UNIVERSITY OF EDINBURGH

COLLEGE OF SCIENCE AND ENGINEERING

SCHOOL OF INFORMATICS

INFR10067 COMPUTER SECURITY

Monday 15th May 2017

14:30 to 16:30

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.

CALCULATORS MAY NOT BE USED IN THIS EXAMINATION

Year 3 Courses

Convener: C. Stirling
External Examiners: A. Cohn, A. Donaldson, S. Kalvala

THIS EXAMINATION WILL BE MARKED ANONYMOUSLY

1. **Passwords**

The social networking website LinkedIn was hacked on 5 June 2012, and passwords for nearly 6.5 million user accounts were stolen by cybercriminals. The stolen passwords, which were only hashed, were cracked and posted on a forum later on that day. Internet security experts said that the passwords were easy to unscramble because of LinkedIn's failure to use a salt when hashing them.

Assume that valid LinkedIn passwords were exactly 8 characters long (8 bytes), and may only contain characters a-z, A-Z, and 0-9. Also assume that LinkedIn users choose truly random passwords amongst the valid ones.

(a) Assume an attacker can perform k hashes per second. Given the hash of a single arbitrary password, how many hours would it take for the attacker to crack a single password by brute force, in the worst case? [*3 marks*]

(b) Assume an attacker can perform k hashes per second. Given the hash of a single arbitrary password, how many hours would it take for the attacker to crack a single password by brute force, on average? [*5 marks*]

(c) The attacker has a better idea. Instead of trying to break one password at a time, he considers computing the hash of every valid password and store them in a table of (hash, password) pairs sorted by hash. With this table, he could take any hash from the leaked LinkedIn password database and find the corresponding password very quickly.

Assuming that the hash function used for storing password is the sha-256 function which has a 256 bits output size, how many bytes would the table occupy? [*2 marks*]

(d) Realistically, the attacker will not be able to store this table. So he now considers using a *rainbow table*. A rainbow table is constructed by computing m chains, each of fixed length ℓ representing ℓ passwords and their hashes. Table 1 shows a simple example of a rainbow table with $m = 3$ chains (c_1, c_2, c_3). Each chain stores $\ell = 4$ passwords (p_0, p_1, p_2, p_3).

	c_1	c_2	c_3
p_0	wikipedia	abcdefgh	password
$h_1 = H(p_0)$	ao4kd	1vn6s	dicm4
$p_1 = R_1(h_1)$	secret	bernie	culture
$h_2 = H(p_1)$	9kpmw	kolscx	re3xes
$p_2 = R_2(h_2)$	jimbo	zurich	crypto
$h_3 = H(p_2)$	v0d$x	8ntpy	1tik0
$p_3 = R_3(p_3)$	rootroot	myname	linux23

Table 1: Example of a rainbow table

Chains are constructed using a family of reduction functions $R_1, R_2, \ldots, R_{\ell-1}$. These are deterministic functions that map each hash value back into a valid password. Each chain begins with a different password p_0. By alternating the hash function

QUESTION CONTINUES ON NEXT PAGE

with the reduction functions, chains of alternating passwords and hash values are formed.

Rainbow tables are constructed in such a way that only the first and last passwords in each chain need to be stored:

- the last password (*endpoint*) is sufficient to identify if a hash value is (likely) part of the chain,
- and the first password (*starting point*) is sufficient to reconstruct the whole chain.

This is precisely why rainbow tables save an enormous amount of space. In the example depicted in Table 1, only passwords p_0 and p_3 need to be stored in in each chain.

With such a table at hand, we can quickly find a password p^* that hashes to a particular hash value h^* as follows:

- We compute $R_{\ell-1}(h^*)$. If it is one of the endpoints present in the table, we can reconstruct the chain from the corresponding starting point and obtain the original value.
- If this value is not present as an endpoint, we move backward one step in the chain and compute $R_{\ell-1}(h(R_{\ell-2}(h^*)))$. If it is one of the endpoints present in the table, we can reconstruct the chain from the corresponding starting point and obtain the original value.
- If this value is not present as an endpoint, we move backward one step in the chain and compute $R_{\ell-1}(h(R_{\ell-2}(h(R_{\ell-3}(h^*)))))$. If it is one of the endpoints present in the table, we can reconstruct the chain from the corresponding starting point and obtain the original value.
- And so on...

If we do find a matching endpoint, reconstructing the chain based on the initial value of the chain will very likely find a password that hashes to h^* (note that collisions in the reduction functions cause occasional false positives).

You will assume that the attacker's rainbow table contains no collisions and that all valid passwords are represented exactly once. Give an equation for the number of bytes in the table in terms of the chain length ℓ and the size of the password set $\mathcal{P}$. [*5 marks*]

