ITIS 6200/8200 (Fall 2018)
Principles of Information Security & Privacy
Homework 3

Deadline: 11:59pm, November 8, 2018. No late submission will be accepted.
Submission method: upload your solutions to Canvas in one PDF file.

Each student is expected to finish this homework **independently**. No collaboration is allowed. Furthermore, if you search the Internet, copy, and paste an answer, you will get **zero** and can be subject to the academic misconduct procedures and sanctions.

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?

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.

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.)

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.

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.