Edward Fay: 460302131, Rowan Burns 480462657

Assignment 1

INFO3616 Semester 2, University of Sydney

1 Reliability and security (5 marks)

Reliability Engineering and Security Engineering are related. Explain: what aspect is absolutely specific to security engineering (and the reason it is its own discipline)?

Reliability engineering is the engineering of a system so that it can fulfil its intended function without failure (to be reliable). Security engineering is similar in that it seeks to engineer a system to fulfil its intended function without failure even when a malicious opponent is attempting to make the system fail.

This second half is the aspect specific to security engineering and not reliability engineering. In security engineering you are seeking to design a system that can tolerate the activities of a malicious opponent. It is about taking preventative actions as well as actively defending against an active attacker. This is in contrast to reliability engineering, which only deals with implicit threats of failure (mechanical wear, hardware malfunction).

It is for this reason that security engineering is its own discipline. It is a constant process, and the work is never finished, as opponents continually find new ways to try to induce failure in your system.

2 Security goals in hospitals (8 marks)

Referring to our reference framework, what kind of statement did your predecessor intend to write up? (2 mark)

The predecessor was intending to write a policy statement as part of the reference framework. The policy statement defines what we mean to achieve, which in other words is what it means to keep the system secure. In this case the core priority of the policy statement would be to keep patient data safe.

It is fairly obvious that 'keeping patient data safe' requires meeting some security goals. Name 3 security goals relating to 'keeping patient data safe'. For each, explain why it is a relevant security goal. Brief but intelligible answer is OK. (3 marks)

Confidentiality: to protect the secret content of a message or data. This is relevant as patient data is not safe if it can be read by anyone in the network.
Privacy: the right to determine what information relating to themselves they want to release or hide. A patient may have non-sensitive information they are happy to share such as that they received a Coronavirus test and are negative, whilst also having information like an existing medical condition they would like to be kept private.
Authorisation: that a verifier can determine whether an entity is allowed to execute some action or access some data. In a hospital setting, not all staff should be able to access the patient data. Accessing patient data may be completely beyond the scope of the role of a cleaner or supplies administrator, whilst doctors will need access to relevant patient data. Authorisation is necessary to control who has access to the data (assuming they are already authenticated).

For each security goal, state incentives against which you intend to defend! Brief but intelligible answer is OK. (3 marks)

Confidentiality: Monetary. Being able to collect huge datasets on patient health is valuable research material and hackers may attempt to intercept patient data to sell.
Privacy: Monetary gain by blackmailing patients. If a patient intends to keep some medical information private, they have a motivation for not letting the information become public knowledge. Hackers may ask for compensation in return for not publishing the information.
Authorisation: Curiosity. Hospital staff members may simply be curious about the conditions of patients that they are around on a daily basis. This information should only be accessed on a need to know basis however, so this needs to be controlled.

3 A common problem (12 marks)

a) The attack (6 marks)

- **Give two security-relevant pieces of information that the attacker extracted. (2 marks)**

  Email address (Rosemary@ttrzine.net) which may be used to send a spoof email as if it was from someone else in the company. **Username (R_Morgan)** which may be used in conjunction with a brute-forced password using the daughters name to gain access to her emails.

- **Which human weaknesses did the attacker exploit? Say why. (3 marks)**

  Confirmation Bias. The attacker builds on Rosemary's pre-existing knowledge of security attackers "Yes, I've heard about that", and so it is easier for her to accept what the attacker is saying.
  Respect for Authority. Rosemary is likely to comply with the attacker because they represent the security department and Rosemary expects them to be experts on the subject.
  Liking. Rosemary likes her work colleagues, the attacker is friendly to her and she wants them to like her, the attacker is able to take advantage of Rosemary's desire to be seen as helpful and comply with the attacker's request.

