ITIS6200 EXERCISE: 03

Amaan syed (asyed15@uncc.edu)

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 t he communication line and get s 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 w ill know that this i s the correct one. Th e attacker has 1,000,000 machines, with each machine having the capabilities to try 5,00 0,000 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?**

Key Length = 192 bits

Number of possible keys = 2^192

50% of the possible number of keys = 2^192/2 =2^191

Total number of machines = 1,000,000

Capability of each machine to decrypt = 5,000,000

Total capacity of all machines = 1,000,000 * 5,000,000 = 1 * 10^6 * 5 * 10^6 = 5 * 10^12

Time taken for attacker to try getting 50% of the keys = (2^191) / (5*10^12) seconds

= 1.1972 * 10^45 seconds.

This is approximately equal to 3.8 * 1037 years.

The lifespan of sun is around 5 billion years. 3.8 * 1037 years is much more than 5 billion years so it’s not possible to crack before this.

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.**

A block cipher algorithm works by dividing plain text into equal-sized blocks for processing. In this method, the output size is the same as the input size, meaning the plain text size must match the cipher text size. For successful decryption, the cipher text must retain the same size as the plain text. Block ciphers use specific techniques to transform plain text into cipher text, and decryption requires using the same techniques, provided that the sizes of the cipher text and plain text are identical. Therefore, compressing data after encryption is not an efficient approach.

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, Epub_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 [Epub_Bob(Alice, message)], can M still impersonate Alice? (Here Alice puts her name in the encryption).**

In the first scenario, Alice sends a message to Bob in the form [Alice, Epub Bob(message)], where Epub Bob represents encryption using Bob's public key. An attacker, M, can impersonate Alice by sending a packet with Alice's name in plaintext, potentially revealing the message's origin. Since M has access to Bob's public key, they can send a malicious message encrypted with it [Alice, Epub Bob(malicious message)]. Upon receiving this, Bob might dismiss Alice's legitimate message due to the misleading encrypted content.

In the second scenario, Alice encrypts both her name and the message using Bob's public key and sends it as [Epub Bob(Alice, message)]. Even though Alice secures her identity and the message, attacker M can still pretend to be Alice by encrypting a message along with her name and sending it to Bob. Without additional verification methods, Bob cannot confirm the true origin of the message, leaving him unsure whether it is from Alice or the attacker.

4. **The public key infrastructure (PKI) needs to handle the revocation of compromised keys. Currently, there are two basic approaches. The first one uses the certificate revocation list (CRL). The CRL is published periodically (for example, 8:00am every day). It contains the public key certificates that have been compromised. Another approach is to use Online Certificate Status Protocol (OCSP). Please study how OCSP works. Then write about 0.5 page to discuss the working procedure of OCSP, and the advantages and disadvantages of OCSP over the CRL.'**

   OCSP is used to either revoke certificates or verify their revocation status. A certificate is provided to the client, who then sends an OCSP request to an OCSP responder, including the certificate's serial number. The responder classifies the certificate as Good, Revoked, or Unknown based on its status and responds accordingly.

   The advantages of OCSP over CRL are as follows:

   - OCSP provides real-time certificate status checking, while CRL relies on previously cached data, making OCSP more suitable for applications requiring up-to-date information.
   - OCSP does not require extra RAM to store certificates and statuses like CRL does, resulting in memory savings.
   - OCSP is more efficient in terms of network traffic compared to CRL.

   However, OCSP has some disadvantages compared to CRL:

   - CRL can be faster since it uses cached data for verification.
   - OCSP requires servers to be available for verification, whereas CRL does not.
   - OCSP is vulnerable to certain types of attack responses, posing security risks.

5. **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-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 a forward search attack.

Alice can determine the details of David's actions through the following steps:

1. First, she might speculate on the content of the message she believes David may have sent.
2. She can then encrypt this guessed message and compare it with David's encrypted message to gather clues about the actual message.
3. By repeating this process, she can eventually discover the secret message David sent and compile a complete list of all potential encrypted messages along the way.

6. 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):

- **The customer's account number at the bank (call it AcNr(C)).**
- **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 m ore 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. A t 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 t he 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 stole n 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:**
  - **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.**

Although Customer M can access the bank's secret key, he doesn't actually own it. He possesses a device capable of reading and altering the data on the magnetic stripe of a bank card. M not only knows customer C's account number but also C's identity. Even if M could figure out C's PIN, without the ability to encrypt it, he cannot execute withdrawals. While he can modify his card's account number to C's, the absence of the secret key prevents him from altering the PIN. Consequently, M is unable to produce the necessary encrypted PIN. Since the PIN requires encryption with the bank's secret key and M cannot modify the PIN on the card, he is unable to use the card for transactions.

