****

Applied Cryptography   
Assignment 2

|  |  |
| --- | --- |
| **Module Code** | ST2502 |
| **Assignment Done By** | Sng Jay Kai (p2003836)  Lu Hao Xuan (p2026086)  Noel Tong Yao Wei (p2026057) |
| **Class** | DISM/FT/1B/03 |
| **Lecturer** | Mr How Kiam Cheng |
| **Submission Date** | 31/1/2021 |

### 

**ACKNOWLEDGEMENT**

This team would like to express our utmost appreciation and gratitude for our lecturer, Mr How Kiam Cheng, from the School of Computing, Singapore Polytechnic, for providing us with the guidance and knowledge needed for this proposal.

**Table of Contents**

[**1 Introduction**](#_a9o8p8cdd6z6)4

[1.1 Background](#_5yw3164uc0og) 4

[1.2 Objective](#_ioo54orvb5ku) 4

[**2 Current System**](#_rt6hrp4ki3rr)5

[2.1 Outline](#_oqwbfoywz7pm) 5

[2.2 Application Flow](#_w2mk7yrs9wsx) 5

[2.3 File Structure](#_19k6klbs6epw) 8

[**3 Assumptions**](#_ym8levap6sdz)9

[**4 Attack Scenarios & Countermeasures**](#_d2m9josucylk)10

[4.1 Day-closing information](#_7grgyrsza2bx) 10

[4.1.1 Eavesdropping/Packet Sniffing](#_nikcazqj7bfu) 10

[4.1.2 Man-In-the-Middle attack/socket hijacking](#_3ehl00v1eia5) 11

[4.2 Menu-of-the-day](#_2wcyi0ouvltr) 12

[4.2.1 Man-In-the-Middle attack/socket hijacking](#_wyu70i7ae72b) 12

[4.3 Physical Intrusion](#_jxq4sylcj7kj) 12

[**5 Proposed System**](#_9kc2n3b2grig)13

[5.1 Use of Transport Layer Security (TLS)](#_5tchd0tmtu39) 13

[5.2 Arbitrated Digital Signatures](#_frn91cn79o9p) 15

[5.3 Upgrading WIFI connection](#_cd9sg84hwgom) 16

[5.4 Diagram of the proposed system](#_x7atz0bb5r6i) 16

[**6 Conclusions & Additional Considerations**](#_4zjtofz3pprg)19

[**7 Planned Task Allocation**](#_domfr5te5qzr)20

[**8 Individual Reflection**](#_q9y4cnperqt8)21

[**9 References**](#_u52q7k7wynl9)23

[**Appendix**](#_qbdrbr95kqi1)25

[Simplified Risk Assessment Form](#_mq1jn4x1thd4) 25

### 

# 1 Introduction

## 1.1 Background

Singapore Polytechnic has decided to set up additional automated menu systems (Singapore Polytechnic Automated Menu 2 - SPAM2) outside of SP. However, due to limited budget, public WIFI, namely the Wireless@SG programme owned by the Infocomm Media Development Authority (IMDA), will be used.

## 1.2 Objective

The objective of this proposal is to present our findings on the structure & risks of the existing system including possible attack scenarios and countermeasures, as well as our final proposed system that will ensure confidentiality, integrity, and non-repudiation of important data being exchanged and stored throughout the SPAM 2 system.

# 2 Current System

## 2.1 Outline

The SPAM2 system requires three key components to function, which are the SPAM2 Server, Internet over Wireless@SG, and the client. The SPAM2 server and client communicate through sockets using a TCP connection.

When the client program at an outlet requests for the daily menu at the start of the day, the SPAM2 server will receive the command and send a file (menu\_today.txt) via the Internet over Wireless@SG. Similarly, at the end of the day, the client will request the to transfer the local file content of day\_end.csv which contains information on the sales made for the day.