- **The attacker did not manage to get all information they wanted—namely the password. How can they possibly still get that piece of information without resorting to trying out all possibilities? (1 marks)**

  Given that the attacker knows that the password is based on the daughter's name, they could narrow down the possible passwords using a dictionary attack of common female names. This would drastically reduce the number of possibilities they would have to try. The attacker may go even further, and track Rosemary down on social media to try and find out explicitly what the daughter's name is from a happy birthday post etc.

b) Policy, our old friend (3 marks)

- **The password policy the attacker states is actually still very common (especially in Australia). Argue: why is it not a good policy? Give two reasons.**

  Recommending that a password is replaced every 90 days does not improve the security of a password.

  1. If a password is strong to begin with (cannot be brute forced), then it will remain strong. Asking someone to constantly change their passwords just entices them into choosing simple easy to remember passwords each time.
  2. A person can only remember a finite amount of information. If they are constantly changing their password, they will be motivated to use this password for multiple logins in an attempt to simplify the amount of information they must remember. If one set of credentials are breached then an attacker could use this password on every other login they access. Eg. crack a university login, and then use that same password to access bank accounts.

c) The aftermath (3 marks)
Let us assume that this story continues. The attacker was ultimately successful in breaking into the system. Rosemary's boss berates her, stating that she was told in her induction that she must read the security policy (she didn't).

- **The boss's argument is unrealistic. What does evidence tell us? (1 mark)**

  Evidence tells us that attempting to educate users provides mixed success depending on the person in question. Just telling them to look at a company policy usually does not have a lasting effect.

- **The attack could have been thwarted by Rosemary in simple ways, had she been trained properly by the company. Give one non-technical defence. (1 mark)**

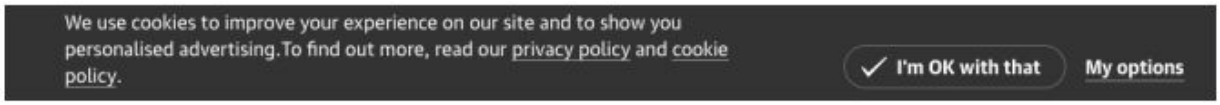
  If the caller was in reality a member of the company Information Security group, they would know of her email in the company email address book. There would be no need to ask for her email address as they would already know it. The simple way for Rosemary to thwart the hacker would be to ask the attacker what is her company email using the company address book.

- **Give Rosemary advice how to choose a better password. (1 mark)**

  A great method would be to install a password manager and have one strong master password protect a database of passwords. If the company does not allow the installation of other software however, a good human friendly method would be to 'letters-from-a-sentence'. You do this by choosing a memorable sentence such as *My daughter is fantastic at soccer! School captain in 2018* which can be shortened to Mdifas!Sci2018. This is ideal because it contains numbers, upper and lower case letters, and a special character.

4 Privacy warnings (3 marks)

In 2018, the EU's General Data Protection Regulation came into force. Around the same time, a large number of websites took to displaying 'cookie warnings'—pop-ups that inform the user that the site will set a cookie. Figure 1 shows the pop-up displayed by theguardian.com.



• Considering what you know about human decision-making, argue whether this is an effective way to get user approval. Be brief. (1 marks)

