Hardware-Oriented Security and Trust
ECE 4156 HST / ECE 6156 HST
Spring 2024

Assoc. Prof. Vincent John Mooney III

Georgia Institute of Technology

Homework 5, 85 pts. (ECE 4156) 100 pts. (ECE 6156)

Solution!

1) (30 pts.) Let F be a pseudorandom function. Show that each of the following MACs is insecure, even if used to authenticate fixed-length messages. (In each case Gen outputs a uniform $k \in \{0,1\}^n$. Let $\langle i \rangle$ denote an $\frac{n}{2}$-bit encoding of the integer i.)

a. (5 pts.) To authenticate a message $m = m_1, \dots, m_\ell$, where $m_i \in \{0,1\}^n$, compute $t := F_k(m_1) \oplus \dots \oplus F_k(m_\ell)$.

Solution

Let there be two messages $m1$ and $m2 \in \{0,1\}^n$ which are distinct. The tag computed for message *(m1, m2)* is identical to the tag generated for *(m2, m1)*. Hence, the adversary can obtain the tag for *(m1, m2)* and then output the message *(m2, m1)* together with the tag received.

b. (10 pts.) To authenticate a message $m = m_1, \dots, m_\ell$, where $m_i \in \{0,1\}^{\frac{n}{2}}$, compute $t := F_k(\langle 1 \rangle \| m_1) \oplus \dots \oplus F_k(\langle \ell \rangle \| m_\ell)$.

Solution

It is assumed that $(m_1, m_1', m_2, m_2') \in \{0,1\}^{n/2}$ where $m1$ and $m1'$ are distinct and $m2$ and $m2'$ are also distinct (i.e., not equal). The attacker obtains (i) tag $t1$ for the message $(m1, m2)$; (ii) tag $t2$ for $(m1', m2)$; and (iii) tag $t3$ for $(m1, m2')$. The attacker can verify $t1 \oplus t2 \oplus t3 = t4$. One can verify that $t4$ is a valid tag for $(m1', m2')$.

c. (15 pts.) To authenticate a message $m = m_1, \dots, m_\ell$, where $m_i \in \{0,1\}^{\frac{n}{2}}$, choose uniform $r \in \{0,1\}^n$ and compute

$$t := F_k(r) \oplus F_k(\langle 1 \rangle \| m_1) \oplus \dots \oplus F_k(\langle \ell \rangle \| m_\ell)$$

where the tag which is transmitted is $<r, t>$.

(NOTE: this is problem 4.7 on page 148 of Katz and Lindell.)

Solution

Let $m_1 \in \{0,1\}^{n/2}$

The attacker can set $r = <1 \| m1>$

The output would be

$<r,t> = <r,0^n> = <1 \| m1, 0^n>$ on message $m1$.

Hence, with $r = <1 \| m1>$ any message $m1$ would authenticate on tag 0^n.

2) (20 pts.) We explore what happens when the basic CBC-MAC construction is used with messages of different lengths.

a. (10 pts.) Say the sender and receiver do not agree on the message length in advance (and so $\mathsf{Vrfy}_k(m, t) = 1$ iff $t \overset{?}{=} \mathsf{Mac}_k(m)$, regardless of the length of m), but the sender is careful to only authenticate messages of length $2n$. Show that an adversary can forge a valid tag on a message of length $4n$.

Solution

First of all, Katz and Lindell note that there are many solutions to this problem! We give here one possible answer.

Let $m_1, m_2 \in \{0,1\}^n$ be arbitrary. The attacker requests a tag for message (m_1, m_2) and obtains tag t. The attacker can check that tag t is a valid tag for the message $(m_1, m_2, t \oplus m_1, m_2)$. Thus, the adversary can forge a valid tag for a message of length $\boldsymbol{4n}$.

b. (10 pts.) Say the receiver only accepts 3-block messages (so $\mathsf{Vrfy}_k(m, t) = 1$ only if m has length $3n$ and $t = \mathsf{Mac}_k(m)$), but the sender authenticates messages of any length a multiple of n. Show that an adversary can forge a valid tag on a new message.

