Weaver
Spring 2022

CS 161 Computer Security

Discussion 5

Question 1 ***Key Exchange Protocols*** ()

Recall that in a Diffie-Hellman key exchange, there are values a, b, g and p. Alice computes $g^a \bmod p$ and Bob computes $g^b \bmod p$.

Q1.1 Which of these values (a, b, g, and p) are publicly known and which must be kept private?

Solution:

g and p are publicly known. Implementations of Diffie-Hellman often have carefully picked values of g and p which are known to everyone. Alice and Bob must keep a and b secret respectively.

Q1.2 Mallory can eavesdrop, intercept, and modify everything sent between Alice and Bob. Alice and Bob perform Diffie-Hellman to agree on a shared symmetric key K. After the exchange, Bob gets the feeling something went wrong and calls Alice. He compares his value of K to Alice's and realizes that they are different. Explain what Mallory has done.

Solution:

Mallory is performing a **man-in-the-middle attack**. Mallory pretends to be Bob when she talks to Alice, and Mallory also pretends to be Alice when she talks to Bob. In this way, both Alice and Bob are unknowingly talking to Mallory. Mallory can then decrypt/re-encrypt the traffic in both directions and modify it however she wishes to.

More technically, when Alice sends $A = g^a \bmod p$ to Bob, Mallory intercepts this (preventing it from going to Bob), and sends back to Alice: $M = g^c \bmod p$. Now when Alice sends a message to Bob, she uses $K_{bad} = M^a \bmod p$ which Mallory knows as $K_{bad} = A^c \bmod p$. Mallory can then decrypt all messages sent from Alice. She can also send messages to Alice which Alice thinks are from Bob. Mallory then does the same trick to Bob.

Now consider the following key exchange protocols which can be used by Alice (A) and Bob (B) to agree upon a shared key, K.

ElGamal-Based Key Exchange

Message 1	$A \rightarrow B$:	$\{K\}_{PK_B}$
	Key exchanged	
Message 2	$A \leftarrow B$:	$\{secret1\}_K$
Message 3	$A \rightarrow B$:	$\{secret2\}_K$

Diffie-Hellman Key Exchange

Message 1	$A \rightarrow B$:	$g^a \mod p$
Message 2	$A \leftarrow B$:	$g^b \mod p$
	Key exchanged $K = g^{ab} \mod p$	
Message 3	$A \leftarrow B$:	$\{secret1\}_K$
Message 4	$A \rightarrow B$:	$\{secret2\}_K$

Some additional details:

- PK_B is Bob's long-lived public key.
- K, the Diffie-Hellman exponents a and b, and the messages themselves are destroyed once all messages are sent. That is, these values are not stored on Alice and Bob's devices after they are done communicating.

Eavesdropper Eve records all communications between Alice and Bob, but is unable to decrypt them. At some point in the future, Eve is lucky and manages to compromise Bob's computer.

Q1.1 Is the confidentiality of Alice and Bob's prior ElGamal-based communication in jeopardy?

Solution: Yes. The compromise of Bob's computer gives Eve access to Bob's private key, allowing Eve to decrypt the traffic she previously recorded that was encrypted using Bob's public key. Once decrypted, she obtains K, and can then apply it to decrypt the traffic encrypted using symmetric key encryption.

Q1.2 What about Alice and Bob's Diffie-Hellman-based communication?

Solution: No. Since Alice and Bob destroy the DH exponents a and b after use, and since the key computed from them itself is never transmitted, there is no information present on Bob's computer that Eve can leverage to recover K. This means that with Diffie-Hellman key exchanges, later compromises in no way harm the confidentiality of previous communication, even if the ciphertext for that communication was recorded in full. This property is called *Perfect Forward Secrecy.*

Question 2 ***Public Key Encryption*** ()

The El Gamal encryption scheme is reproduced below:

- **Key Generation**: public key $= (g, h, p)$, where $h = g^k \pmod p$, private key $= k$
- **Encryption**: $c = (c_1, c_2) = (g^r \bmod p, m \times h^r \bmod p)$, where r is randomly sampled from $\{1, \ldots, p-1\}$.
- **Decryption**: $m = c_1^{-k} \times c_2 \pmod p$

Look at each scenario below and select the appropriate options.

Q2.1 With El Gamal, it is not a problem if the adversary can learn the value of g somehow.

[✓ (A) True]
(B) False
(C) —
(D) —
(E) —
(F) —

Solution: g is part of the public key, so it is fine for it to be known to the public (including the adversary).

Q2.2 With El Gamal, it is not a problem if the value r used during encryption is accidentally revealed after the encryption is complete.

(G) True
[✓ (H) False]
(I) —
(J) —
(K) —
(L) —

Solution: If the adversary learns r, they can compute $c_2 h^{-r} \bmod p$, and that will reveal the message m.

Now imagine that Alice (A) and Bob (B) want to communicate over an insecure network and they know each other's public key. Consider the following message exchange:

A: Hey Bob, it's Alice. How many dollars do I owe you?
B: 10000

The message is encrypted with Alice's public key using ElGamal encryption.
Alice decrypted this successfully, but suddently remembered that she only owed Bob \$100.

Q2.1 Assume Bob would not lie. How did an attacker tamper with the message?

Solution: The attacker multiplied c_2 by 100, or multiplied $c_1 \cdot c'_1, c_2 \cdot c'_2$ where c' is a valid encryption of 100, or they encrypted an entirely new message.

Q2.2 What could Bob have additionally sent that would've stopped this attack?

Solution: Bob could attach a signature to his original message.

Question 3 ***Why do RSA signatures need a hash?*** **()**

To generate RSA signatures, Alice first creates a standard RSA key pair: (n, e) is the RSA public key and d is the RSA private key, where n is the RSA modulus. For standard RSA signatures, we typically set e to a small prime value such as 3; for this problem, let $e = 3$.

Suppose we used a **simplified** scheme for RSA signatures that skips using a hash function and instead uses message M directly, so the signature S on a message M is $S = M^d \bmod n$. In other words, if Alice wants to send a signed message to Bob, she will send (M, S) to Bob where $S = M^d \bmod n$ is computed using her private signing key d.

Q3.1 With this **simplified** RSA scheme, how can Bob verify whether S is a valid signature on message M? In other words, what equation should he check, to confirm whether M was validly signed by Alice?

Solution: $S^3 = M \bmod n$.

Q3.2 Mallory learns that Alice and Bob are using the **simplified** signature scheme described above and decides to trick Bob into beliving that one of Mallory's messages is from Alice. Explain how Mallory can find an (M, S) pair such that S will be a valid signature on M.

You should assume that Mallory knows Alice's public key n, but not Alice's private key d. The message M does not have to be chosen in advance and can be gibberish.

Solution: Mallory should choose some random value to be S and then compute $S^3 \bmod n$ to find the corresponding M value. This (M, S) pair will satisfy the equation in part (a).

Alternative solution: Choose $M = 1$ and $S = 1$. This will satisfy the equation.

Q3.3 Is the attack in Q3.2 possible against the **standard** RSA signature scheme (the one that includes the cryptographic hash function)? Why or why not?

Solution: This attack is not possible. A hash function is one way, so the attack in part (b) won't work: we can pick a random S and cube it, but then we'd need to find some message M such that $H(M)$ is equal to this value, and that's not possible since H is one-way.

Comment: This is why the real RSA signature scheme includes a hash function: exactly to prevent the attack you've seen in this question.