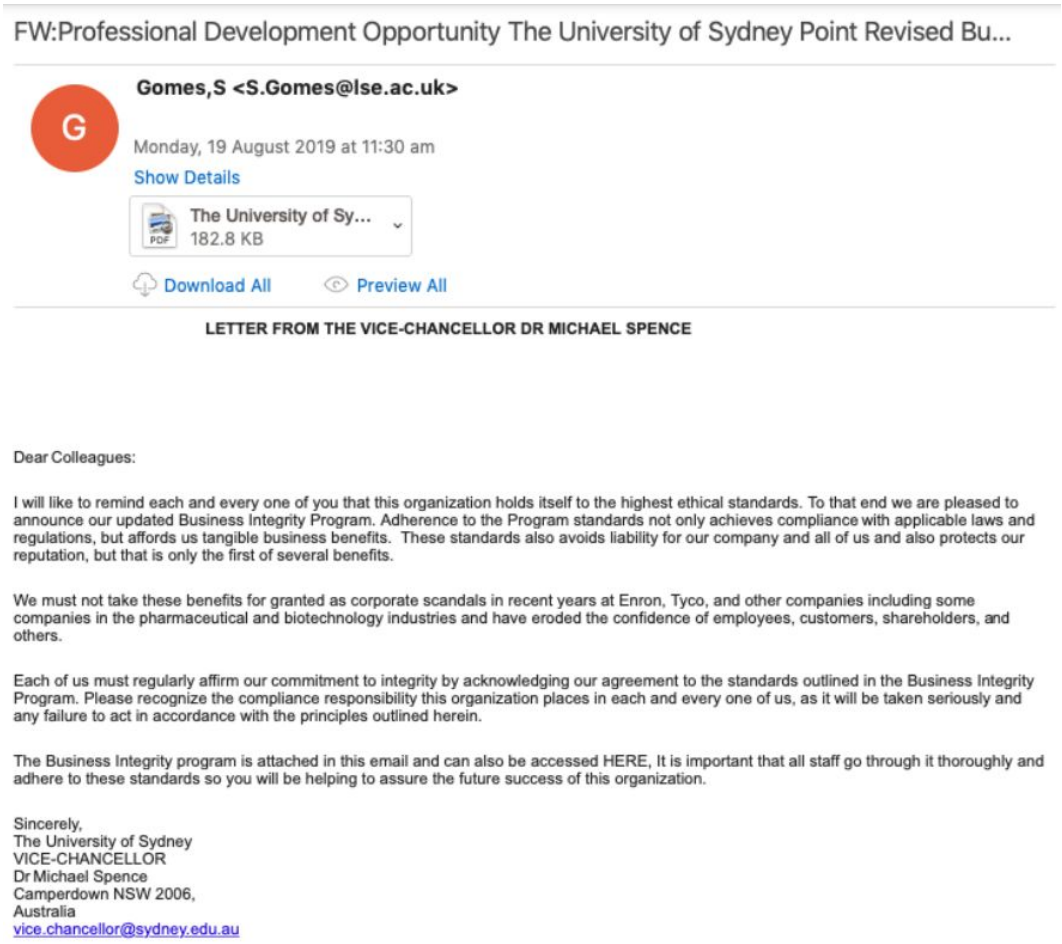
No. Users under pressure generally do not weigh alternative options and instead pick the first working solution to get rid of the warning.

• Discuss: can this be exploited by attackers? 1-2 sentences are enough. (2 marks)

Given that people will click anything to get rid of the warning, an attacker could spoof a cookie warning on a website where by clicking "agree to cookies" it actually executes some malicious code. The risk is that users receive no feedback on whether clicking the box was good or bad, and implicitly they assume that it was good because there is no negative reinforcement. However, the malicious code may have shared browser saved credit card credentials, and now attackers can then take money from the user months after the bad decision without the user associating this with the cookie box.

5 A suspected phish (6 marks)

Figure 2 shows a suspected phishing email that landed in the inboxes of USYD staff on 2019-08-20. It's also linked from the assignment on Canvas, for better readability. It is a screenshot from an Outlook email client. There was considerable confusion on staff side whether it is real or not.



FW:Professional Development Opportunity The University of Sydney Point Revised Bu...

Gomes,S <S.Gomes@lse.ac.uk>

Monday, 19 August 2019 at 11:30 am

Show Details

The University of Sy... 182.8 KB

Download All Preview All

LETTER FROM THE VICE-CHANCELLOR DR MICHAEL SPENCE

Dear Colleagues:

I will like to remind each and every one of you that this organization holds itself to the highest ethical standards. To that end we are pleased to announce our updated Business Integrity Program. Adherence to the Program standards not only achieves compliance with applicable laws and regulations, but affords us tangible business benefits. These standards also avoids liability for our company and all of us and also protects our reputation, but that is only the first of several benefits.