Solution

First of all, Katz and Lindell note that there are many solutions to this problem! We give here one possible answer.

Let $m_1, m_2, m_3 \in \{0,1\}^n$ be arbitrary. A tag t_1 can be obtained for message m_1. Similarly, tag t_2 for $m_2 \oplus t_1$ and tag t_3 for $m_3 \oplus t_2$ can be obtained. Tag t_3 will be valid for the message (m_1, m_2, m_3).

(NOTE: this is problem 4.13 on page 149 of Katz and Lindell.)

The rest of this homework is based on the "Cryptography Part VII: Diffie Hellman" lecture the concept was introduced of key exchange where two people Alice and Bob choose secrets x and y such that when they exchange $\alpha^x \bmod p$ and $\alpha^y \bmod p$ they each can compute a shared secret key $K = (\alpha^x)^y \bmod p = (\alpha^y)^x \bmod p$. This shared secret key cannot be recovered by an adversary with any reasonable probability of success where the adversary records all channel exchanges and thus acquires both $\alpha^x \bmod p$ and $\alpha^y \bmod p$ (as well as α and p which are public). In the following questions you will both provide a discussion of possible scenarios and calculate actual values for Diffie-Hellman key exchange.

3) (20 pts.) The first step in Diffie-Hellman key exchange is the publication of an appropriate prime number p and generator α of Z_p^*. Is this first step a possible weakness in the protocol? In other words, are there scenarios where an adversary may or may not be able to manipulate this first step and gain a significant advantage?

a. (5 pts.) Describe your scenario. In other words, give a specific case, e.g., two embedded devices communicating over a radio link or two personal computers communicating over the internet, in sufficient detail to explain your answers to the rest of this question, question 3 on homework 5. Use Alice and Bob for the two legitimate communicating parties, and use Elektra to refer to the adversary. Please make sure that all relevant details are explained here, e.g., you may want to include (or not include) some form of a "Trusted Third Party" or TTP whose operation you should specify in sufficient detail. HINT: one suggestion is to first answer parts b, c and d below, then return to this part, part a, and write up the scenario.

Solution

Alice and Bob each have a personal computer (PC) and are communicating using Internet Protocol version 4 (IPv4). Alice and Bob both have access to sufficient compute resources to work with very large (e.g., 128-bit or 256-bit or 2048-bit) numbers and to generate large prime numbers as well as large truly random numbers. However, Alice and Bob do not know each other from any previous interactions and do not, for example, know what each other looks like.

b. (5 pts.) Describe the attack surface. For the scenario of part a, discuss how the adversary, Elektra, may choose to manipulate the setting of the first step of Diffie-Hellman Key Exchange. You should also discuss how Alice and Bob may choose to set up the scenario such that it is extremely difficult for an adversary to carry out any manipulations successfully. Your answer here should be comprehensive, i.e., the tradeoffs you will show in your answers to parts c and d below should be described here in terms of the strengths and weaknesses of various setups for your chosen scenario. HINT: one suggestion is to first answer parts c and d below, then return to this part, part b, and write up a description of the attack surface.

Solution

The attack surface includes internet packets with incorrect source information, i.e., a packet which is not sent by Alice or Bob may have the name of the sender changed to say "Alice" or "Bob" even though the packet was sent by someone else. Alice and Bob also may have malware on their computers. Alice and Bob may be subject to video surveillance which is able to capture keystrokes visually. Finally, it is possible at some point in the future that the PCs being used by Alice and Bob may be acquired by an adversary.

c. (5 pts.) For this part – part c of question 3 – your overall answer is supposed to be that yes, the adversary (Elektra) may be able to set up the scenario with an inherent weakness. Please precisely describe the scenario and attack surface in detail and describe how the attack could be carried out successfully. Please note that all steps in the attack should bereasonable for today's technology, e.g., you are not allowed to assume the availability ofa 10,000-bit quantum computer, but you may have some of the hardware (e.g., Alice's phone or Bob's phone) acquired by Elektra somehow.

