UNIVERSITY OF EDINBURGH

COLLEGE OF SCIENCE AND ENGINEERING

SCHOOL OF INFORMATICS

INFR10067 COMPUTER SECURITY

May 2020

13:00 to 15:00

INSTRUCTIONS TO CANDIDATES

Answer any TWO of the three questions. If more than two questions are answered, only QUESTION 1 and QUESTION 2 will be marked.

All questions carry equal weight.

This is an OPEN BOOK examination.

Year 3 Courses

Convener: S.Ramamoorthy
External Examiners: S.Rogers, S.Kalvala, H.Vandierendonck

THIS EXAMINATION WILL BE MARKED ANONYMOUSLY

1. **Networking**

(a) Link and network layer:

i. Pick one of the following 4 answers (A,B,C,D) and explain your reasoning: An IP packet has the characteristic that: (A) it is encrypted with the public-key of the receiver, (B) it is easy for an intermediate PC in the route between sender and receiver to modify its contents, (C) contains only the IP address of the receiver, (D) has a special field that contains a digital signature. [*1 mark*]

ii. Pick one of the following 4 answers (A,B,C,D) and explain your reasoning: The network layer of the Internet protocol stack has the following characteristic: (A) it contains information about how many other packets are related to the current packet, (B) contains information about the size of the file that is being transmitted, (C) contains information related to how many "hops" the packet has traversed in the network, (D) carries the hardware address of the receiver. [*2 marks*]

iii. Eve being evil wants to eavesdrop on Alice and Bob. Assume that Alice, Bob, and Eve are on the same local-area network and are configured with the following IP and MAC addresses. Note that Alice and Bob only know each others IP addresses.

Party name	IP address	MAC address
Alice	`192.168.0.1`	`00:11:22:33:44:11`
Bob	`192.168.0.2`	`00:11:22:33:44:22`
Eve	`192.168.0.3`	`00:11:22:33:44:55`

How can Eve stage a person-in-the-middle attack to eavesdrop on Alice and Bob using an ARP spoofing attack? [*3 marks*]

(b) TCP uses a three way handshake to initialise connections.

i. Describe the process of how sequence numbers are used in the TCP protocol. [*3 marks*]

ii. Explain how TCP messages can be used to perform a SYN flooding denial of service attack. Which of the six security properties is violated by this attack? [*2 marks*]

iii. Do randomly generated initial sequence numbers in the TCP handshake harden the protocol against SYN flooding?
Can you think of an attack different from SYN flooding that is made more difficult by randomly generated sequence numbers? [*3 marks*]

(c) TLS and Tunneling

i. Explain what forward security means. [*2 marks*]

QUESTION CONTINUES ON NEXT PAGE

ii. Over the past 25 years, various versions of the SSL/TLS protocol have been developed. The more recent the version, the more secure the protocol is. Why are older versions of the protocol still supported by many servers and browsers? Discuss security issues that can emerge from supporting old and new versions of the protocol simultaneously. [*4 marks*]

iii. TLS-RSA is not included in the newest version of TLS. Do you think it is a good idea to remove TLS-RSA? Justify your answer. [*3 marks*]

iv. How can Alice surf the web without her activity being logged by her ISP. [*2 marks*]

2. **Cryptography**

(a) Let $\Pi_1 = (\mathsf{Enc1}, \mathsf{Dec1})$ and $\Pi_2 = (\mathsf{Enc2}, \mathsf{Dec2})$ be two symmetric encryption schemes for which it is known that one of them is secure, but not the other. More precisely, Π_1 and Π_2 have the same key space $\mathcal{K}$, the same message space $\mathcal{M}$, and the same ciphertext space $\mathcal{C}$:

$$\begin{aligned} \mathsf{Enc1}, \mathsf{Enc2} &: \mathcal{K} \times \mathcal{M} \to \mathcal{C} \\ \mathsf{Dec1}, \mathsf{Dec2} &: \mathcal{K} \times \mathcal{C} \to \mathcal{M} \end{aligned}$$

The thing is that you don't know which one is secure and which one is not. Relying on this you want to build a secure scheme $\Pi = (\mathsf{Enc}, \mathsf{Dec})$. You suggest to double the key and ciphertext sizes and use the following encryption algorithm combining $\mathsf{Enc1}$ and $\mathsf{Enc2}$:

i. Compute the first half of the ciphertext c_1 as the encryption of the message m under the first half of the key k_1 using $\mathsf{Enc1}$;
ii. Compute the second half of the ciphertext c_2 as the encryption of the message m under the second half of the key k_2 using $\mathsf{Enc2}$;
iii. Return the concatenation of c_1 and c_2.

