try every possible key to get the plaintext M1.

To find;

Result = 50% of the keys attacker needs to try.

Therefore,

Capability for decryption of C1 with different keys per second is
= (5*10^6) *(1*10^6)
=**5*10^12**

As said, attack tries to conduct brute force attack method to find every possible key.
Thus, the attacker needs to try
= 2^n possible attempts

But n = 192,

=2^192

But as said in the question, we need to find only 50 % of the keys attacker needs to try. So, therefore
= 2^192 * 0.5 possible attempts

Thus, time taken in seconds for attacker to try above 50% of the keys
= (2^192 * 0.5) / (5*10^12)

= 6.2777*10^44 seconds

Conversion seconds into year:
6.2777*10^44 = > 6.2777*10^44 /(60*60*24*365) ≈ 2*10^37 years

Along with, Expected Life time of sun (from internet) = 5 *10^9 years

Since, 2*10^37 years > 5 *10^9 years

We can conclude that, 50 % of the keys are not sufficient to crack the key before the expected life time of the sun.

---

2. Let us consider the block cipher type of symmetric encryption. The basic idea is that you have a block of plaintext (for example, 128 bits) and a key (for example, 128 bits) as inputs to the encryption algorithm. The output will be a block of cipher-text (for example, 128 bits). If the encryption algorithm does not conduct compression, the output block size will be at least as large as the plaintext block (in other words, the cipher text is of the same size or longer than the plaintext.) Please explain why it is like this.

What is symmetric encryption?

- Symmetric encryption uses same cryptographic keys for both encryption plain text and decryption ciphertext. Symmetric keys can use either of the below –
- Stream cipher
- Block cipher.
- As said, the second type i.e. block cipher is used in the above encryption.

What is block cipher?

A block cipher takes a block of plaintext bits and generates a block of ciphertext bits usually of same size. The size of block is fixed in the given scheme

What is block cipher padding?

- Block ciphers process keeps blocks of fixed sizes – for example a block of 128 bits.
- The length of plaintexts is mostly not a multiple of the block size.i.e The block sizes are not same for all blocks.
- For example, a 152-bit plaintext provides two blocks of 64 bits each with third block of balance 24 bits. In our example, the remaining 22 bits need to have additional 42 redundant bits added to provide a complete block.
- This process of adding bits to the last block is referred to as padding.
- Also, these blocks of plaintext will go through an encryption function the nature of encryption function is dependent on the algorithm used, but every encryption function with

respect to block cipher takes in an input block(plaintext) of size n bits and key of k bits as input parameters and yield an n-bit output block.

Encryption function

Below is the encryption function for block cipher:

$(P) := E(K,P): \{0,1\}^k * \{0,1\}^n \rightarrow \{0,1\}^n$

Where,
K-Key of size k bits
P- n bits Plaintext size/block size

Conclusion:

Thus, from the explanation and the above equation we can conclude below;

"A block cipher is designed in such a way that every encrypted message will have a size equal to the block size"

Therefore,
Given:

block size n =128 bits
Key size k= 128 bits.

Considering, encryption algorithm does not conduct compression - applying the concept of block cipher on the above condition, the encrypted output should be having the size n that is 128 and padding the ciphertext will be of the same size or longer than the plaintext and never be less than its size.

---

3. Bob has a public-private key pair (pub_Bob, pri_Bob). Alice needs to send some information to Bob. She wants to make sure that when Bob opens the message, he can verify that this is from Alice but not anyone else. So she sends out the message as: [Alice, E_{pub_Bob}(message)] to Bob. Basically, she first sends out her name in clear text, then encrypts the message with Bob's public key. Please discuss, can an attacker M send out a packet in Alice's name? How can he do it? Here we assume that M also has the public key of Bob. For the same question, if Alice sends out [E_{pub_Bob}(Alice, message)], can M still impersonate Alice? (Here Alice puts her name in the encryption.)

Answer:

Given in above question is the case where Alice wants to send message to Bob and Bob has (public – private) key pair i.e. (pub_Bob, pri_Bob).

As informed above Alice will have the public key of Bob (pub_Bob) which Alice uses to encrypt the message which she wants to send Bob.

Therefore, Given below =
M also has the public key of Bob

Considering below cases –

Case 1:

Alice sends out the message as: [Alice, E_{pub_Bob}(message)] to bob :

- As Alice sends the message with her name in clear text and then encrypts the message using Bob's public key to bob.
- The attacker M can easily impersonate Alice and send false message to Bob because the name Alice is in Plain text and which can be easily identified by the attacker.
- Now as the attacker knows the public key of bob, M(attacker) destroys the message from Alice and sends a different message to bob which will be having Alice name in plaintext and encrypted false message/information. Thus, the message received by the Bob will be of the below form:
- [Alice, E_{pub_Bob}(message)]
- which in turn Bob may assume that it is from Alice but in reality it is a fake/modified message from the attacker.

Case 2:

Alice sends out the message as: [E_{pub_Bob}(Alice, message)] to bob :

- In this case Alice will encrypt her name within the message.
- In a case where attacker get this packet, he will destroy it and by using the public key of bob he encrypts a malicious/fake message and sends it to Bob.
- Since, the attacker is now unaware that the name of Alice was also present in the encrypted message which came from Alice .
- Therefore, the attacker may leave the name field empty or even if the name field is added by the attacker the name will vary or the format of name will be different.
- Thus, when Bob decrypts he will identify that the message is not from Alice.