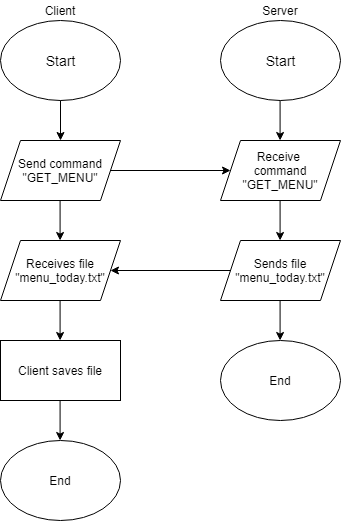
## 2.2 Application Flow

The application flow is as follows:

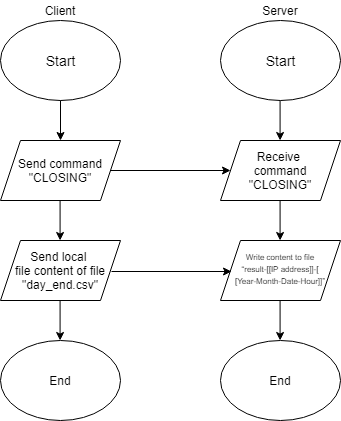
|  |  |
| --- | --- |
| **Prior to operations** | |
| Server starts | Client starts |
| **Operation 1 (Start Day / 10:00 am)** | |
| - | Client sends command “GET\_MENU” |
| Server receives command Server sends file “menu\_today.txt” | - |
| - | Client receives file “menu\_today.txt”  Client saves file |
| **Operation 2 (End Day / 10:00 pm)** | |
| **-** | Client sends command “CLOSING”  Client sends local file content of file “day\_end.csv” |
| Server writes file content to file  “result-[[IP address]]-[[Year-Month-Date-Hour]]” | - |

The application flow is better illustrated by flowcharts.

The flowchart for Operation Start Day at 10:00am is as follows:



The flowchart for Operation End Day at 10:00pm is as follows:

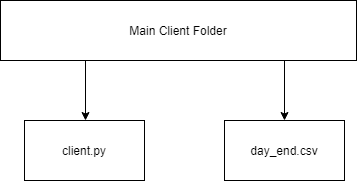


## 

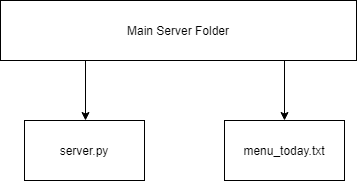
## 2.3 File Structure

The provided source code can be divided into two main groups. The server folder contains files pertaining to the SPAM2 server while the client folder contains files pertaining to the client.

The file structure for the client is as follows:



The file structure for the SPAM2 server is as follows:



# 3 Assumptions

With regards to the existing system, we have made the following assumptions:

1. The SPAM2 system runs on the public network Wireless@SG
2. The server and client machine can support TLS
3. The arbiter is trustable and would not modify the data being sent

# 4 Attack Scenarios & Countermeasures

## 4.1 Day-closing information

Day-closing information would require confidentiality, integrity, as well a non-repudiation.

### 4.1.1 Eavesdropping/Packet Sniffing

The data is not encrypted when they are being sent to and from the client and server. This means that as long as attackers are able to intercept the packets while they are in transit, for example using Wireshark while in the same network, all the data would be visible to them in plain text. These tools are widely available for free, and require little to no knowledge to use, making this attack a particularly common one. Since the outlets would be connected to a public WIFI, anyone can connect to the same network and carry out packet sniffing.

To counter this, sensitive data, such as the day-closing information should be encrypted to maintain the data’s confidentiality. A possible encryption method for this is the Advanced Encryption Standard (AES). AES is a symmetric encryption algorithm which uses 128, 192 or 256-bit keys, meaning that the same key is used for both encryption and decryption. AES has 10 rounds for 128-bit keys, 12 rounds for 192-bit keys and 14 rounds for 256-bit keys. This means that as the length of the keys used increases, the time taken for the plain text to be encrypted using AES also increases. However, the time taken for attackers to brute-force the key increases exponentially as well, making the use of longer keys much more secure. This trade-off between efficiency and security would be necessary, especially since the day-closing information is confidential and sensitive information, hence we recommend using AES with a key size of 256 bits.

By using AES in Counter (CTR) mode, we can also increase efficiency. This encrypts the counter with a unique key, and the resulting plaintext gets XOR with the plaintext, resulting in the ciphertext. This means that the process of both encryption and decryption can be carried out with parallelization, increasing the speed. Since the encryption of each block does not rely on the previous one, should data in a block get corrupted, the other blocks would still be able to be received and read normally after decryption. However, when using AES-CTR, we need to ensure that each subkey and counter used is unique. Should the same subkey or counter get used in the encryption of two or more blocks, portions of the output stream can be recovered, compromising the confidentiality of the data.

