COMP90043: Cryptography and security

Workshop-8 Solution

Week 8 Workshop Activity

Part A: MAC, Hash Continued:

1. What is a message authentication code?

It is an authenticator that is a cryptographic function of both the data to be authenticated and a secret key.

2. What types of attacks are addressed by message authentication?

Masquerade: Insertion of messages into the network from a fraudulent source. This includes the creation of messages by an opponent that are purported to come from an authorized entity. Also included are fraudulent acknowledgments of message receipt or non-receipt by someone other than the message recipient.

Content modification: Changes to the contents of a message, including insertion, deletion, transposition, and modification.

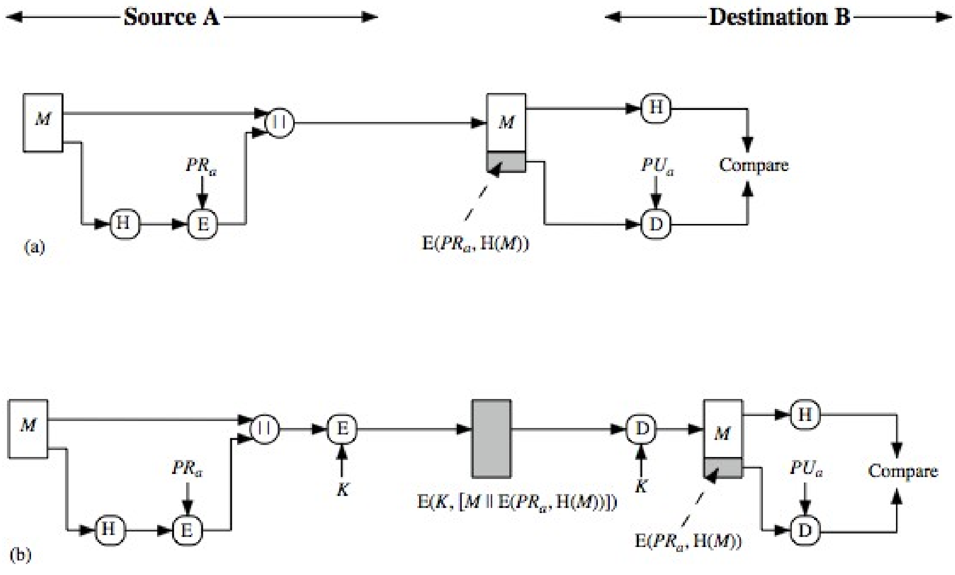
Sequence modification: Any modification to a sequence of messages between parties, including insertion, deletion, and reordering.