7. Alice's computer stores the files in the following way: for every file F, the computer will calculate the hash value of the file hash(F) and store it after the file. Every time when Alice login, the machine will automatically hash all the files and compare the results to the stored hash values. In this way, if by accident the hard drive is mis-functioning and flips a few bits in a file, Alice can immediately detect it since the hash value will be different. Now an attacker hacks into Alice's machine and he tries to change several files. The attacker also knows the hash function that the computer uses. Please describe what the attacker needs to do so that the next time Alice login, the machine will not detect the changes. Also, please discuss how we should improve the mechanism to detect such changes.

An attacker can alter files and recalculate their hash values using the same hash function as Alice's computer, replacing the original hashes. This allows the attacker to create a new file, calculate its hash, and update the hash values to match the modified files. As a result, the system fails to detect any discrepancies, verifying the altered files against their hash values successfully. To enhance tampering detection, a keyed hash mechanism can be used, where the hash is computed as H(K, H(K, F)), with K being a secret key. This method prevents the attacker from modifying the hash values since they would lack the secret key needed to generate the correct hashes for the altered or new files.

8. Please draw a binary Merkle's hash tree with 9 leaf nodes. The leaf nodes are labeled as Leaf1 to Leaf9, which correspond to the hash values of the files F1 to F9, respectively. Now please answer:

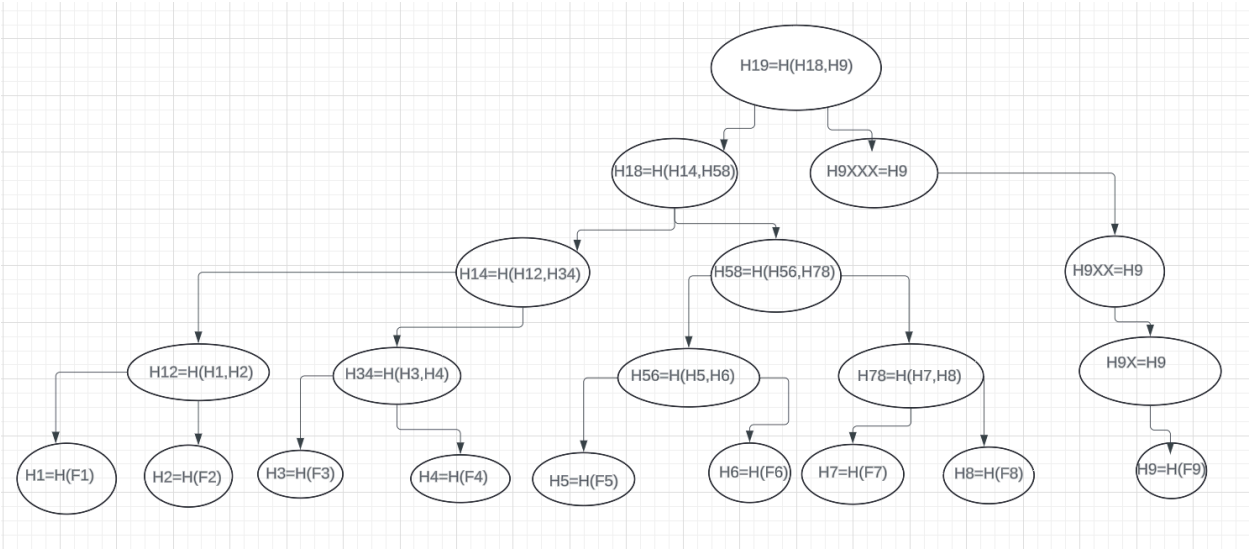
- **For each node in the tree, please label clearly how the hash value is calculated based on its children; Please note that the number of leaf nodes is not power of 2. Therefore, you may need to change the way in which intermediate nodes are calculated. There are different ways to handle this. Please label clearly how you calculate each node.**
- **Now the creator of the file F6 needs to verify that his file's hash value is integrated in the root of the tree. Please show the minimum number of hash values in the tree that the creator needs to accomplish the task. Please also show how the verification is accomplished.**

• Leaf nodes are named file 1 to file 9.

• The hash of each file is represented as H1, H2, H3, ----, and H9 for leaf nodes leaf 1 to • leaf 9.

• The hash of parent leaf nodes 1 & 2 is H1, H2.

• The hash of parent leaf nodes 3 & 4 is H3, H4. And so on



To confirm the inclusion of H6 within H19, one must compute H(H18, H9) to derive H19. H18 is obtained from H14 and H58. H58 is the result of combining H56 with H78, and H56 requires H5. Hence, the creator must acquire H5, H78, H14, and H9. Consequently, to ascertain that H6 is incorporated into the root hash H19, the creator must determine a minimum of four hash values.