However, this also provides a new problem- key distribution. Both the server and the client require the use of the same key to decrypt and encrypt text with AES. This means that they would need to have a channel to securely send over the AES key to the other side in order for the text to be decrypted, or the attack would be able to intercept the key and use it to decrypt the data sent. To solve this, we can use Rivest–Shamir–Adleman (RSA). RSA is an asymmetric cipher that uses a public key to decrypt, and a private key to encrypt. This means that the server can store a private key known only to itself, and transmit a public key for the outlets to use to decrypt the AES keys being sent. Even if the attackers were to get hold of the public key, they can only use it for decryption and not encryption, keeping the AES keys secure, and by extension, the day-closing information as well. We suggest only using RSA for encrypting of the AES keys for secure key distribution as RSA takes a much longer time to execute than AES.

### 4.1.2 Man-In-the-Middle attack/socket hijacking

Since both the server and the client runs on SOCK\_STREAM sockets, if an attacker were to be able to successfully hijack the socket, for example via TCP hijacking and IP spoofing, the attacker can then send forged packets to the server. Combined with Denial-of-Service (DOS) attacks to prevent the actual client from communicating with the server, attackers can send false information to the server, for example sending the sales of the day to be much higher than the actual sales. To prevent this, as well as for non-repudiation protection, arbitrated digital signatures can be used. Arbitrated digital signatures involved a third party, also known as a trusted arbiter, to receive and sign messages from the sender. They would then validate the content and the origin of the data sent, followed by dating the message and indicating that it has been verified before sending it to the recipients. However, this method requires complete trust from both the sender and the recipient that the arbiter would do as intended, as well as not modifying any of the data sent. Since the data would be visible to the arbiter, the data being sent should be encrypted first, before signing with their private key and sending to the arbiter. This way, the arbiter would only view the encrypted data, which they would then be validating. After which, the recipient would then validate the arbiter’s signature, before finally decoding the data.

## 4.2 Menu-of-the-day

### 4.2.1 Man-In-the-Middle attack/socket hijacking

Similar to the Day-closing information, the menu-of-the-day is vulnerable to socket hijacking as well. After being able to successfully send forged packets over to the client, the attacker can then send false menus to the outlets for the outlets to display. To combat this, we can use Hash based Message Authentication Code (HMAC) with SHA-1. HMAC produces a Message Authentication Code (MAC) based on a secret key, as well as a hashing algorithm, which in this case would be SHA-1 due to its speed, as well as its security. As long the key used is sufficiently long, the time taken to brute force the key would be infeasible to execute. Since HMAC uses hashes, which cannot be reversed, knowing a collection of the hashes would not help the attacker in finding out the secret key used. The recipient then uses the same private key to carry out the same hashing process. Since the MAC generated relies on the private key, as long as the private key remains uncompromised and secured, even if the attacker were to send a message with a valid MAC generated with a different private key, by using the sender’s public key, the resulting MAC generated would be different from the one appended to the message and the recipient would then know that the message has been altered or does not originate from the sender.

## 4.3 Physical Intrusion

If an attacker were to gain physical access to the system, the use of keys and encryptions would not be able to prevent them from sending false information or obtaining the menu data. This is since all the keys required would be stored on the machine itself. To counter this, a login system should be implemented. The client programs should request for a username and password, which would then be encrypted by RSA, and then sent over to the server for verification. The server can then verify the login credentials.   
It will then send over data to the client or receive data from the client.

# 5 Proposed System

## 5.1 Use of Transport Layer Security (TLS)



For our proposed system, we will be using Transport Layer Security (TSL) protocol to protect information transmitted between our client and server. TSL is a cryptographic protocol and the successor to Secure Sockets Layer (SSL) which was an older method that clients and servers used to establish a secure connection. It provides data authentication, integrity, and confidentiality all in a single protocol, hence attackers cannot read or modify data that is being transmitted.

To get a TLS certificate we can simply buy one from a certificate authority like this website (<https://www.websecurity.digicert.com/en/sg/ssl-certificate>). If the company does not wish to spend money, they can create a self-signed certificate; however, there are risks that come with it such as fake certificates created to impersonate servers.

