CSCE 44303/54203 Homework 2

Release: September 6, 2024

Due: September 13, 2024

Full Grade: 122 pts

Note: If you use handwriting, please write clearly. Unrecognizable writing might cause loss of points.

Problem 1. (6 pts) Classify each of the following as a violation of confidentiality, integrity, or authenticity. Choose the best match for each.

- Paul makes a copy of Linda's private data.
  - Confidentiality
- Gina modifies Roger's homework solution.
  - Integrity
- Henry spoofs Julie's IP address to send a message to a computer.
  - Authenticity

Problem 2. (6 pts) For a cipher, if you use a 16-bit long key, what will be the problem?

- **The main problem will be Inadequate Security as the small key size only makes 65,536 possible combinations which a modern computer can run through fairly quickly.**

Problem 3. (10 pts) One-time pad. (1) In one-time pad, suppose a one-byte message is 10111001, and the key is 10010011. Show the encryption process at the sender and the decryption process at the receiver. (2) Can an attacker in the network (between the sender and the receiver) do something over the ciphertext and cause the receiver to incorrectly decrypt the second bit in the message and, if so, how? (3) If one key is used to encrypt two different messages, what will be the problem?

- 1)
  - Encryption Process
    - Message = 10111001
    - Key = 10010011
    - One-time pad = Bitwise XOR
    - Message XOR key = 00101010 = Cypher after Encryption
  - Decryption Process
    - Cypher = 00101010
    - Key = 10010011
    - Cypher XOR Key = 1011001 = Original Message
- 2)
  - Yes, if the Attacker is able to modify the cypher in transition to the receiver, then when the receiver decrypts the message it will be modified. If the attacker flips the second bit of the cypher the decrypted message will now be 11111001 instead of 10111001.
- 3)
  - Lets say the attacker obtains the universal key and both of the encrypted messages.

- C1 = Cypher Of first Message = M1 (Message one) XOR K(Key)
- C2 = Cypher of Second Message = M2 (Message two) XOR K(Key)
- With one universal K (key) both cyphers can be broken since the attacker has the universal key. If there were multiple keys K1 and K2, then the attacker could only break one of the messages and not both.

Problem 4. (6 pts) Why can't a block cipher have a smaller output (i.e., ciphertext block) size than the input (i.e., plaintext block) size?

- One of the major problems with the block cipher having a smaller output than the input is Loss of information. The encryption process must be invertible, and if the ciphertext has fewer bits of information than the original plaintext then the Key will not work and it will be impossible to recover the data with the Key.

Problem 5. (10 pts) Suppose a message has been encrypted using an old system with the basic DES algorithm under key k. (i) If the receiver of the message is running a new system with 3DES implemented. Can the receiver decrypt the ciphertext by setting k1=k2=k3=k and why? (ii) If the encryption of 3DES used the EEE mode instead of EDE and the decryption used the DDD mode instead of DED, can the receiver decrypt the ciphertext by setting k1=k2=k3=k and why?

- I)
  - Yes, as 3DES follows Encrypt with k1, Decrypt with k2, and Encrypt with K3. If k1=k2=k3=k then the decryption of the second step cancels out the first encrypt stage as the keys are the same. Then the Encrypt with k3 just encrypts to the same state that k1 encrypted to
    - Plain text = 10010101
    - k1=k2=k3=k
    - k1 encrypts Cypher text to 11110011
    - k2 (since it has the same key as k1) decrypts Cypher text back to plain text k2 = 10010101
    - k3 encrypts the text from k2 back into the original cypher k1 created 11110011, leading to the assumption that this is DES with extra steps.
      - (Note this is not the proper way DES actually encrypts data, only an example of the concept)
- II)
  - Over all Yes, if text is encrypted with EEE then decrypted with DDD mode the receiver would be able to decrypt if k1=k2=k3=k. The reasoning is practically the same as the answer provided in I), As it uses the same key to encrypt each iteration, using the same key 3 times to decrypt, DDD, will decrypt the Cipher the same way.

Problem 6. (12 pts) In EDE mode of 3DES, encryption of message m with keys k1, k2 and k3 works as follows: $C=\mathbf{E}_{k1}(\mathbf{D}_{k2}(\mathbf{E}_{k3}(m)))$, where E and D denote the encryption and decryption operation respectively. Given some <plaintext, ciphertext> pairs, how can an attacker find the three keys with effort in the order of 2^{112}? Describe in details. (hint: meet-in-the-middle attack)

1) Encrypt the plaintext m with k3
2) Decrypt cipher C with k1
3) Find the intermediate values $X = E_{k3}(m)$ of all possible values of k3 then store them
4) Do the same for $Y = D_{k1}(c)$ for all possible values of k1 and store them
5) Do $D_{k2}(X)=E_{k2}(Y)$

Problem 7. (15 pts) a. Suppose Alice shares a secret block cipher key, K_{AB} with Bob, and a different secret block cipher key, K_{AC} with Charlie. Describe a method for Alice to encrypt an m-block message such that it can only be decrypted with the cooperation of both Bob and Charlie. The ciphertext should only be a constant size greater than m blocks. For example, if the message size is 2m blocks, it doesn't meet this requirement; if the message size is ~~2m+c~~ m+c blocks where c is a constant, it meets this requirement. You may assume that Bob and Charlie have a pre-established secret channel on which to communicate.

Alice First encrypts the Block message with Ks. Then Generates a random key for Bob Kb. For Charlie, Alice takes (Ks XOR Kb) = Kc for Charlie. Now it requires both keys in order to access the message. Next, Alice encrypts Kb with the pre-shared key Kab and does the same for Kc with Kac.

