Homework7: Security

COM-208: Computer Networks

With respect to a message m, we define the following security properties:

- **Confidentiality** holds iff, additionally to the sender of m, only the legitimate recipient of m gets to learn the content of m.
- **Authenticity** holds iff the legitimate recipient of m can verify that no third party assumes the identity of the legitimate sender of m.
- **Integrity** holds iff the legitimate recipient of m can verify that no third party has modified m.

Providing authenticity

End-systems Alice and Bob share a secret key K and use the following protocol such that Alice authenticates herself to Bob:

S1. Alice tells Bob that she has something to say.

S2. Bob sends Alice a nonce R.

S3. Alice sends Bob $m, K\{R, m\}$, i.e., her message plus the nonce and message encrypted with their shared secret key.

Questions:

1. (*Basic*) Consider an adversary, Persa, who is sitting on the communication channel between Alice and Bob. Does this protocol prevent impersonation and enable key reuse?
2. (*Advanced*) Consider an adversary, Trudy, who is *not* on the communication channel between Alice and Bob (she is just another end-system somewhere on the Internet). And suppose that while Alice is authenticating herself to Bob, Bob uses the same protocol to authenticate himself to Alice. Can you think how Trudy can take advantage of this situation and impersonate Alice to Bob?

Hint: Consider that, while Alice and Bob authenticate themselves to each other, their messages can be arbitrarily interleaved. Also, consider that, when Alice authenticates herself to Bob, it is Bob that chooses the nonce, and vice versa.

Routing attack

Intermediate

A router's link-state message includes a list of its directly connected neighbors and the direct costs to these neighbors. Once a router receives link-state messages from all of the other routers, it can create a complete map of the network, run its least-cost algorithm, and configure its forwarding table. One relatively easy attack on the routing algorithm is for the attacker, after intercepting the routers' messages, to arbitrarily modify them so as to distribute bogus link-state messages with incorrect link-state information. How can this be prevented?

Providing all three properties

Basic

A and B are end-systems. M is a malicious adversary sitting on the communication channel between them. A wants to send message m to B. M wants to read and/or modify message m. A knows this and tries to send message m to B in a way that guarantees confidentiality, authenticity, and data integrity.

For each of the following cases, identify if confidentiality and/or authenticity and/or data integrity hold, and justify your answer.

1. A sends: $K_s\{m\}$
2. A sends: $m,\ K_s\{m\}$
3. A sends: $K_A^+\{m\}$
4. A sends: $m,\ K_B^+\{m\}$
5. A sends: $K_A^-\{K_B^+\{m\}\}$

where:

- K_s is a symmetric key, shared only between A and B
- $K_A^+,\ K_A^-$ is A's public and private key, respectively
- $K_B^+,\ K_B^-$ is B's public and private key, respectively

With security fixes

End-systems A and B want to secure their communication against an adversary sitting on the communication channel between them. In each of the following scenarios:

i Identify an existing security problem or weakness and, if applicable, describe an attack that exploits it.

ii Provide a solution that fixes the weakness (i.e. what should A send instead). Make sure to provide enough detail for your solution to be understandable. For example, if you say "A should use a MAC for authentication", but you do not explain how the MAC should be computed, then your answer is not complete.

Scenarios:

1. (*Basic*) A wants to send one message m to B, ensuring confidentiality and authenticity. For this, A sends: $H\,(K, m)$.

2. (*Basic*) A wants to send one message m to B, ensuring confidentiality and authenticity. For this, A sends: $K\{m\}$.

3. (*Intermediate*) A wants to send one message m to B, ensuring confidentiality and authenticity. A knows B's true public key K_B^+. A sends: $K_B^+\left\{m\ ,\ K_B^+\{H(m)\}\right\}$.

4. (*Intermediate*) A is sending a number of messages to B. Whenever B receives a message m, B should be able to verify that A indeed sent a message with the same content as m at least once. For this, A appends $H(K)$ to each message it sends.

5. (*Advanced*) A wants to send one message m to B, ensuring authenticity and data integrity. For this, A cuts m into two pieces, m_1 and m_2, sends first $m_1, H(K)$, then $m_2, H(K)$.

6. (*Advanced*) A and B are friends that want to have a sensitive online conversation, ensuring confidentiality, authenticity, and data integrity (A receives exactly the sequence of messages sent by B and vice versa). For this, they use SSL as described in class: B sends a nonce and a certificate with its public key to A, A sends a nonce and a master key to B (encrypted with B's public key), and from the master key and the nonces, they both derive other keys that they use for confidentiality and authenticity.

   Hint: In the example we saw in class, B was an online store. In this question, A and B are friends having a sensitive conversation.

where:

- H is a cryptographic hash function that is known to everyone.
- K is a symmetric key, shared only between A and B.
- K_B^+ is B's public key.

The role of sequence numbers

End-systems A and B secure their communication with the following protocol:

S1: A sends a "hello" message with a nonce n_A and a certificate containing her public key (K_A^+)

S2: B responds with a nonce n_B and a certificate containing her public key (K_B^+)

S3: After this exchange, to communicate a message m_i, A sends: $K_B^+ \left\{ m_i, K_A^- \left\{ H\left(n_B, m_i\right) \right\} \right\}$

S4: Similarly, to communicate a message m_j, B sends: $K_A^+ \left\{ m_j, K_B^- \left\{ H\left(n_A, m_j\right) \right\} \right\}$,

where K_A^-, K_B^- are A's and B's private keys and H is a globally known cryptographic hash function.

Questions:

1. (*Basic*) Does this protocol guarantee confidentiality?
2. (*Intermediate*) Assume a man-in-the-middle, who records all the messages sent by A, and, when the communication between A and B ends, she resends them to B, trying to impersonate A. Will her attack be successful (or not) and why?
3. (*Advanced*) Is this protocol vulnerable to any attack(s) other than the ones described above? If yes, briefly describe the attack(s) and provide the changes that make the protocol completely secure. If necessary, you can add new steps in the protocol, but do not modify the existing ones.

Out-of-order delivery atop TCP

Intermediate

In class, we said that SSL relies on sequence numbers to prevent an attacker from changing the order of messages (it was the scenario where Alice communicates with an online store, and we want to prevent an attacker sitting on the communication channel between them from swapping Alice's requests to the store). Why do we need this technique? Won't TCP ensure that the online store receives Alice's messages in the correct order?