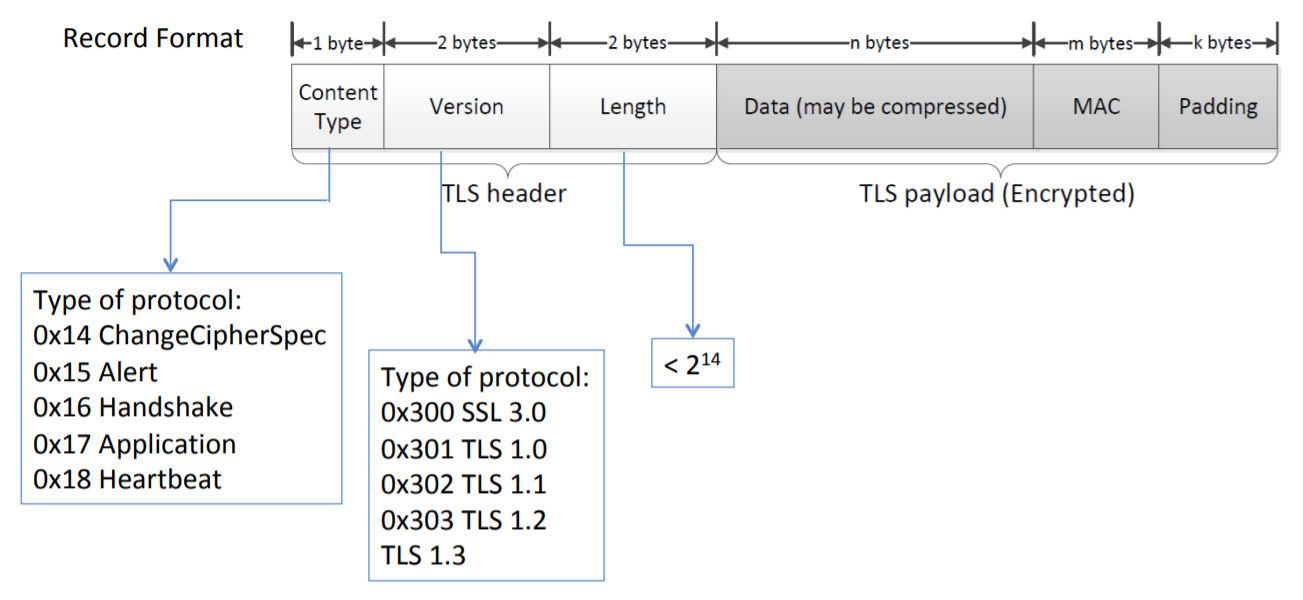
How TLS protects data:

* Confidentiality - TLS uses both symmetric and asymmetric encryption to ensure that the data being transmitted is encrypted (see how in TLS handshake below), hence attackers that intercept the message cannot understand it and decrypting the message takes an infeasible amount of resources.
* Integrity - TLS uses a message authentication code, a message authentication code uses a hash function with a shared key, the receiving end will then regenerate and compare the MAC to ensure data is unaltered
* Authentication - The authentication in TLS takes place during its handshake. TLS uses digital certificates and public key encryption to authenticate users. Digital certificates are checked with certificate authority for validation. The public key encryption provides authentication as the handshake can only be processed if the public/private key pairs that the users possess are correct, then the secret key can be used to confirm the connection by sending the “finished” messages.

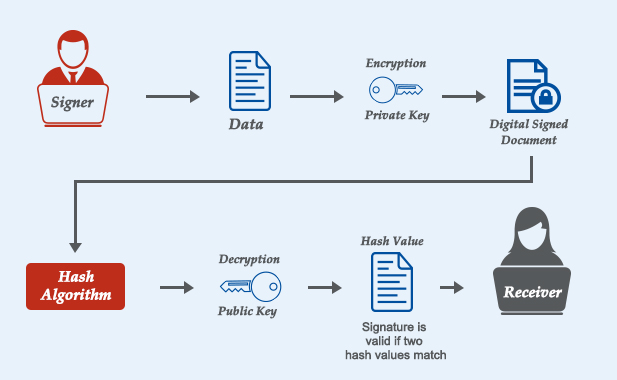
TLS uses a handshake to create a secure connection, the handshake is used to establish trust and then decide on a secret key for encrypting/decrypting the messages:

1. Client sends a “client hello” message that includes information such as the TLS version and the cipher suites supported
2. The server the client is connecting to responds with a “server hello” message that includes information such as the cipher suite chosen, the server’s digital certificate and the server’s public key
3. Client receives and verifies the digital certificate by contacting the server’s certificate authority, hence proving the authenticity of the server
4. Client generates a random string of bytes (pre-master secret), encrypts with server’s public key and sends it to the server
5. Session keys are generated so that the server and client can communicate with data being encrypted
6. Client sends the server a “finished” message and server responds with “finished” message and the secure connection is established

