German University in Cairo
Faculty of MET (CSEN 1001 Computer and Network Security Course)
Dr. Amr El Mougy
Reham Ayman
Abdelrahman Osama

Practice Assignment 4

Discussion 01/03/2020 - 04/03/2020

1 Diffie-Hellman

1. Users A and B use the Diffie-Hellman key exchange technique with a common prime $q = 71$ and a primitive root $a = 7$.

   (i) If user A has private key $X_A = 5$, what is A's public key Y_A?

   (ii) If user B has private key $X_B = 12$, what is B's public key Y_B?

   (iii) What is the shared secret key?

   Solution:

   (i) $Y_A = a^{X_A} \bmod q = 7^5 \bmod 71 = 51$

   (ii) $Y_B = a^{X_B} \bmod q = 7^{12} \bmod 71 = 4$

   (iii) $K = {Y_B}^{X_A} \bmod q = 4^5 \bmod 71 = 30$

2. This problem illustrates the point that the Diffie-Hellman protocol is not secure without the step where you take the modulus; i.e. the "Indiscrete Log Problem" is not a hard problem! You are Eve and have captured Alice and Bob and imprisoned them. You overhear the following dialog.
   Bob: Oh, let's not bother with the prime in the Diffie-Hellman protocol, it will make things easier.
   Alice: Okay, but we still need a base a to raise things to. How about $a = 3$?
   Bob: All right, then my result is 27.
   Alice: And mine is 243.
   What is Bob's secret X_B and Alice's secret X_A? What is their secret combined key?

   Solution: Since there is no modulus operation involved here. This simply reduces to getting the *log* directly. So Bob's secret can be computed as follows:

   $$X_B = \log_a Y_B = \log_3 27 = 3$$

   Similarly for Alice's:

   $$X_A = \log_a Y_A = \log_3 243 = 5$$

   Then the secret key K is:

   $${Y_B}^{X_A} = {Y_A}^{X_B} = 27^5 = 243^3 = 14348907$$

3. Alice and Bob wish to exchange a secret key using the Diffie-Hellman algorithm. They agree to use the prime number 71 and its primitive root 7. Alice chooses the private key 6 while Bob chooses 12. To their misfortune, Eve, intercepts the public keys sent by Alice and Bob and executes a Man In The Middle (or rather Woman In the Middle) attack to trick them into believing they have exchanged a key with each other, while in reality they would have exchanged two keys with Eve. Execute Eve's attack and obtain the secret keys shared with Alice and Bob. You may assume Eve chooses the private keys 8 and 14.

Solution: Having $q = 71, a = 7, X_A = 6$ and $X_B = 12$, we calculate the public keys of both Alice and Bob as follows;

$$Y_A = a^{X_A} \bmod q = 7^6 \bmod 71 = 2$$
$$Y_B = a^{X_B} \bmod q = 7^{12} \bmod 71 = (7^{11} \bmod 71 * 7 \bmod 71) \bmod 71 = (31 * 7) \bmod 71 = 4$$

We can also calculate Eve's public key with Alice and Eve's public key with Bob knowing that her private key with Alice is $X_C = 8$ and her private key with Bob is $X_D = 14$;

$$Y_C = a^{X_C} \bmod q = 7^8 \bmod 71 = 27$$
$$Y_D = a^{X_D} \bmod q = 7^{14} \bmod 71 = (7^{11} \bmod 71 * 7^3 \bmod 71) \bmod 71 = (31 * 59) \bmod 71 = 54$$

We now can use these public keys to calculate Eve's common key with Alice K_{AC} and her common key with Bob K_{BD};

$$K_{AC} = {Y_A}^{X_C} \bmod q = 2^8 \bmod 71 = 43$$
$$K_{BD} = {Y_B}^{X_D} \bmod q = 4^{14} \bmod 71 = 5$$

4. In the man-in-the-middle attack on the Diffie-Hellman key exchange protocol the adversary generates two public−private key pairs for the attack. Could the same attack be accomplished with one pair? Explain.

Solution: Yes, it is possible. Consider the following scenario. A and B want to share a secret key using the Diffie-Hellman protocol, while *Eve* is the attacker. q and a are known as usual to the three entities.

1. *Eve* chooses X_E and generates $Y_E = a^{X_E} \bmod q$
2. A chooses X_A and generates $Y_A = a^{X_A} \bmod q$
3. B chooses X_B and generates $Y_B = a^{X_B} \bmod q$
4. A sends B his public key Y_A. However, *Eve* intercepts the message and sends B her public key Y_E instead.

5. B receives Y_E and computes the presumed common key with A; $K_{BE} = {Y_E}^{X_B} \bmod q$

6. B now sends A his public key Y_B. However, *Eve* intercepts the message and sends A her public key Y_E instead.

7. A receives Y_E and computes the presumed common key with B; $K_{AE} = {Y_E}^{X_A} \bmod q$

8. *Eve* has both public keys for A and B; Y_A and Y_B respectively. So she computes both shared keys $K_{AE} = {Y_A}^{X_E} \bmod q$ and $K_{BE} = {Y_B}^{X_E} \bmod q$

Now K_{AE} is the common shared key between *Eve* and A, while K_{BE} is the common shared key between *Eve* and B.

2 Hash Functions

Hash functions are mainly used to check message integrity. Any function won't provide integrity unless it has some requirements. Here are the requirements for hash functions:

1. Can be applied to any sized message M
2. Produces fixed-length output h
3. Easy to compute $h = H(M)$ for any message M
4. Given h is infeasible to find x s.t. $H(x) = h$ **one-way property**
5. Given x is infeasible to find y s.t. $H(y) = H(x)$ **weak collision resistance**
6. Infeasible to find any x, y s.t. $H(y) = H(x)$ **strong collision resistance**

2.1 Questions

1. (i) Consider the following hash function. Messages are in the form of a sequence of decimal numbers, $M = (a_1, a_2, \cdots, a_t)$. The hash value h is calculated as $(\sum_{i=1}^{t} a_i) \bmod n$, for some predefined value n. Does this hash function satisfy the requirements for a hash function? Explain your answer.

   (ii) Repeat part (i) for the hash function $h = (\sum_{i=1}^{t} (a_i)^2) \bmod n$

   (iii) Calculate the hash function of part (ii) for $M = (189, 632, 900, 722, 349)$ and $n = 989$.

Solution:

(i) It satisfies properties 1 through 3 but not the remaining properties. For example, for property 4, a message consisting of the value h satisfies $H(h) = h$. For property 5, take any message M and add the decimal digit 0 to the sequence; it will have the same hash value.

(ii) It satisfies properties 1 through 3. Property 4 is not satisfied since the attacker can generate a message to give the same value of the summation $(\sum_{i=1}^{t}(a_i)^2)$. Properties 5 and 6 are not satisfied because $-M$ will have the same value as M.

(iii) 955

2. Using an encryption algorithm to construct a one-way hash function. Consider using RSA with a known key. Then process a message consisting of a sequence of blocks as follows: Encrypt the first block, XOR the result with the second block and encrypt again, and so on. Show that this scheme is not secure by solving the following problem. Given a two-block message $B1$, $B2$, and its hash

$$RSAH(B1, B2) = RSA(RSA(B1) \oplus B2)$$

and given an arbitrary block $C1$, choose $C2$ so that $RSAH(C1, C2) = RSAH(B1, B2)$. Thus, the hash function does not satisfy weak collision resistance.

Solution: The goal is to find $C2$ such that $RSAH(C1, C2) = RSAH(B1, B2)$.

$$\text{Since } RSAH(C1, C2) = RSA(RSA(C1) \oplus C2)$$

$$\text{and } RSAH(B1, B2) = RSA(RSA(B1) \oplus B2)$$

Given that the key, $B1$, $B2$, and $C1$ are known. So $C2$ could be the following:

$$C2 = RSA(C1) \oplus RSA(B1) \oplus B2$$

This will make

$$RSAH(C1, C2) = RSA(RSA(C1) \oplus RSA(C1) \oplus RSA(B1) \oplus B2)$$

Knowing the facts that $X \oplus X = 0$ and $X \oplus 0 = X$. So

$$RSAH(C1, C2) = RSA(RSA(B1) \oplus B2) = RSAH(B1, B2)$$

3. A and B want to verify that they possess a common key K, using a public one-way function h. The verification protocol works as follows:

   1. A sends $h(h(K))$ to B.
   2. B verifies that the received value is correct.
   3. B sends $h(K)$ to A.
   4. A verifies that the received value is correct.

   Given the verification protocol, answer the following questions:

   (i) Why not have A send $h(K)$ to B and then have B send $h(h(K))$ to A?

(ii) What keeps C from intercepting A's transmission of $h(h(K))$ and then sending $h(K)$ back to A (assuming C doesn't know K)?

Solution:

(i) If A sends $h(K)$ to B, then anyone intercepts this message can play the role of B. The reason is that the hash function used is a public hash function. Then anyone intercepts $h(K)$ can give it as input to the hash function and get $h(h(K))$ and send it back to A as if it was B.

(ii) Since C doesn't know K. And it is given that h the hash function used is one-way. So it is infeasible to get $h(K)$ out of $h(h(K))$.

3 Security Requirements Question 1

For this problem, assume that Alice wants to send a **single message M** to Bob. To do so, Alice and Bob can potentially use a number of different approaches and cryptographic technologies, which we will describe using the following terminology:

M	Plaintext for a single message
s_k	Symmetric cryptography key
AES_{s_k}	Symmetric-key encryption using CBC mode, with the key s_k
SHA_{256}	SHA-256 hash function
$AES\text{-}EMAC_{s_k}$	Keyed MAC function with symmetric key s_k
K_A	Alice's public key
${K_A}^{-1}$	Alice's corresponding private key
K_B	Bob's public key
${K_B}^{-1}$	Bob's corresponding private key
E_K	Public-key encryption with the key K
$Sign_{k^{-1}}$	Digital signature using the private key

You can assume that the public keys have been securely distributed, so Alice and Bob know their correct values. Symmetric keys have not been exchanged.

