COMP90043 Cryptography and Security
Semester 2, 2020, Workshop Week 9 Solutions

Message authentication

1. 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.

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

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

   (b) A third party can select the key and physically deliver it to A and B.

   (c) 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.

   (d) 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.

3. 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.

4. 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.

Symmetric Key Distribution Protocol

1. Consider a variation of the symmetric key distribution protocol discussed in the lecture involving n users and a KDC. Here every user decides to generate random number themselves for the communication they seek to start. All users share a master key with the KDC, all communications can be observed by all users.
   The steps are as follows:

(a) A generates a random session key K_s and sends to the KDC his identity ID_A, destination ID_B, and $E(K_A, K_s)$.

(b) KDC responds by sending $E(K_B, K_s)$ to A.

(c) A sends $E(K_s, M)$ together with $E(K_B, K_s)$ to B.

(d) B knows K_B, thus decrypts $E(K_B, K_s)$, to get K_s and will subsequently use K_s to decrypt $E(K_s, M)$ to get M.

Is this secure?

It's not secure. Consider the following steps:
An attacker Z could send to the server the source identity ID_A, the destination ID_Z (his own), and $E(K_A, K_s)$, as if A wanted to send Z a message encrypted under the same key K_s as A did with B.
The server will respond by sending $E(K_Z, K_s)$ to A which could be intercepted by Z.
Because Z knows his own key K_Z, he can decrypt $E(K_Z, K_s)$, thus getting his hands on K_s that can be used to decrypt $E(K_s, M)$ and obtain M.

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

$$\begin{aligned} A \to B &: ID_A, N_A \\ B \to A &: E(K_{AB}, [N_A, K_s]) \\ A \to B &: E(K_s, N_A) \end{aligned}$$

(a) Why would A and B believe after the protocol ran that they share K_s with each other?

A believes that she shares K_s 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_s with A since N_A was encrypted with K_s, which could only be retrieved from message 2 by someone who knows K_{AB} (and this is known only by A and B).

(b) Why would they believe that this shared key K_s is fresh?

A believes that K_s is fresh since it is included in message 2 together with N_A (and hence message 2 must have been constructed after message 1 was sent).
B believes (indeed, knows) that K_s is fresh since He chose it himself.

(c) 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 he has only been communicating with C). Thus, in particular, the belief in (a) is false.

Consider the following interleaved runs of the protocol:

$$
\begin{aligned}
A \to C :& ID_A, N_A \\
& C \to A : ID_B, N_A \\
& A \to C : E(K_{AB}, [N_A, K_s]) \\
C \to A :& E(K_{AB}, [N_A, K_s]) \\
A \to C :& E(K_s, N_A)
\end{aligned}
$$

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. Note that C cannot decrypt any further message from A, nor sending any message to A, but A will accept the unprimed protocol run and believe that B is present.

(d) Propose a modification of the protocol that prevents this attack.

To prevent the attack, we need to be more explicit in the messages. For example, by changing message 2 to include both the sender and receiver: $E(K_{AB}, [ID_A, ID_B, N_A, K_s])$.