This diagram shows what is included when data is being transmitted using TLS:



## 5.2 Arbitrated Digital Signatures

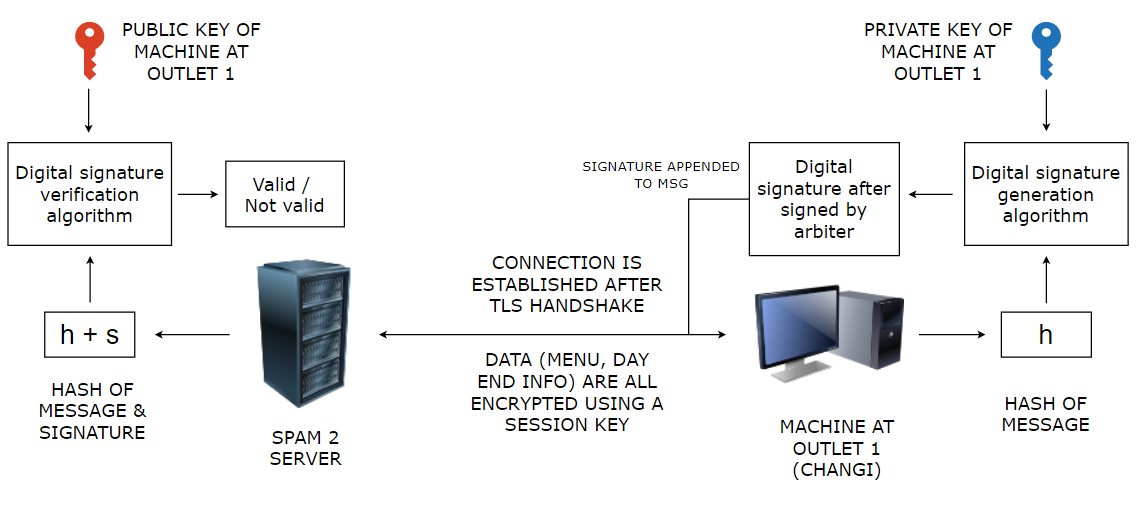
****The communications between the server and client in TLS are only using a shared secret key (symmetric encryption), hence there is no non-repudiation as the sender can deny the message originated from him/her. To solve this problem, the system can use arbitrated digital signatures which introduces a third party into the whole process known as an arbiter, the arbiter then validates the content and origin from subject, message and signature.

After validation, the arbiter then adds his/her own signature and a timestamp to the message. This provides non repudiation as there is an extra layer of authentication through the arbiter and the arbiter can step in and verify the origin of the message if a user denies it. It is better than direct digital signature as in direct digital signature, the sender can simply deny the validity of the signature and say his/her private key was lost or stolen.

## 5.3 Upgrading WIFI connection

Assuming that the current Wi-Fi that the system is using Wireless@SG, the connection would be unsecured as Wireless@SG is obsolete and will be discontinued after 12 October 2020. A better solution is to use the new Wireless@SGx WIFI as according to M1, Wireless@SGx network is more secure as data is encrypted and it uses industry-based security standards and authentication (i.e. IEEE 802.1x and Wi-Fi Alliance) (M1, n.d.), more information can be found on the official M1 website here (<http://www.m1net.com.sg/wireless-sg/faqs.html>). Future plans can also include purchasing a business Internet access plan from a telecommunication company such as Singtel, M1 or Starhub to ensure a more secure and reliable connection.

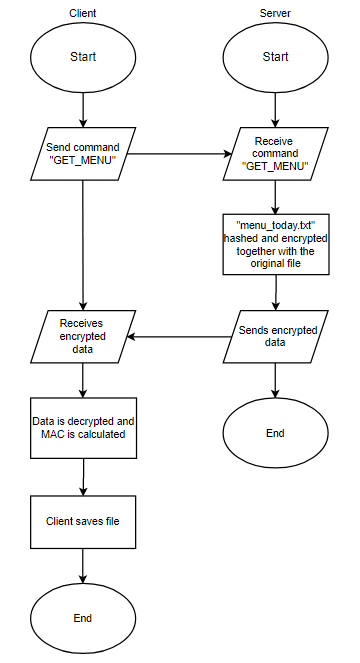
## 5.4 Diagram of the proposed system



Note: In this diagram digital signatures are only used for client to server communication to ensure non repudiation of day\_end.csv

The following pages contain the flowcharts illustrating the application flow after implementing the proposed system.

