CS5285 PS2

Gerhard Hancke

Question 1

In a system, each user has an entry in the system's password file: (y; s) where y is computed as follows and s is a salt.

i. *y = H(s; password)* XOR *password*

ii. *y = H(s) XOR H(password)*

iii. *y = Es(H(password))* where E is AES algorithm

iv. *y = $MAC_{password}(s)$* where MAC is CBC-MAC constructed with AES

List which of the methods above for computing y are effectively secured by the salt against precomputed dictionary attack. Assume that s is adequately long and random.

Question 1 Cont.

Solution:

The question asks for which approach does the salt actually influence the size of the dictionary as intended. The first and the fourth are OK, two other two only need dictionary size equal to if there was no salt. (2 marks per answer)

Question 2

In TLS you can specify different ciphersuites for communication.

(a) How many ciphersuites are there in the latest specification TLS 1.3

Solution:

Five (2 marks)

TLS_AES_128_GCM_SHA256

TLS_AES_256_GCM_SHA384

TLS_CHACHA20_POLY1305_SHA255

TLS_AES_128_CCM_SHA256

TLS_AES_128_CCM_8_SHA256

Question 2 Cont.

(b) For the following mode how is a data message encrypted and session keys generated?

TLS_AES_128_CCM_SHA256

Solution:

Use AES128 (2 marks) to calculate CBC-MAC (2 marks) and encrypt message with Counter Mode (2 marks). SHA256 is used for HKDF (HMAC-based Key Derivation Function) (2 marks).

Question 3

a) Find out who issued the certificate for https://mail.google.com and how long the certificate will be valid.
b) Find out or estimate how many certificates (approximately, no need to count them explicitly) your browser contains.
c) What is the significance of a CA certificate being contained in the browser?

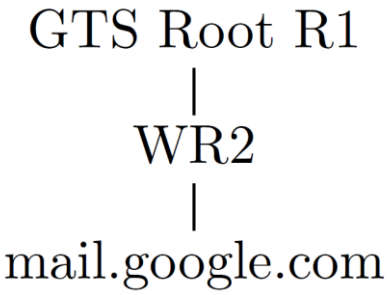
Question 3 Count.

d) The identity of the certificate for the question above is a DNS hostname. Certificates can also be used for signing and encrypting email. For a certificate used for email, what identifier would be used as the identity in the certificate?

Question 3 Solution

- Two marks for each question.

a) The certificate for https://mail.google.com was issued by *WR2*, which in turn was issued by *GTS Root R1 (in some cases GTS Root 1 self-signed/in others issued by GlobalSign Root CA-R1)*. The certificate chains could look like this:



The certificate is valid until 13.1.2025.

(Other answers with similar GTS chain also valid)

Question 3 Solution

b) This varies by browser - however there should be quite a few. About 100 Trusted Root CA certificates in the browser. Intermediate+Trust Root CAs.

[Exact number not important - there are 10s of certs, up to 100+, depending om browser]

Question 3 Solution

c) The special property of CA certificates included in a web browser is that the web browser implicitly trusts any certificate signed by one of these CAs.

d) Certificates used for encrypting or signing emails bind a public key to an ***email address***.

Question 4

- We are currently using Zoom (so are a lot of other people). Please provide a short explanation of what ``zoombombing'' is. Your answer should mention at least two main security services that are failing, as well as the main technical vulnerability that is allowing this to happen. Conclude by providing some mechanisms for mitigating issues for each of the security services you mentioned.

Question 4 Solution

- Failure of authentication and access control. Also possibly confidentiality.
- Main issue: In past meeting ID is short, brute force search for active links, meeting links shared in public.
- You need password, you need waiting room, disable annotation/screen sharing/chat.