Cb = Enc(Kab, Kb)

Cc = Enc(Kac, Kc)

Now then Alice transmits the information in the following format

C = (Cm, Cb, Cc)

Where Cm is the cipher text message, Cb is the encrypted share for Bob, and Cc is the encrypted share for Charlie.

b. Now, suppose Alice shares a block cipher key, K_{AB} with Bob, a block cipher key K_{AC} with Charlie, and a block cipher key K_{AD} with David. Describe a method for Alice to encrypt an m-block message such that any two of Bob, Charlie, and David can decrypt (for example, Bob and Charlie can decrypt), but none of them can decrypt the message themselves. Again, the ciphertext should only be a constant size greater than m blocks. (Hint: Pick a random message encryption key to encrypt the message with. Then add three ciphertext blocks to the ciphertext header.)

Like the previous question Alice sends a ciphertext m to Bob and Charlie encrypted with Ka. Now Alice encrypts Ka with a key Ku which is generated by XORing the 2 pre-shared keys Ke = Kab XOR Kac. Now then a third person has joined the Group (David), Alice will generate 3 cipher keys. These being Ke = Kab XOR Kac , Kk = Kab XOR Kad , and Kac XOR Kad. This any of the 2 group members to decrypt the messages.

c. How does your solution from part (b) scale as we increase the number of recipients? In other words, suppose Alice has a secret key with each of n recipients and wants to encrypt so that any k out of n recipients can decrypt, but any k-1 cannot. What would be the length of the header as a function of n and k?

For a Scaled header size will be C(n, k) = n!/k!(n-k)! which allows the header to grow the quickest.

Problem 8. (6 pts) When using block cipher to encrypt a large message, padding is usually needed. Suppose Dr. Smart has designed a padding scheme, which will append a positive number of bytes of value "10101010" to the message until the padded message size is a multiple of the block size. Is this a good padding scheme and why?

No, If the sender decides to Pad 10101010 to the message "multiple of the block size" how will the receiver know what is padding and what is message. For example if the last bytes of the message contain 10101010 then the receiver would not be able to tell what is pad and what is message.

Problem 9. (9 pts) For block cipher modes, under the ECB mode, if ten consecutive message blocks of the same message are the same, will their ciphertext blocks also be the same? How about in the CBC mode? How about in the Counter mode?

a) For ECB which uses the same key to encrypt each message each of the ciphertext blocks will be the same as the same key is used on each
b) For CBC the plaintext is XORed with the cypher text of the previous block, with the initial block being XORed with some vector. This leads the assumption that if ten consecutive blocks of the same message were sent, all of there cipher blocks would be different.
c) For CRT mode a counter is paired with a key to produce a unique key for each block. The plaintext is then XORed with the key to make the ciphertext . Therefore each of the ciphertext blocks on 10 consecutive messages will be different.

Problem 10. (10 pts) For the CBC mode: (1) If one IV value is used to encrypt two messages, what will be the problem? (2) Suppose a message is divided into five message blocks. If the third ciphertext block is modified during transit, which blocks can the receiver correctly decrypt and which not?

1) If the same IV is used to encrypt two different messages identical plaintext blocks will produce the same ciphertext blocks, thereby allowing a hacker to derive which parts of the plaintext are identical.
2) If the 3rd ciphertext block has been modified then the Third and Fourth blocks would be corrupted. This is due to the Third Block already being corrupted, and the Fourth block depending on the Third Block it is corrupted. Leaving the First, Second, and Fifth blocks uncorrupted.

Problem 11. (10 pts) For the Counter mode, suppose a message is divided into five message blocks. (1) If the nonce part of the counter is modified during transit, which blocks can the receiver correctly decrypt and which not? (2) If the third ciphertext block is modified during transit, which blocks can the receiver correctly decrypt and which not?

1) If the Nonce is corrupted, then the keystream generated at encryption will differ than the new keystream generated by the new Nonce.
2) For the cyphertext block 3 being modified is modified then only the 3rd plaintext block will be incorrectly decrypted, the rest will be successfully decrypted.

Problem 12. (10 pts) Suppose public-key cryptography is used to encrypt the communications between Alice and Bob. Alice's public key is e_A, private key is d_A; Bob's public key is e_B, private key is d_B. Now Alice wants to send a message m to Bob. (1) Can Alice encrypt the message using key d_B and why? (2) Can Alice encrypt the message using key d_A and why?

1) No, the reason behind this is the public key eB is meant to be used publicly to encrypt messages intended for Bob and the private key dB is used to decrypt the messages encrypted with eB. If Alice used bobs private key dB to encrypt a message then anyone who has Bob's Public key can decrypt the message.
2) Yes, If Alice encrypts her message with her private then anyone with her Public key can decrypt the message allowing verification that Alice was the original sender.

Problem 13. (12 pts) Alice's RSA parameters are $p = 3; q = 11; e = 3; d = 7$. Bob's RSA parameters are $p = 7; q = 13; e = 5; d = 29$. Suppose Alice wants to send a message m=3 to Bob, and protect the

confidentiality of the message with RSA encryption. Show how Alice will encrypt the message and how Bob will decrypt the message.

1) Encryption
2) $C = m^e \% n$
3) $C = 3^9 \% 91$
4) $C = 245 \% 91$
5) $C = 61$
6) Decryption
7) $m = C^d \% n$
8) $m = 61^{29} \% 91$
9) $5.951 \times 10^{51} \% 91$
10) $M = 3$