The flowchart for Operation Start Day at 10:00am is as follows:

****

The flowchart for Operation End Day at 10:00pm is as follows:

# 

# 6 Conclusions & Additional Considerations

In conclusion, we have identified three major vulnerabilities that can compromise the Confidentiality, Integrity, Non-repudiation, and even Availability of the SPAM2 system. We have also proposed countermeasures to combat these vulnerabilities, as well as additional security features that will help to harden the security of the SPAM2 system. It is crucial that security features similar to those that we have suggested be implemented as if the security were to be ignored to save costs, time or for convenience, attackers would have many methods to compromise the security of the system. This would in turn lead to the data being compromised.

However, it is important to remember that no amount of security is enough to permanently keep attackers from successfully breaking into the system. It is important to have best practices in use, such as using strong passwords, key rotation, as well as storing credentials securely to reduce the probability of successful attacks occurring. The company should also have measures put into place to reduce the damage dealt when a successful attack occurs.

Furthermore, it should be noted that humans are the weakest link in cybersecurity. Singapore Polytechnic should train its staff to ensure compliance with best practices such as those listed above and implement policies to ensure that staff do not mishandle data.

# 7 Planned Task Allocation

The work for this assignment including code and proposal was split fairly among the members of the team. More details on code allocation will be found in the contributions.txt file submitted with the code package.

Proposal Proposed Task Allocation:

|  |  |
| --- | --- |
| **Hao Xuan**  (Will do more code) | Section 5: Proposed System  Appendix 1: Simplified Risk Assessment  Editing of all sections |
| **Jay Kai**  (Will do less code) | Section 1: Introduction  Section 2: Current System  Section 3: Assumptions  Section 6: Conclusion & Additional Considerations  Section 8: References  Appendix 1: Simplified Risk Assessment  Editing & Formatting of all sections |
| **Noel**  (Will do more code) | Section 3: Assumptions  Section 4: Attack Scenarios & Countermeasures  Section 8: References  Appendix 1: Simplified Risk Assessment  Editing of all sections |

# 8 Individual Reflection

**Hao Xuan:** This assignment was very fun and challenging, it allowed me to apply the things we learned in lessons into real-life situations, which reinforced my knowledge of the cryptography methods we learned. I oversaw highlighting the illustrated proposed systems and it made me research into modern encryption and secure connection methods such as TLS, improving my understanding of cryptography. It also tested my ability to implement the code we learn in lessons into client and server applications, in lessons we do not use servers and clients but instead use the code we learn to encrypt simple messages and images to observe the output. It also helped me further understand the basic requirements of message security such as integrity, authentication, confidentiality, and non-repudiation through the different cryptography methods such as asymmetric encryption and hashing. Overall, I feel that if I were to encounter this situation or even help a company harden the security of a company’s system again, I would be much more experienced and capable.

**Tong Yao Wei, Noel:** The assignment has allowed us to gain deeper insight into modern applications of cryptography. The need to actively identify vulnerabilities and possible attacks, as well as their solutions was interesting as it is not something I usually do. In doing so, I have learnt more about how to identify potential vulnerabilities in a system, as well as measuring their impact, which can come in helpful in the future, be it for other modules, or in real-life applications. Identifying and coming up with a suitable countermeasure for the vulnerabilities has also helped me to deepen my familiarity with the different cryptography methods as I had to read up on the different methods, and then compare them to select a suitable solution. At the same time, it has also helped me to learn how to better collaborate with my teammates, as we had to separate our tasks, while at the same time ensuring that our information did not contradict each other. After the assignment, I feel that I have gained a better understanding of the different encryption methods, learnt how to be a better teammate, as well as better identifying vulnerabilities.