We must not take these benefits for granted as corporate scandals in recent years at Enron, Tyco, and other companies including some companies in the pharmaceutical and biotechnology industries and have eroded the confidence of employees, customers, shareholders, and others.

Each of us must regularly affirm our commitment to integrity by acknowledging our agreement to the standards outlined in the Business Integrity Program. Please recognize the compliance responsibility this organization places in each and every one of us, as it will be taken seriously and any failure to act in accordance with the principles outlined herein.

The Business Integrity program is attached in this email and can also be accessed HERE, It is important that all staff go through it thoroughly and adhere to these standards so you will be helping to assure the future success of this organization.

Sincerely,
The University of Sydney
VICE-CHANCELLOR
Dr Michael Spence
Camperdown NSW 2006,
Australia
vice.chancellor@sydney.edu.au

• State two weaknesses of the human mind that the design of this email targets, and give an example from the email for each. (4 marks)

Respect for authority. The attacker references the staff's responsibility to the university 'Please recognize the compliance responsibility to integrity…'. It is also signed off under the name of the Vice-Chancellor Dr Michael Spence. Belief in authorisation leads to compliance.
Consistency. The tendency in humans to comply with requests after making a previous public commitment. The attacker is playing on people's previous commitment to the Business Integrity Program. By playing this document off as an update to previously agreed upon standards, the targets are more likely to comply.

• The email client's rendering of the email actually tries to counteract some weaknesses of the human mind. State one way the email client tries to achieve this. Also state the weakness. (2 marks)

The email client displays the full email address of the attacker. This shows us that the email is not recognisable as Vice Chancellor Dr Michael Spence's or at all affiliated with the University of Sydney.

The weakness here is **Human need for order**, the email not matching the university signals to the user that something is off. It also undermines the **Respect for Authority** as it is clear that the email was not written by a genuine USYD member.

6 Evaluating a Bell-LaPadula-based security policy (10 marks)

a) Access Control (2 marks)
• What form of Access Control is implemented by Table 1? Say why. (1 mark)

Mandatory Access Control (MAC). An external entity defines the access of subjects to objects in a matrix.

• What form of Access Control is implemented by Tables 2 and 3? Say why. (1 mark)

Tables 2 and 3 also define Mandatory Access Control (MAC). This is for the same reason, the rules on access are clearly defined and cannot be revised by a subject.

b) Rule consistency (6 marks)
Are the rules in the tables consistent with each other? If not, which definitions in the ACL collide with the definitions via the clearance levels? Say why for credit.

- Arthur should not be able to read up for beetle.exe as he has protected access but it requires secret access. Arthur should also not be able to execute mspaint.exe as he only has protected access but it requires top secret.
- Bertie should not be able to write down for /tmp and transfers.csv as he has secret access but it only requires protected access. Bertie should also not be able to execute mspaint.exe as he only has secret access but it requires top secret.
- Kelly should not be able to read up for missile_codes.rar as she has secret access but it requires top secret access. Kelly should also not be able to write down for transfers.csv as she has secret access but it only requires protected access
- Dylan should not be able to write down for /tmp as he has top secret access but it only requires protected access
- Time.exe should also not be able to execute mspaint.exe as it only has secret access but it requires top secret.

c) Malicious operatives (2 marks)
• Kelly is the accountant. Is there anything stopping her from 'cooking the books', i.e., forging transfers? Why?

Yes, even though she has write access to the transfers.csv file in the ACL table, according to tables 2 and 3 she should not be able to write down (Bell-LaPadula model) as she has 'secret access' but it only requires 'protected'. In order to gain access to an object, all rules defined in the three tables must be correct. Given that there is a collision, she is not able to write down and forge transfers.

• Dylan is a beetle.exe addict and has had his permissions revoked. Can you find a way for him to get his fix?

Dylan can execute time.exe which can then execute beetle.exe.