Consider the following properties that Alice and Bob might desire their communication to have: **Confidentiality**, **Integrity**, and **Non-Repudiation**.

For each of the following possible communication approaches, Mention (and explain why) which of these properties will securely hold (or not hold) in the presence of Mallory, a Man In The Middle (MITM) attacker.

Mention **None** if none of the properties hold. If an approach fails entirely (will not result in Bob being able to read a given message M), mention **Broken**.

Note that || denotes concatenation.

1. Alice generates a new symmetric key s_k and sends to Bob:

$$E_{K_A}(s_k)||AES_{s_k}(M)$$

Solution: Broken. Only Alice can decrypt using her private key and obtain the symmetric key.

2. Alice sends to Bob:

$$E_{K_B}(M)||Sign_{k_A{}^{-1}}(SHA_{256}(M))$$

Solution: Confidentiality due to encryption Bob's public key. Integrity due to presence of the hash of M. Non-repudiation due to the digital signature.

3. Alice and Bob privately exchange a symmetric key s_k in advance. Alice later uses this key to send to Bob:

$$AES_{s_k}(M)||AES - EMAC_{s_k}(SHA_{256}(M))$$

Solution: Confidentiality due to AES encryption. Integrity due MAC.

4. Alice generates a new symmetric key s_K and sends to Bob:

$$E_{K_B}(s_k)||Sign_{k_A{}^{-1}}(SHA_{256}(s_k))||AES_{s_k}(M)$$

Solution: Confidentiality due to encryption with Bob's public key. Non-repudiation due to the presence of a digital signature. Note that integrity is not guaranteed because the hash is applied to the key, not the message.

4 Security Requirements Question 2

For this problem, assume that Alice wants to send a **single message M** to Bob. To do so, Alice and Bob can potentially use a number of different approaches and cryptographic technologies, which we will describe using the following terminology

M	Plaintext for a single message
s_k	Symmetric cryptography key
AES_{s_k}	Symmetric-key encryption using CBC mode, with the key s_k
SHA_{256}	SHA-256 hash function
$AES\text{-}EMAC_{s_k}$	Keyed MAC function with symmetric key s_k
K_A	Alice's public key
${K_A}^{-1}$	Alice's corresponding private key
K_B	Bob's public key
${K_B}^{-1}$	Bob's corresponding private key
E_K	Public-key encryption with the key K
$Sign_{k^{-1}}$	Digital signature using the private key

You can assume that the public keys have been securely distributed, so Alice and Bob know their correct values. Symmetric keys have not been exchanged.

Consider the following properties that Alice and Bob might desire their communication to have: **Confidentiality**, **Integrity**, and **Non-Repudiation**.

For each of the following possible communication approaches, Mention (and explain why) which of these properties will securely hold (or not hold) in the presence of Mallory, a Man In The Middle (MITM) attacker.

Mention **None** if none of the properties hold. If an approach fails entirely (will not result in Bob being able to read a given message M), mention **Broken**.

Note that || denotes concatenation.

1. Alice generates a new symmetric key s_k and sends to Bob:

$$E_{K_A}(s_k)||E_{K_B}(s_k)||AES_{s_k}(M)$$

Solution: This scheme only provides Confidentiality. While Bob cannot recover s_k from $E_{K_A}(s_K)$ (because Bob lacks Alice's private key), he can do so from $E_{K_B}(s_k)$. The use of AES provides confidentiality. However, without a separate MAC, the communication lacks integrity, and because Alice does not sign her message, it also lacks non-repudiation. Note that confidentiality for multiple messages is undermined by the lack of use of an Initialization Vector. However, the problem framing specifically discusses Alice sending a single message.

2. Alice sends to Bob:

$$E_{K_A}(M)||Sign_{K_A^{-1}}(SHA_{256}(M))$$

Solution: This scheme provides Confidentiality (via the use of AES) and Integrity (via use of the keyed MAC function). It does not provide non-repudiation because the integrity/authentication component does not demonstrate possession of Alice's private key. That said, a legitimate criticism of this approach is the reuse of the same key for encryption and the MAC computation, which may make it easier to break the secret key.

3. Alice generates a new symmetric key sK and sends to Bob:

$$E_{K_A}(s_k)||E_{K_B}(s_k)||Sign_{K_A^{-1}}(s_k)||AES_{s_k}(M)$$

Solution: The crucial insight for this problem is that Alice's signature over s_k allows Mallory to recover s_k simply by computing $E_{K_A}(Sign_{K_A^{-1}}(s_k))$, which Mallory can easily do since K_A is well-known. Given possession of s_k, all of the properties fail to hold: Mallory can read the message and can alter it, so there is no confidentiality and no integrity. There is no non-repudiation, either; all that the signature can demonstrate is that Alice signed s_k, but not that she signed M. However, Bob can still recover M. Therefore this scheme is marked None, rather than Broken.