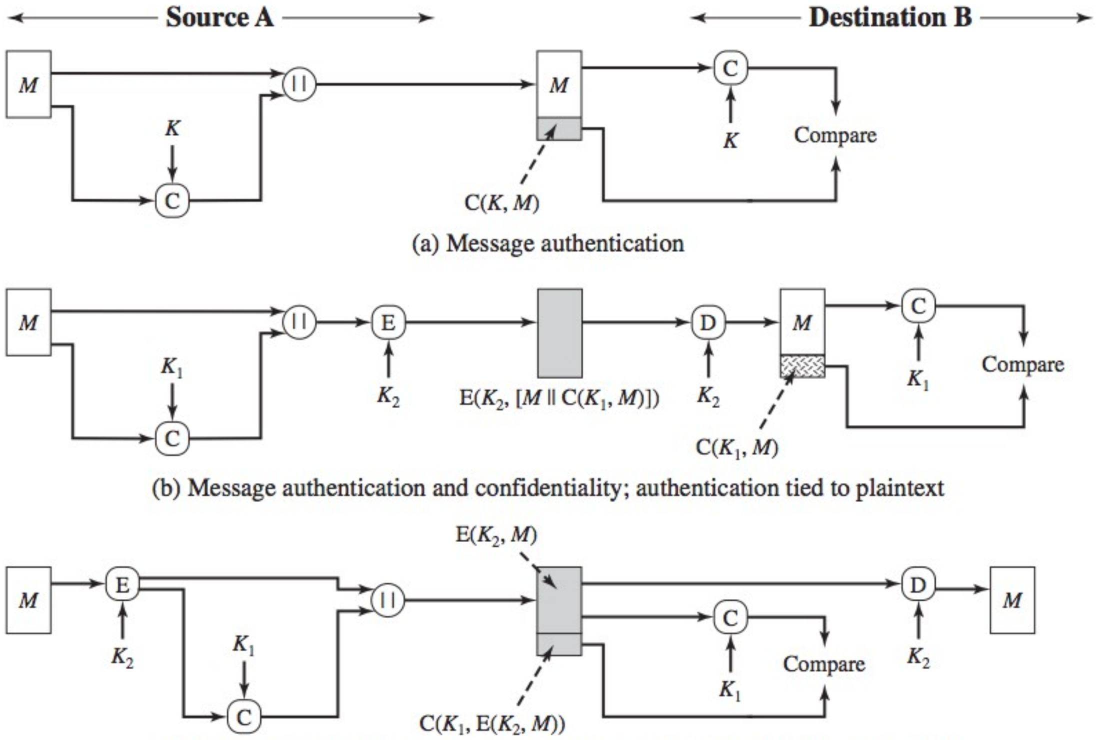
Timing modification: Delay or replay of messages. In a connection-oriented application, an entire session or sequence of messages could be a replay of some previous valid session, or individual messages in the sequence could be delayed or replayed. In a connectionless application, an individual message (e.g., datagram) could be delayed or replayed.

3. What is the main difference between hash functions and Message Authentication codes?

A hash function, by itself, does not provide message authentication. A secret key must be used in some fashion with the hash function to produce authentication. A MAC, by definition, uses a secret key to calculate a code used for authentication.

4. In what ways a hash value can be secured so as to provide message authentication?





(a) Message authentication

(b) Message authentication and confidentiality; authentication tied to plaintext

(c) Message authentication and confidentiality; authentication tied to ciphertext

Figure 12.4 Basic Uses of Message Authentication code (MAC)

5. Discuss two scenarios for using MACs for implementing authentication and confidentiality discussed in lectures?

   Refer to Fig 12.4 attached.

6. List two disputes that can arise in the context of message authentication.

Suppose that John sends an authenticated message to Mary. The following disputes that could arise:

- Mary may forge a different message and claim that it came from John. Mary would simply have to create a message and append an authentication code using the key that John and Mary share.
- John can deny sending the message. Because it is possible for Mary to forge a message, there is no way to prove that John did in fact send the message.

7. What are the properties a digital signature should have?

- It must be able to verify the author and the date and time of the signature.
- It must be able to authenticate the contents at the time of the signature
- The signature must be verifiable by third parties, to resolve disputes.

8. What are some threats associated with a direct digital signature scheme?

- The validity of the scheme depends on the security of the sender's private key. If a sender later wishes to deny sending a particular message, the sender can claim that the private key was lost or stolen and that someone else forged his or her signature.
- Another threat is that some private key might actually be stolen from X at time T. The opponent can then send a message signed with X's signature and stamped with a time before or equal to T.

9. List ways in which secret keys can be distributed to two communicating parties.

For two parties A and B, key distribution can be achieved in a number of ways, as follows:

1. A can select a key and physically deliver it to B.

2. A third party can select the key and physically deliver it to A and B.

3. If A and B have previously and recently used a key, one party can transmit the new key to the other, encrypted using the old key.

4. If A and B each has an encrypted connection to a third party C, C can deliver a key on the encrypted links to A and B.

10. . What is the difference between a session key and a master key?

A session key is a temporary encryption key used between two principals.

A master key is a long-lasting key that is used between a key distribution center and a principal for the purpose of encoding the transmission of session keys. Typically, the master keys are distributed by noncryptographic means.

11.. What is a nonce?

A nonce is a value that is used only once, such as a timestamp, a counter, or a random number; the minimum requirement is that it differs with each transaction.

11. . Explain the problems with key management and how it affects symmetric cryptography?