7 Access control and Operating Systems (12 marks)

a) OSes in general (2 marks)
• Can Operating Systems provide access control without having to rely on the hardware? Why? (1 mark)

No. Operating Systems rely on the hardware structure of the CPU to implement access control. The core of the OS is the kernel, which operates on the highest privilege ring of the CPU. From this ring, the OS can access all hardware, including all memory. Applications on the other hand, run on a lower privilege CPU ring and must ask the OS (kernel) to access the memory for them. At this stage the OS can control if the application receives access to the memory or not (access control). If there were not different hardware rings on the CPU, then an application could send machine instructions to be executed that access any bank of memory, effectively accessing any file. In this way the OS must rely on the hardware to provide access control.

• How do Operating Systems prevent an application from overwriting another application's memory? (1 mark)

OS's prevent applications from overwriting another application's memory using the high privilege level OS kernel. The kernel allocates access to specific blocks of memory to an application. Applications cannot access memory that has not been allocated to it, and attempting to do so results in a segmentation fault (segsev).

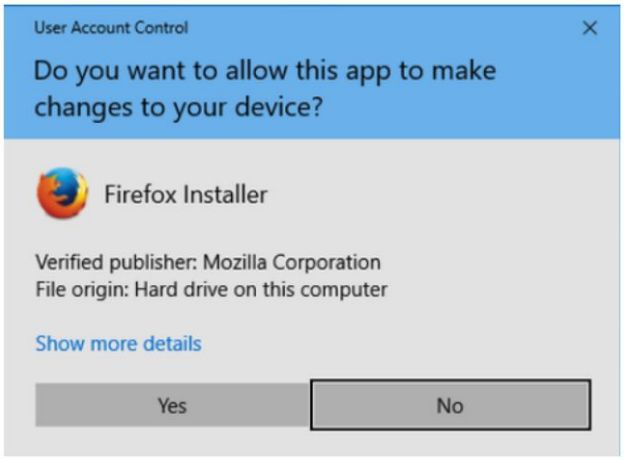
b) Linux (3 marks)
Consider the Linux operating system and how it handles file permissions.
• The mount command allows the Linux operating system to access additional block devices, this includes things like USBs, additional hard disks, CD-ROMS or even iso images. What are the permissions on this command? (1 mark)

The mount command is root owned, meaning you need root privileges to execute the mount command and mount a disk on linux.

• Write a script (using either Python or Bash) that would search the operating system and return a list of all files with similar permissions that the current user has access to (No "Permission Denied" strings in the output). Submit the script. Hint: if you attended the tutorial, this can be a one-liner. (2 marks)

To purely get the file names use $ **find / 2>/dev/null** this lists all files on OS and sends std errors to the null folder.

c) Windows (5 mark)
Figure 3 shows a dialogue of Windows 10. This is actually an implementation of a security model.



• What model is implemented here? (1 mark)

The security model implemented here is the Biba model. It is the opposite of the Bell-LaPadula model in that entities can only write down, and only read up.

• Name the rules of the model that cause this dialogue to appear. (1 mark)

The rule of the model that causes this dialogue to appear is that entities can only write down. This dialogue appears because firefox is attempting to write up, which goes against the Biba model. Because of this, the os seeks permission from the owner of the device, who presumably has high level authorisation.

Look into what an autorun.inf file is.
• How does a CD-ROM make use of an autorun.inf file? (1 mark)

A CD-ROM uses an autorun.inf file to launch an application when the CD is inserted into a drive. In this way it can autoplay a start menu without any user input.

• Compare and contrast how Windows handles inserting a CD-ROM to how it handles the case in Figure 3. (1 mark)

If the CD-ROM contains an autorun.inf file, it can use this immediately to run an application, even one that is defined on the disk. On the other hand, the case in figure 3 requires permission to be run.