In other words, you define the encryption algorithm as follows:

$$\mathsf{Enc}(k_1||k_2, m) = \mathsf{Enc1}(k_1, m)||\mathsf{Enc2}(k_2, m)$$

Is this construction secure given that only one of the symmetric encryption schemes is secure? If your answer is yes, provide an explanation of why it can hide the message. If your answer is no, demonstrate an attack where the adversary recovers the plaintext underlying a ciphertext without knowing the encryption key. *[5 marks]*

(b) Let H_1 and H_2 be two hash functions for which it is known that one of them is one way, but not the other. More precisely, H_1 and H_2 have the same output size n:

$$H_1, H_2 \;:\; \{0,1\}^* \to \{0,1\}^n$$

The thing is that you don't know which one is one way and which one is not. Relying on this you want to build a hash function H which satisfies one wayness. You suggest to double the output size and use the following hashing algorithm combining H_1 and H_2:

i. Compute the first half of the hash h_1 as the hash of the message m under H_1;
ii. Compute the second half of the hash h_2 as the hash of the message m under H_2;
iii. Return the concatenation of h_1 and h_2.

QUESTION CONTINUES ON NEXT PAGE

In other words, you define the hashing algorithm as follows:

$$H(m) = H_1(m)||H_2(m)$$

Is this construction one way given that only one of the hash functions is one way? If your answer is yes, provide an explanation of why it prevents efficiently computing pre-images. If your answer is no, demonstrate a pre-image attack. *[5 marks]*

(c) Let H_1 and H_2 be two hash functions for which it is known that one of them is collision resistant, but not the other. More precisely, H_1 and H_2 have the same output size n:

$$H_1, H_2 \;:\; \{0,1\}^* \to \{0,1\}^n$$

The thing is that you don't know which one is collision resistant and which one is not. Relying on this you want to build a hash function H which satisfies collision resistance. You suggest to double the output size and use the following hashing algorithm combining H_1 and H_2:

i. Compute the first half of the hash h_1 as the hash of the message m under H_1;
ii. Compute the second half of the hash h_2 as the hash of the message m under H_2;
iii. Return the concatenation of h_1 and h_2.

In other words, you define the hashing algorithm as follows:

$$H(m) = H_1(m)||H_2(m)$$

Is this construction collision resistant given that only one of the two hash functions is collision resistant? If your answer is yes, provide an explanation of why it prevents efficiently finding collisions. If your answer is no, demonstrate a collision attack. *[5 marks]*

(d) Let $\Pi_1 = (\mathsf{Sign1}, \mathsf{Vrfy1})$ and $\Pi_2 = (\mathsf{Sign2}, \mathsf{Vrfy2})$ be two Message Authentication Codes (MAC) schemes for which it is known that one of them is secure, but not the other. More precisely, Π_1 and Π_2 have the same key space $\mathcal{K}$, the same message space $\mathcal{M}$, and the same tag space $\mathcal{T}$:

$$\begin{aligned} \mathsf{Sign1}, \mathsf{Sign2} &: \mathcal{K} \times \mathcal{M} \to \mathcal{T} \\ \mathsf{Vrfy1}, \mathsf{Vrfy2} &: \mathcal{K} \times \mathcal{M} \times \mathcal{T} \to \{\top, \bot\} \end{aligned}$$

The thing is that you don't know which one is secure and which one is not. Relying on this you want to build a secure scheme $\Pi = (\mathsf{Sign}, \mathsf{Vrfy})$.

QUESTION CONTINUES ON NEXT PAGE

You suggest to double the key and tag sizes and use the following signing algorithm combining $\mathsf{Sign1}$ and $\mathsf{Sign2}$:

i. Compute the first half of the tag t_1 as the signature of the message m under the first half of the key k_1 using $\mathsf{Sign1}$;
ii. Compute the second half of the tag t_2 as the signature of the message m under the second half of the key k_2 using $\mathsf{Sign2}$;
iii. Return the concatenation of t_1 and t_2.