Solution

The most obvious attack depends on lack of authentication, specifically, entity authentication. In other words, Elektra inserts herself in between Alice and Bob. Elektra pretends to be Bob when talking to Alice; and Elektra pretends to be Alice when talking to Bob. The steps are as follows:

i. Alice chooses a random secret x, $1 \leq x \leq p\text{-}2$, and sends **α^x mod p** to Bob' (i.e., Elektra).
ii. Elektra chooses a random secret x', $1 \leq x' \leq p\text{-}2$, and sends **$\alpha^{x'}$ mod p** to Bob claiming to be Alice (so Bob is communicating with Alice', i.e., Elektra, but Bob thinks he is communicating with Alice).
iii. Bob chooses a random secret y, $1 \leq y \leq p\text{-}2$, and sends **α^y mod p** to Alice' (i.e., Elektra)
iv. Elektra chooses a random secret y', $1 \leq y' \leq p\text{-}2$, and sends **$\alpha^{y'}$ mod p** to Alice claiming to be Bob (and Alice thinks she is communicating with Bob, but in fact she is communicating with Elektra).
v. Alice and Elektra compute the shared secret **K1 = $(\alpha^x)^{y'}$ mod p**
vi. Bob and Elektra compute the shared secret **K2 = $(\alpha^{x'})^y$ mod p**

The result of the above is that every message between Alice and Bob passes through Elektra who decrypts the message and then properly encrypts the message to pass it on to the recipient. Alice and Bob think that they are communicating with each other, which they are in fact doing; but they are communicating with each other only indirectly (i.e., through Elektra). Alice and Bob have used Diffie-Hellman Key Exchange. However, by not authenticating each other, they have become vulnerable to this attack which is known as the "Man-in-the-Middle" attack.

A lot of students said that another possible attack is to change p and α so that they are both small numbers. This idea does not work for the following reason: it is easy to check for this case! In other words, Bob or Alice can simply look at p and α to and carry out mathematical comparison operations to see if either of the values is too small; this check will take constant time, i.e., will be fast to carry out.

On the other hand, a legitimate attack is the following: the adversary could guess that x and y are small numbers. The reason that a device might choose nonrandomly and prefer small numbers is the complexity of exponentiation. A battery powered device can potentially save a lot of energy by choosing a small number for x or y. So, the adversary can guess that both x and y are small numbers – e.g., if the two devices are energy constrained – and construct the secret K from a brute force attempt over a set of small numbers and potentially find the actual shared secret K.

A final good answer that some students gave is that p and α are changed so that they are not a proper prime number-generator pair. In other words, they are both large numbers, and p does appear to be prime, but it is not in fact prime. In reality, p can be easily factored, and as a result the search space can be dramatically reduced. This was a very good answer.

d. (5 pts.) Now give a set of assumptions, e.g., involving a TTP, such that no attack you can think of could succeed. You should describe at least three attacks that fail including the attack in your answer to part c.

Solution

A TTP could provide entity authentication. In particular, the TTP can verify that a message claiming to be from Bob really is from Bob. Similarly, the TTP can verify that a message claiming to be from Alice really is from Alice. Entity authentication is important prior to establishing the shared key K. Once Bob and Alice have a shared key K, they are the only two entities who can read their messages, and so from then on the TTP is no longer needed. In summary, a TTP can provide entity authentication but then is no longer needed for the rest of the communication session.

i. The above will stop a "Man-in-the-Middle" attack.
ii. Assuming that the random secret (e.g., x) and shared key K are erased, physical possession of the personal computer will not allow an attacker to decrypt past messages which the attacker grabbed from the internet and stored. E.g., for a ciphertext c communicated earlier and recorded by the attacker, assuming the decrypted plaintext m, random secret x and shared key K are no longer on the device (the PC), physical possession of the device at a later date will not provide the attacker with a way to decrypt c.
iii. The attack of choosing a small p and a small α still fails. Alice and Bob both check for small numbers and complain to the TTP which then can investigate the source of the fake p and α.