---

4) David is a lobbyist and he is secretly visiting different states for a marketing plan. Every midnight, David will send an email to Bob, who is his supervisor, to report the two states that he will visit tomorrow. To protect the information, David will encrypt the message with Bob's public key. For example, if David will visit North Carolina and South Carolina tomorrow, the message he sends to Bob will look like $(NC, SC)_{pub\text{-}Bob}$. (which means the short names of the two states

encrypted by the public key of Bob.) A reporter, Alice, is following the secret plan. Alice gets a copy of Bob's public key but she does not know the private key of Bob. One night, Alice uses her laptop to eavesdrop on the message that David sends to Bob. She gets a copy of the encrypted message. Please illustrate how Alice can use forward search to figure out which two states David will visit tomorrow.
Hint: this is an example of forward search attack.

Answer:

Summarizing the above problem, we can consider the below users involved:

a) David: A lobbyist who is secretly visiting different states for marketing plan
b) Bob: David's supervisor
c) Alice: Adversary.
Therefore, above is a type of **asymmetric encryption**.

Given;

- David sends encrypted secret message to Bob using Bob's public key which will look like (NC, SC)pub-Bob which means the short names of the two states encrypted by the public key of Bob.
- Thus, set of plain text messages is small.

The adversary Alice is following the secret plan of David and Bob and also has the public key of Bob.

Given a hint: forward search attack

Therefore,

- As the size of plain text is small and **forward search attack is feasible**. She can compute the total of combinations using

  50C2

  Where, 50 is the total number of states in USA
- 2 is the number of states that David visit every time which 1225 combinations is.
- In each combination, Alice applies Bob's public key and obtains the corresponding cipher text.
- She again compares each cipher text with the acquired encrypted message and finds out the matching patterns.
- Hence, Alice will get to know the corresponding cipher text.

---

5) There is a bank ***B*** that allows its customers to withdraw cash from their accounts at hundreds of specialized automated teller machines (ATMs) that are only for cash withdrawals (not for checking balances or performing other transactions). The ATMs

operate in the following way. (In what follows $E_B()$ refers to encryption with the bank's secret key, in a symmetric cryptosystem.) The bank asks the customer ***C*** to select a secret number (called "personal identification number", denoted by PIN (***C***)). Then the bank issues the customer ***C*** a special magnetized card that contains the following two pieces of information (**on separate portions of the magnetized strip on the card**):

(1) The customer's account number at the bank (call it AcNr(***C***)).
(2) E_B(PIN(***C***)).
Each ATM of that bank can perform $E_B(*)$ computation, and also stores a list of all the valid account numbers. It does not store the dollar balance in each account (each ATM limits cash withdrawals to no more than $200 per day for each account, and each account contains at least $500 - the bank automatically closes an account whose balance falls below the $500 minimum).
When the customer ***C*** wants to withdraw cash from an ATM, ***C*** inserts the card and the ATM reads the information on it and then challenges ***C*** to enter PIN (***C***). The ATM then
(1) verifies that the AcN r(***C***) that it reads from the card is on its list of valid account numbers, and then
(2) encrypts (i.e., does $E_B(*)$) what ***C*** just entered and verifies that the result equals to the E_B(PIN(***C***)) that is stored in the card.
If both (1) and (2) are successfully verified, the ATM allows the customer to withdraw the cash (subject to the constraint that the total amount withdrawn by ***C*** that day from that ATM does not exceed $200). The ATM also stores a record of the transaction that consists of the account number and the amount just withdrawn. At midnight every day, all the ATM machines communicate with the bank's main computer. The computer will update all the customer accounts by subtracting from their balances the amounts of cash withdrawn that day. This off-line operation of the ATM allows the customers to quickly withdraw cash even when the network is down or very slow (at peak-hours during the day); contrast this to an on-line operation, which would have required communication with the bank's main computer before a transaction can complete (and would have been problematic if the network was down or very slow at the time of the transaction).
Note that, if the card is stolen from the customer, the thief cannot obtain PIN(***C***) from the card because it is encrypted (this is why it is E_B(PIN(***C***)) rather than PIN(***C***) that is stored on the magnetic strip of the card - **the latter would be insecure because the information on the magnetic strip of a card is easy to read and modify if you have the equipment**).

Please answer the following question:

1) How can a dishonest customer ***M*** (who also has an account of Bank ***B*** and a Card from Bank ***B***) steal money from ***C*** (by withdrawing cash from the account of ***C***). Here we assume that ***M*** knows ***C***'s account number. He also has a machine that can modify information on the magnetic strip. However, ***M*** does not know the secret key of the Bank.

Answer:

Given;

- Bank B provides a card to its customer.
- Card has a magnetic strip.
- In turn the strip have two information pieces in it

    1) PIN encrypted with the banks secret key.
    2) Account Number

    Observing carefully in the above problem statement, there are below flaws

    1) Any unauthorized person has a device which is used like card reader or manipulator, he can read and modify the information present on the magnetic chip.
    2) The PIN is entered by the customer to ATM machine and then the machine checks whether it matches with the existing PIN stored in the magnetic chip.

- The above flaws may lead to various problems for example, if any customer of Bank B who owns the card provided by Bank B, the device/manipulator and if he gets to know the account number of any other customer of the same bank he can just modify the account number in his card to the obtain account number
- The he can go to any ATM Machine ,swipe the card, provide his own pin and he can draw up to the maximum limit which is $200 for a given day from other customers account.
- Moreover, all this he can do without knowing the secret key used by the bank to encrypt the PIN.
- Therefore, we can conclude that any other invalid/unauthorized customer M who has an account of Bank B and a Card from Bank B and if M knows C's account number and also has a device that can modify information on the magnetic strip, then M will change the account number in his card to C's account number by using the device manipulator and go to any ATM Machine swipe the card, provide his own pin and he can draw up to $200 for a given day from C's account.