In other words, you define the signing algorithm as follows:

$$\mathsf{Sign}(k_1||k_2, m) = \mathsf{Sign1}(k_1, m)||\mathsf{Sign2}(k_2, m)$$

Is this construction secure given that at only one of the two MAC schemes is secure? If your answer is yes, provide an explanation of why it prevents forgeries. If your answer is no, demonstrate a forgery attack. [*5 marks*]

(e) Let $\Pi_1 = (\mathsf{Enc1}, \mathsf{Dec1})$ and $\Pi_2 = (\mathsf{Enc2}, \mathsf{Dec2})$ be two symmetric encryption schemes for which it is known that one of them is secure, but not the other. More precisely, Π_1 and Π_2 have the same key space $\mathcal{K}$, the same message space $\mathcal{M}$, and the same ciphertext space $\mathcal{C}$:

$$\begin{array}{rcl} \mathsf{Enc1}, \mathsf{Enc2} & : & \mathcal{K} \times \mathcal{M} \to \mathcal{C} \\ \mathsf{Dec1}, \mathsf{Dec2} & : & \mathcal{K} \times \mathcal{C} \to \mathcal{M} \end{array}$$

The thing is that you don't know which one is secure and which one is not. Relying on this you want to build a secure scheme $\Pi = (\mathsf{Enc}, \mathsf{Dec})$. You suggest using the following encryption algorithm combining $\mathsf{Enc1}$ and $\mathsf{Enc2}$:

i. Compute the intermediary ciphertext c' as the encryption of the message m under the key k using $\mathsf{Enc1}$;
ii. Compute the actual ciphertext c as the encryption of the intermediary ciphertext c' under the key k using $\mathsf{Enc2}$;
iii. Return c.

In other words, you define the encryption algorithm as follows:

$$\mathsf{Enc}(k, m) = \mathsf{Enc1}(k, \mathsf{Enc2}(k, m))$$

Note that in this construction we use the same key for both encryptions, and the key or ciphertext sizes have not been doubled. Is this construction secure given that only one of the symmetric encryption schemes is secure? If your answer is yes, provide an explanation of why it can hide the message. If your answer is no, provide a decryption attack. [*5 marks*]

3. **TOR**

In class, we discussed the details of the Onion Routing Protocol, and the TOR network. In particular, we saw that when connecting to a website through the TOR network, the client computer first fetches a list of all the TOR nodes. It then selects the entry node, the intermediate nodes and an exit node to create a circuit through the TOR network.

(a) Assume you live in a censored country. Your country (the censor) does not however block TOR and does not control any TOR relays. Can TOR help you hide from the censor which HTTP websites you visit? If yes, what is the minimum number of TOR relays (including the exit node) needed for your TOR circuit, to prevent the censor from blocking your traffic? Explain your answer. [*3 marks*]

(b) Assume there is a single dishonest TOR relay in the TOR network, but that you do not know which one it is. The malicious relay can even be an exit node. Can you use TOR, but still prevent the malicious TOR relay from learning which HTTP websites you visit? If yes, what is the minimum number of TOR relays (including the exit node) needed for your TOR circuit, to guarantee the malicious relay will not discover where your traffic is being routed to? Explain your answer. [*3 marks*]

(c) Assume now that there are multiple independent dishonest TOR relays (*i.e.* these nodes do not collude) on the TOR network, but that you do not know which ones these are. The malicious relays can even be exit nodes. Can you use TOR, and still prevent the malicious TOR relays from learning which HTTP websites you visit? If yes, what is the minimum number of TOR relays (including the exit node) needed for your TOR circuit, to guarantee the malicious relays will not discover where your traffic is being routed to? Explain your answer. [*3 marks*]

(d) Assume now that there are multiple colluding dishonest TOR relays on the TOR network, but that you do not know which ones these are. The malicious relays can even be exit nodes. Can you use TOR, and still prevent the malicious TOR relays from learning which HTTP websites you visit? If yes, what is the minimum number of TOR relays (including the exit node) needed for your TOR circuit, to guarantee the malicious relays will not discover where your traffic is being routed to? Explain your answer. [*3 marks*]

(e) In the previous questions we considered privacy. We now turn to integrity. Assume there is a single dishonest TOR relay in the TOR network, but that you do not know which one it is. The malicious relay can even be an exit node. Can you use TOR, and still guarantee data integrity for the HTTP websites you visit? If yes, what is the minimum number of TOR relays (including the exit node) needed for your TOR circuit, to guarantee

QUESTION CONTINUES ON NEXT PAGE

the malicious relay will not discover where your traffic is being routed to? Explain your answer. [*3 marks*]

We are now going to attempt to improve the privacy guarantees of TOR, by slightly changing the design of the protocol. Instead of building one circuit to the exit node, our new protocol requires that the client builds two circuits to the same exit node. The client will then send the same randomly chosen cookie to the exit node through both circuits, to inform the exit node that these should be considered in pairs. Now, when the client wants to send a packet to a website (*i.e.* through the exit node) it randomly picks one of the two circuits to send his packet through. Similarly, when the exit nodes wants to relay a packet back to the client from the server, it randomly selects one of the two circuits to send his packet through.

(f) What type of privacy attacks does this new protocol render more difficult? [*5 marks*]

(g) What type of privacy attacks does this scheme render easier? [*5 marks*]