**Sng Jay Kai:** Despite being concerned at completing the assignment due to the large amount of assignments from all modules, I am very glad for this opportunity to sharpen up my knowledge on different aspects of cryptography, building upon the knowledge learnt in class. Although we may use systems like the SPAM2 system in daily life, we rarely stop to reflect on the inner workings and what really secures these systems. I especially enjoyed learning more about how attackers can conduct Man-in-the-middle (MITM) attacks on sockets, and how to counter such attacks. I must also thank my teammates for being great to work with as we had no conflicts at all throughout the project. Despite not being specifically in charge of certain sections, we all edited each other’s work and, in that process,, learnt from each other not just regarding the report, but also how to present and write the report. In conclusion, I am happy to report that I have gained a better understanding of concepts set in class, as well as a better team player to work effectively with others.

# 

# 9 References

Cloudflare. 2021. [online] Available at: <https://www.cloudflare.com/learning/ssl/transport-layer-security-tls/> [Accessed 25 January 2021].

Cloudflare. 2021. [online] Available at: <https://www.cloudflare.com/learning/ssl/what-happens-in-a-tls-handshake/> [Accessed 30 January 2021].

Comodosslstore.com. 2021. What is Digital Signature and How It Works?. [online] Available at: <https://comodosslstore.com/blog/what-is-digital-signature-how-does-it-work.html> [Accessed 26 January 2021].

Cse.iitm.ac.in. 2021. Department of Computer Science & Engineering. [online] Available at: <http://cse.iitm.ac.in/> [Accessed 17 January 2021].

Ibm.com. 2021. IBM Knowledge Center. [online] Available at: <https://www.ibm.com/support/knowledgecenter/en/SSFKSJ_7.5.0/com.ibm.mq.sec.doc/q009940_.htm> [Accessed 19 January 2021].

Infocomm Media Development Authority. 2021. Wireless@SG - Infocomm Media Development Authority. [online] Available at: <https://www.imda.gov.sg/programme-listing/Wireless-At-SG> [Accessed 28 January 2021].

Internet Society. 2021. What is TLS & How Does it Work? | ISOC Internet Society. [online] Available at: <https://www.internetsociety.org/deploy360/tls/basics/#:~:text=Transport%20Layer%20Security%20(TLS)%20encrypts,card%20numbers%2C%20and%20personal%20correspondence> [Accessed 22 January 2021].

M1net.com.sg. 2021. M1 Net Wireless@SG Announcement | M1. [online] Available at: <http://www.m1net.com.sg/wireless-sg/faqs.html> [Accessed 20 January 2021].

ProtonMail Blog. 2021. What is a TLS/SSL certificate, and how does it work?. [online] Available at: <https://protonmail.com/blog/tls-ssl-certificate/> [Accessed 23 January 2021].

TechRepublic. 2021. TCP hijacking. [online] Available at: <https://www.techrepublic.com/article/tcp-hijacking/> [Accessed 22 January 2021].

# 

# Appendix

## Simplified Risk Assessment Form

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Risk** | **Likelihood** (e.g. High, Medium, Low) | **Severity**  (e.g. High, Medium, Low) | **Response Strategy**  (e.g. Avoid, Transfer, Mitigate or Accept) | **Actions required** | **Due Date** |
| Attackers eavesdropping on the data transmitted, e.g. attacker reads the sales file that the outlet sends back, the information can be sold to competitors and cause loss of profit (Confidentiality) | High | High | Avoid | Use strong encryption algorithms such as AES, so attackers are unable to read the message and decrypt it | 14th Feb 2021 |
| Attackers physically breaking into the outlets to access the menu or send false data to the server (Confidentiality and Integrity) | Low | High | Mitigate | Implement the need of a password when accessing the system so that without a password, attackers are unable to send or obtain data | 14th Feb 2021 |
| Attacker intercepting connections and sending or requesting for data (confidentiality and integrity) | Medium | High | Avoid | Use methods to verify sender, such as HMAC and arbiter digital signature to prevent attackers from pretending to be a client | 14th Feb 2021 |
| Phishing, where attackers pretend to be someone with authority to convince the employees to give away sensitive data, such as account passwords | High | Medium to High | Avoid | Conduct training for employees to teach them to spot phishing attempts, to never give out passwords and implement the use of anti-spam to filter out phishing emails | 14th Feb 2021 |
| Malicious access points, where they set up access points for employees to connect to, for example naming the SSID Wireless@SG to trick employees, and then log all packets that passes through | Medium | Medium | Avoid | Conduct training for employees to carefully check for spelling mistakes when connecting to the wireless connections to prevent checking to malicious access points | 14th Feb 2021 |

The following is the risk matrix we used to determine the severity and likelihood columns:

