Nick Weaver
Fall 2018

CS 161
Computer Security

Discussion 5

Week of September 24, 2018

Question 1 ***Diffie–Hellman key exchange*** **(15 min)**

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 $g^b \bmod p$.

(a) Which of these values 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.

(b) Eve can eavesdrop on everything sent between Alice and Bob, but can't change anything. Alice and Bob run Diffie-Hellman and have agreed on a shared symmetric key K. However, Bob accidentally sent his b to Alice in plain text. If Eve viewed all traffic since the beginning of the exchange, can she figure out what K is?

Solution:

Yes, this will be very easy for Eve: she can use the value $A = g^a \bmod p$ which Alice sent to calculate $K = A^b \bmod p$.

(c) Mallory can not only view all Alice—Bob communications but also intercept and modify it. 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 and what she can now do.

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.

Question 2 ***Perfect Forward Secrecy*** **(15 min)**

Alice (A) and Bob (B) want to communicate using some shared symmetric key encryption scheme. Consider the following key exchange protocols which can be used by A and B to agree upon a shared key, K_{ab}.

El Gamal-Based Key Exchange Protocol

Message 1	$A \rightarrow B$:	$\{K_{ab}\}_{K_B^{pub}}$
	Key exchanged	
Message 2	$A \leftarrow B$:	$\{secret1\}_{K_{ab}}$
Message 3	$A \rightarrow B$:	$\{secret2\}_{K_{ab}}$

Diffe-Hellman Key Exchange

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

Some additional details:

- K_B^{pub} is Bob's long-lived public key.
- All messages are destroyed immediately after reading them.
- K_{ab} and DH exponents a and b are destroyed once all messages are sent.

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.

(a) Is the confidentiality of Alice and Bob's prior El Gamal-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_{ab}, and can then apply it to decrypt the traffic encrypted using symmetric key encryption.

(b) What about Alice and Bob's Diffe-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_{ab}. 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 3 ***Why do RSA signatures need a hash?*** **(20 min)**

This question explores the design of standard RSA signatures in more depth. 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 $e = 3$.

To generate a standard RSA signature S on a message M, Alice computes $S = H(M)^d \bmod n$. If Bob wants to verify whether S is a valid signature on message M, he simply checks whether $S^3 = H(M) \bmod n$ holds. d is a private key known only to Alice and $(n, 3)$ is a publicly known verification key that anyone can use to check if a message was signed using Alice's private signing key.

Suppose RSA signatures skipped using a hash function and just used 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.

(a) 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, S was validly signed by Alice?

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

(b) Mallory learns that Alice and Bob are using the simplified (hash-less) signature scheme described above and decides to trick Bob. Mallory wants to send some (M, S) to Bob that Bob will think is from Alice, even though Mallory doesn't know the private key. Explain how Mallory can find M, S 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. She can choose both M and S freely. 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.

(c) Is the attack in part (b) possible against the real RSA signature scheme (the one that includes the cryptographic hash function)? Why or why not? Assume the cryptographic hash function can be treated as a random oracle. (Do not worry if you do not know what this means).

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.