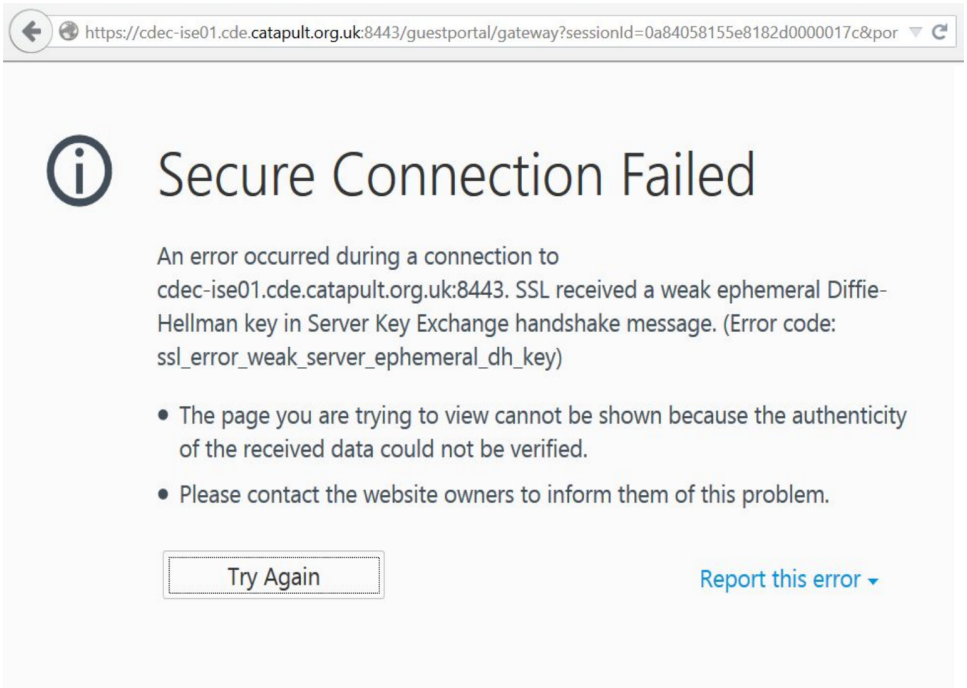
(e) Imagine LinkedIn had done the following thing: instead of storing h(password) it was storing h(`linkedin_secret` || password), where `linkedin_secret` would be a randomly generated 32-bits secret stored on the server (the same secret for all the passwords).

How would this design have (partially) defended against rainbow table attacks? Note that if the hashes are compromised, it is likely that the secret is too. [*5 marks*]

(f) Briefly explain how you would adjust the design in (e) to provide stronger protection. [*5 marks*]

2. **Analyze a warning**

When trying to open a website the instructor received the following warning in her web browser:



SSL supports several cryptographic algorithms and during the initial handshake the browser and the server negotiate which one they will use. Some of these algorithms are older and no longer considered to be secure by the cryptographic community. The warning above is telling the user that the remote server is only willing to use an older insecure protocol, so the browser has blocked the connection for the user's safety.

(a) To be secure a system must have five core properties. Name one property that would be violated if the user were to ignore the warning and view the page. Explain how the property would be violated. [*3 marks*]

(b) What Cyber Essentials property has the website owner failed to follow? [*2 marks*]

(c) What action(s) does the warning author want the user to take in response to the warning? [*2 marks*]

(d) Use the human-in-the-loop framework discussed in class to analyze the warning. Name three components (small boxes) of the framework. For each named component, apply it to the above warning. [*12 marks*]

QUESTION CONTINUES ON NEXT PAGE

(e) Suppose that Alice got the warning above when connecting to a National Health Service (NHS) website. She is concerned about other people learning about her private health information. For each of the following, would engaging in the described activity protect Alice's information from interception by a third party? Explain why or why not.

i. Use a Virtual Private Network (VPN) [*2 marks*]

ii. Turn on her browser's Incognito or Private Browsing Mode [*2 marks*]

iii. Log in from a different computer [*2 marks*]

3. **Web security**

AcmeMail is a webmail service provider. Its homepage can be accessed visiting the URL `https://www.coldmail.com`. To protect its millions of users, all pages on the AcmeMail website use HTTPS. Its users can delete their account by clicking on the "delete account" link on the AcmeMail homepage. This directs them to the following page

```
<!DOCTYPE html>
<html>
   <body>

   <h1>Click delete to confirm the deletion of your
                                                AcmeMail account</h1>

   <form action="deleteAccount.php" method="POST">
      <input type="hidden" name="user" value="[username]">
      <input type="submit" name="delete">
   </form>

   </body>
</html>
```

If the user is logged-in, the server will replace `[username]` with the user's username. The following pseudo-code describes the implementation of `deleteAccount.php`.

```
if valid_account($_POST["user"])
then delete_account($_POST["user"]);
     echo("<p>Your AcmeMail account has been deleted. Bye!</p><br>")
else echo("<p>$_POST["user"] is not a valid account!</p><br>")
```

(a) Assume that Eve knows the AcmeMail username of Alice. How could Eve delete Alice's account without interacting at all with Alice, or Alice's browser? [*5 marks*]

To thwart this attack, the security officer of AcmeMail has changed the implementation of `deleteAccount.php` to further check the authenticity of the cookie sent by the browser, and that it was issued for the specific username. `deleteAccount.php` now implements the following pseudo-code

```
if valid_cookie($_POST["user"], $_COOKIE["login_cookie"])
then delete_account($_POST["user"]);
     echo("<p>Your AcmeMail account has been deleted. Bye!</p><br>")
else echo("<p>Invalid account!</p><br>")
```

You will assume that `login_cookie` is hard to guess and unique to the user's account, and that the function `valid_cookie` checks that `login_cookie` is a valide cookie for user `$_POST["user"]`.

(b) Briefly explain cross-site request forgery (CSRF) attacks. Explain what are CSRF tokens and how they help defend against these attacks. [*5 marks*]

QUESTION CONTINUES ON NEXT PAGE

(c) Explain why this site is vulnerable to a CSRF attack that would allow Eve to delete Alice's AcmeMail account. [*5 marks*]

(d) Briefly explain cross-site scripting (XSS) attacks. Explain both types of XSS attacks. [*5 marks*]

(e) The site is also vulnerable to a XSS attack that would allow Eve to delete Alice's AcmeMail account. Explain why and how you would prevent this vulnerability. [*5 marks*]