The primary weakness of symmetric encryption algorithms is keeping the single key secure. Known as key management, it poses a number of significant challenges. If a user wants to send an encrypted message to another using symmetric encryption, he must be sure that she has the key to decrypt the message. How should the first user get the key to the second user? He would not want to send it electronically through the Internet, because that would make it vulnerable to eavesdroppers. Nor can he encrypt the key and send it, because the recipient would need some way to decrypt the key. And if he can even get the get securely to the user, how can be he certain that an attacker has not seen the key on that person's computer? Key management is a significant impediment to using symmetric encryption.

Part B: Symmetric Key Distribution protocol:

Q1 This is a variation of the protocol discussed in the class symmetric key description involving n users and a KDC. Here every user decides to generate random number themselves for the communication they seek to start.

The steps are as follows:

1. A generates a random number R and sends to the KDC his name A, destination B, and E(Ka, R).

2. KDC responds by sending E(Kb, R) to A.

3. A sends E(R, M) together with E(Kb, R) to B.

4. B knows Kb, thus decrypts E(Kb, R), to get R and will subsequently use R to decrypt

E(R, M) to get M.

Is this secure?

PS: Assume all other assumptions made in the protocol. All users share a master key with KDC, all communications can be observed by the users.

Solution:

i) sending to the server the source name A, the destination name Z (his own), and E(*Ka*, *R*), as if A wanted to send him the same message encrypted under the same key R as A did it with B
ii) The server will respond by sending E(*Kz*, *R*) to A and Z will intercept that
iii) because Z knows his key *Kz*, he can decrypt E(*Kz*, *R*), thus getting his hands on R that can be used to decrypt E(*R*, *M*) and obtain *M*.

Q2.

Consider the following protocol, designed to let A and B decide on a fresh, shared session key K=AB. We assume that they already share a long-term key K_{AB}.

1. A → B: A, N_{A}.

2. B → A:E(K_{AB}, [N_{A}, K'_{AB}])

3. A → B:E(K'_{AB}, N_{A})

a. We first try to understand the protocol designer's reasoning:

— Why would A and B believe after the protocol ran that they share K'_{AB} with the

other party?

—Why would they believe that this shared key K'_{AB} is fresh?

In both cases, you should explain both the reasons of both A and B, so your answer should complete the sentences

A believes that she shares K'_{AB} with B since…

B believes that he shares K'_{AB} with A since…

A believes that K'_{AB} is fresh since…

B believes that K'_{AB} is fresh since…

b. Assume now that A starts a run of this protocol with B. However, the connection is intercepted by the adversary C. Show how C can start a new run of the protocol using reflection, causing A to believe that she has agreed on a fresh key with B (in spite of the fact that she has only been communicating with C). Thus, in particular, the belief in (a) is false.

c. Propose a modification of the protocol that prevents this attack.

Solution: a. A believes that she shares K'_{AB} with B since her nonce came back in message 2 encrypted with a key known only to B (and A). B believes that he shares K'_{AB} with A since *NA* was encrypted with K'_{AB}, which could only be retrieved from message 2 by someone who knows K'_{AB} (and this is known only by A and B). A believes that *K*'AB is fresh since it is included in message 2 together with *NA* (and hence message 2 must have been constructed after message 1 was sent). B believes (indeed, knows) that K'_{AB} is fresh since he chose it himself.

b. B. We consider the following interleaved runs of the protocol:

1. A →C(B) : A, *NA*

1`. C(B) →A : B, *NA*

2`. A →C(B) : E(*KAB*, [*NA*, K'_{AB}])

2. C(B) →A : E(*KAB*, [*NA*, K'_{AB})

3. A →C(B) : E(*K'AB*, *NA*)

C cannot encrypt A's nonce, so he needs to get help with message 2. He therefore starts a new run with A, letting A do the encryption and reflecting the reply back. A will accept the unprimed protocol run and believe that B is present.

c. To prevent the attack, we need to be more explicit in the messages, e.g. by changing message 2 to include the sender and receiver (in this order), i.e. to be E(*KAB*, [A, B, *NA*, *K'AB*]).