In closing, one student mentioned that overloading the network could result in denial-of-service (DoS) which no scheme would prevent. It is true that if Alice and Bob cannot communicate due to the network not being available, then they are denied service and this DoS attack is not stopped. However, we meant here an attack that obtains the plaintext.

4) (15 pts.) Now we are going to carry out Diffie-Hellman Key Exchange. Consider the case of using $\alpha = 6$ as a generator for Z_1^*.

a. (1 pt.) Choose appropriate Alice and Bob secrets x and y, email the TA, Kevin Hutto, at khutto30@gatech.edu and then check the course webpage for homeworks http://mooney.gatech.edu/Courses/ECE4156/hwlabexam/index.html to see if your number has already been taken or not. You need to make a first attempt to choose a number prior to 2pm on Tuesday Feb. 20. If your first name, first letter of your last name, and selected pair of numbers appears under homework 5, you are done. Otherwise, prior to 2pm on Wednesday Feb. 21, you need to make a second attempt to choose your secrets x and y and send another email to another email to the TA. Again, if your first name and number appears under homework 5, you are done. Otherwise, prior to 2pm on Thursday Feb. 22, you need to try again.

   Choose private keys **before the deadline to not lose a point.**

b. (8 pts.) For your given secrets x and y, compute the following quantities:

i. (2 pts.) α^x

ii. (2 pts.) α^y

iii. (2 pts.) $(\alpha^x) \bmod p$

iv. (2 pts.) $(\alpha^y) \bmod p$

Solution

Here is a partial list of answers provided by the class (note that these answers have not been checked for accuracy):

x	*y*	a^x	a^y	$(a^x) \bmod p$	$(a^y) \bmod p$	*K*
2	3	36	216	10	8	12
2	4	36	1292	10	9	3
2	5	36	7776	10	2	4
2	7	36	279936	10	7	10
2	8	36	1679616	10	3	9
2	10	36	60466176	10	2	3
3	4	216	2592	8	9	1
3	5	216	7776	8	2	8
3	6	216	46656	8	12	12
3	7	216	279936	8	7	5
3	11	216	362797056	8	11	5
4	6	1296	46656	9	12	1
4	7	1296	279936	9	7	9
4	8	1296	1679616	9	3	3
4	9	1296	10077696	9	5	1
4	11	1296	362797056	9	11	3
5	6	7776	46656	2	12	12
5	7	7776	279936	2	7	11
5	8	7776	1679616	2	3	9
7	8	279936	1679616	7	3	3
7	9	279936	10077696	7	5	8
7	11	279936	362797056	7	11	2
8	10	1679616	60466176	3	4	3
9	11	10077696	362797056	5	11	8

c. (6 pts.) Finally, compute the shared secret key $K = (\alpha^x)^y \bmod p = (\alpha^y)^x \bmod p$ and show the intermediate step of $(\alpha^x)^y = (\alpha^y)^x$.

Check the answer above with the table.

5) [ECE 6156 only!] (15 pts.) Write pseudocode to carry out a brute-force attack against Diffie-Hellman key exchange. Assume that the adversary has acquired α, p, ((α^x) mod p) and ((α^y) mod p). Show the steps to carry out the brute-force attack to your answer to the previous question (question 4) on this homework.

Solution

It is assumed that the attacker has acquired α, p, ((α^x) mod p) and ((α^y) mod p)
Various solutions exist, one pseudocode solution is as follows:

```
int p, alpha, alicePub, bobPub; //Provided
int aliceGuess, xGuess = 0; //We only need x or y, don't need to search for both
aliceGuess = alpha ^ xGuess % p;
        while (aliceGuess != alicePub){
                xGuess++;
                aliceGuess = alpha ^ xGuess % p;
        }

key = bobPub ^ xGuess % p;
```

Note: Some students stated you can decrypt with random keys until you receive a message that "makes sense". This does not work as the question makes no mention of any retrieved messages outside of the key exchange, no encryption algorithm for the exchanged key is stated, and wrong keys can cause individual messages to appear to "make sense", even though the retrieved unencrypted message is the wrong message.