• Compare and contrast how Windows handles inserting a CD-ROM to how Linux handles mounting a CD-ROM. (1 mark)

Linux has an additional level of 'safety' where it does not immediately mount and run an autorun file on a CD-ROM when it is inserted. A user has to actively mount the drive to access it.

d) The USB key and virtualized OSes (2 marks)
Assume you find a USB key on the floor. You pick it up, but distrust it. You start a VM to inspect its contents.
• Explain by referring to virtualization as access control: does this protect you against all risks involved by plugging in the USB key? Why? (2 marks)

An OS running in VM sees only its 'own' hardware. This raises the bar for attackers as they effectively have to "break out" of the VM to do anything malicious on your device and to its memory. This does not protect against all risks though of plugging in the USB key. A well written piece of malware can detect it is running on a VM and act differently.

8 Linux Privilege Escalation (6 marks)

When an attacker breaks into to a server, usually they initially get a least privilege shell/user access. Next the attacker attempt to obtain a higher privilege shell like the root through multiple ways. Conduct your own research and describe three methods attackers usually use for privilege escalation.

Exploiting SUDO rights. SUDO allows users to run programs with the security privileges of another user. If an attacker can find a user that has SUDO rights to a system, the attacker can compromise that account to gain root privileges. The attacker will exploit the user's SUDO rights to the command the generally harmless 'find', however, it can be used for command execution coupled with the root access.

Kernel Exploit. Attackers can take advantage of known vulnerabilities that arise in the Linux Kernel. By developing exploit code, they can apply and execute it on vulnerabilities to gain root access to the system. An example of this is the Dirty COW exploit found on older versions of the Linux Kernel. It allowed the attacker to use race conditions to make a read-only file writable, and, in conjunction with other exploits, access root.

Exploiting services running as root. Attackers are able to identify the programs that are running with root access and exploit them to get remote code execution with root access. For example, there is a known vulnerability in MySQL which can be exploited to execute arbitrary commands from the MySQL shell but with root access.

9 Nice images (8 marks)

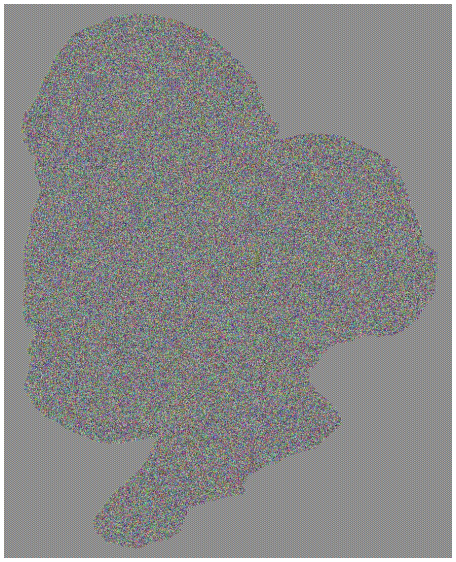
• Submit your code and the 'encrypted' image, i.e. the final PNG. In the report briefly explain how your code works.

```
#Define encryption key
key = b'INFO3616INFO3616'
#Generate cipher object using key
cipher = AES.new(key, AES.MODE_ECB)
#Pad data
paddedData = pad_list(dataIn)
#Encrypt data
encryptedData = cipher.encrypt(paddedData)
```

The code is very simple, and makes use of the PyCryptodome library to implement AES. The code generates a cipher object using the key 'INFO3616INFO3616'. The image is read as a continuous series of bytes. Given that the length of the image in bytes is not necessarily an integer multiple of 16, a pad_list function was defined which adds 'n' b'0' bytes to the data. See the function below:

```
def pad_list(data):
    requiredFill = (AES.block_size - len(data[0])) % AES.block_size
    for _ in range(requiredFill):
        data[0] = data[0] + b'0'
    return(data[0])
```

The data is then encrypted and outputted to file. The 'file reading code' is not included here, but is available in the submitted zip. See below for the input and output image files.



The ECB mode was clearly run given these outputs, as blocks of the same colour (same byte value) generate the same encrypted byte values, leading to a visible pattern for blocks of the same colour.