Question 5(a)

a) Consider the following key exchange protocol which is similar to IKE Phase 1 Aggressive Mode. *p* is a large prime number and *g* is a generator of Z_p^*

1. $A \rightarrow B : g^a \bmod p,\ \{\text{“}Alice\text{”}\}_{Bob},\ \{R_A\}_{Bob}$
2. $A \leftarrow B : g^b \bmod p,\ \{\text{“}Bob\text{”}\}_{Alice},\ \{R_B\}_{Alice},\ \text{proof}_B$
3. $A \rightarrow B : \text{proof}_A$

Where

$$\text{proof}_A = h(g^{ab} \bmod p,\ g^a \bmod p,\ g^b \bmod p, \text{“}Alice\text{”})$$
$$\text{proof}_B = h(g^{ab} \bmod p,\ g^b \bmod p,\ g^a \bmod p, \text{“}Bob\text{”})$$
$$K = h(g^{ab} \bmod p)$$

Question 5(a) Cont.

i) First explain if the protocol authenticates A and B, and achieves secure key agreement (discuss key control and key authentication). {m}X denotes a message m encrypted with public key of x.

ii) Modify the protocol so that R_A and R_B can be eliminated but the protocol can mutually authenticate *A* and *B*. In your modification, no additional message protocol, secret keys or signature can be used.

Question 5 (a) i) Solution

A authenticates B: (2 points)
Challenge: $\{R_A\}_{Bob}$
Response: nil as proof_B has nothing to do with the challenge
B authenticates A: (2 points)
Challenge: $\{R_B\}_{Alice}$
Response: nil as proof_A has nothing to do with the challenge

The intention is that this is key agreement, with no key authentication. However, A and B are not authenticated to each other so the protocol fails overall.

Only Bob could know RA, or Alice know RB but this is never used in the proof.

Trudy sends $g^t \bmod p$ to A and B then calculate keys $g^{at} \bmod p$ and $g^{bt} \bmod p$, can also generate valid proofs to both.

Question 5 (a) ii) Solution

- How to fix this problem?
- A modified protocol without R_A and R_B: (6 points)

1. $A \rightarrow B : \{g^a \bmod p\}_{Bob},\ \{\text{“}Alice\text{”}\}_{Bob}$
2. $A \leftarrow B : \{g^b \bmod p\}_{Alice},\ \{\text{“}Bob\text{”}\}_{Alice},\ \text{proof}_B$
3. $A \rightarrow B : \text{proof}_A$

- *$\{g^a$ mod p, “Alice”$\}_{Bob}$* making it a single message is also OK.

Question 5(b)

(b) Consider the following simplified IKE Phase 1 in Aggressive Mode.

$$A \rightarrow B: \quad \text{“}Alice\text{”}, \text{“}Bob\text{”}, g^a \bmod p$$

$$A \leftarrow B: \quad \text{“}Bob\text{”}, \text{“}Alice\text{”}, g^b \bmod p, [g^a \bmod p]_B$$

$$A \rightarrow B: \quad \text{“}Alice\text{”}, \text{“}Bob\text{”}, [g^b \bmod p, g^a \bmod p]_A$$

$[X]_A$ denotes a signature on message X generated by A. The session key established between A and B is $g^{ab} \bmod p$. Show that this simplified version is insecure (allows attacker to establish a key with one of the participants while pretending to be the other participant) Hint: consider that this IPSec system has multiple users.

Question 5(b) Cont.

Solution (10 marks):

The protocol is vulnerable to man-in-the-middle attack where an attacker *E* can impersonate *B* while getting the session key of *A*. Suppose *E* is an attacker.

1. E intercepts the first message from A to B:

$$A \rightarrow B : \text{“}Alice\text{”}, \text{“}Bob\text{”}, g^a \bmod p$$

2. E sends the following message to B using her real identity

$$E \rightarrow B : \text{“}Eve\text{”}, \text{“}Bob\text{”}, g^a \bmod p$$

3. E receives the following message from B:

$$E \leftarrow B : \text{“}Bob\text{”}, \text{“}Eve\text{”}, g^b \bmod p, [g^a \bmod p]_B$$

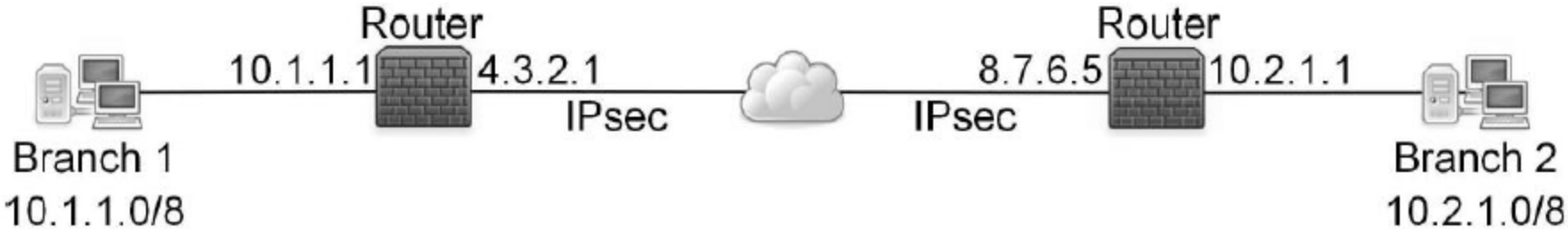
4. E sends the following to A while impersonating B:

$$A \leftarrow E : \text{“}Bob\text{”}, \text{“}Alice\text{”}, g^r \bmod p, [g^a \bmod p]_B$$

where *r* is randomly chosen by *E*. *A* computes the session key as g^{ar} *mod p* which can also be computed by *E*. Point deduction strategy.

Question 6

Imagine two branches of a corporate network are connected through the Internet. Specifically, each of the two branches has a router facing the Internet, communicating with the router of the other branch over IPSec (using ESP in tunnel mode).



Question 6 Count.

The network is setup such that two nodes from the two branches can communicate transparently. Imagine node 10.1.1.5 from branch 1 is sending a packet to node 10.2.1.6 from branch 2. Describe in detail the steps of how the packet travels between the two nodes and how it is encapsulated and decapsulated on the way.

Question 6 Solution

The following steps will take place as a packet travels between the two nodes:

1. Node 10.1.1.5 prepares a packet with the address 10.2.1.6 as the destination address and forwards it to the router of its network (the default gateway for the node).

Question 6 Solution Cont.

2. The router receives the packet on its internal interface 10.1.1.1 and realises that it is destined for a node in the network of branch 2. It therefore adds the ESP header, trailer and auth and then encapsulates the packet into an outer packet, with the address 4.3.2.1 as the source address (the external address of the router of branch 1) and the address 8.7.6.5 as the destination address (the external address of the router of branch 2). The packet is then sent over the Internet to the router of branch 2.

Question 6 Solution Count.

3. The router of branch 2 receives the packet on its external interface. It decapsulates the packet, checks the authentication trailer and decrypts the inner packet (which still has the destination address 10.2.1.6). It then forwards the inner packet onto the internal network of branch 2.
4. Node 10.2.1.6 receives the packet.

Question 7

IMPORTANT: Must have screenshot as requested in question

a&b) Recover the passwords in file1, file 2, file 3

Solution:

Search for salt/pass with MD5 five characters (lower, upper, numeric)

hashcat -m 20 -a 3 file1.txt -D 1 -1 ?l?u?d ?1?1?1?1?1

- File 1: 5-digits (A-Z,a-z,0-9) (2 marks)
  - Abc12, pwdss, 5285A, MyCaT, CityU, A2023
  - File 2: 5-digits with 8-bit SALT (2 marks)
  - XyZ89, 13579, 5285X, w00fY, myPwD, A2020

hashcat -m 20 -a 3 file3.txt -1 ?l?u?d ?1?1?1?1?1?1

- File 3: 6-digits (A-Z,a-z,0-9) (4 marks)
  - PWoRds, 192837, CS5285, mYfIsH, CSEEDS, SemA23

Question 7 cont.

c) How long did it take you to recover file 1 and file 2. Can you explain the time difference if any between the two recoveries?

Solution:

Not that noticeably different- because we are not building an entire dictionary. We know the salt so brute force searching for *hash(s,pwd)* not much longer than *hash(pwd)* (6 times the effort) as we are only looking for all password combinations with the 6 salts given. If we had more entries with 255 unique salt then we would be looking at getting the full extra 8-bit dictionary space. (4 marks)

d) How much longer did it take to find the 6-character passwords?

Solution: It take noticeably longer to find 6-character passwords, compared to 5 character. (4 marks)

Question 8

(a) Generate your own keypair using PGP
(b) Download the tsgexercise_cert.asc and import it into PGP

(c) Send us some text encrypted with the downloaded public key first and then signed with your own private key, rather than only send me your public key.

Different approaches, there is different email/PGP plugins you can use. We will provide individual feedback on this question if we receive your email.

- If we do not have email for you we will contact you
- If you hear nothing your submission was correct.