10 Poor man's AES (12 marks)

• State clearly what the weakness is and define (in pseudocode or prose) a way to break the cipher (4 marks).

There are a few key weaknesses in this implementation of AES. The most critical of which is that the nonces are clearly defined in code and follow a predictable pattern being series of 15 '0' bytes, followed by a byte of the character '0'-'9' depending on line. This is done by a modulus 10 operation. The code would be improved significantly if the nonces were only used once (the definition, number only used once). The fact that we know 10 input lines, and there are only 10 nonces means we can unencrypt the entire file instead of just the lines that we know inputs for..

The weakness tied to this is the use of the XOR relationship between encrypted nonces, and input data to create an encrypted output. The XOR equation can be rearranged (applied again to return to the original). Equation 1 implies equation 2 and 3.

$$1.\ \textit{encrypted text} = (\textit{plain text})\ XOR\ (E_{key}[\textit{nonce}])$$
$$2.\ E_{key}[\textit{nonce}] = (\textit{plain text})\ XOR\ (\textit{encrypted text})$$
$$3.\ \textit{plain text} = (E_{key}[\textit{nonce}])\ XOR\ (\textit{encrypted text})$$

In looking at the file, we know that the message starts with '################' etc (see plainStart in main.py). As such we know the plain text for line 1. Given the encrypted file we also know encrypted text. This lets us find $E_{key}[\textit{nonce}]$ for each line (using plainStart and plainEnd).

It is then possible to calculate the original plain text using equation 3. It is not even necessary to find the nonce encryption key to decrypt the text.

• Then implement it in code (8 marks).
See submitted zip of code.

11 Breaking Diffie-Hellman (6 marks)

• Describe the basic idea (2 marks).

The attacker can use a man in the middle attack to read every message they send. They do this by creating their own diffie-hellman partnership with Alice and Bob separately. When Alice sends Bob her private key, the attacker intercepts it and sends Alice the attacker's private key pretending to be Bob and vice versa. This allows the attacker to decrypt each message and then encrypt it and send it to the other party, potentially modifying it on the way.

• Write up the protocol flow (in the style as shown in the lecture), with the attacker breaking the key establishment (2 marks).

Alice	Attacker	Bob
Choose random value a < p	Choose random value c < p	Choose random value b < p
Compute X = g^a mod p	Compute Z = g^b mod p	Compute Y = g^b mod p
Send X	Send Z	Send Y
Receive Z	Receive X and Y	Receive Z
Compute k1 = Z^ a mod p	Compute k1 = X^ c mod p and Compute k2 = Y^ c mod p	Compute k2 = Z^b mod p
Alice obtains shared key k1 with attacker	Attacker obtains shared keys k1 and k2	Bob obtains shared key k2 with attacker

• Which key is Alice going to use? And which key is Bob going to use? (1 mark)

Alice will use her key shared with the attacker (k1), and Bob will use his key shared with the attacker (k2).

• Show why the attacker can get the plaintext of every message Alice and Bob send after the handshake (1 mark).

The attacker has the shared keys required to decrypt Alice and Bob's messages into plaintext as they come in, the attacker will then re-encrypt the plaintext with its other shared key to send to the other party.

12 Hybrid cryptography in Python (12 marks)

In the report explain your code and how to run it.

The code works by first generating an RSA key for both Bob and Alice, this includes both a public and private key. Alice is then able to create a cipher with a random symmetric key to use to encrypt her message to Bob. Alice sends Bob the encrypted message, as well as the symmetric key which has been encrypted using her public key. Bob then has to use his key and Alice's public key to recreate the cipher they are sharing. Bob is then able to decrypt the symmetric key Alice has sent, and then use that to decrypt the encrypted message. Bob then receives Alice's message without an attacker being able to access it.

To run the code open the terminal and enter the command 'python3 hybrid.py'. If the code is successful at encrypting and decrypting it will output the console message "Hi Bob, it's Alice\nIt worked!".