**ACG CA2 Report**





*Module Code:* [***ST2504***](https://esp.sp.edu.sg/webapps/blackboard/execute/launcher?type=Course&id=_46916_1&url=)

*Class: DISM/FT/1B/04*

*Name: Lee Yi Terng (1904189)*

*Name: Sean Chye (1904291)*

Table of Contents

[1. Introduction 3](#_Toc32139093)

[1.1 Overview 3](#_Toc32139094)

[1.2 Assumptions 3](#_Toc32139095)

[1.3 Contents of Document 3](#_Toc32139096)

[2. SPAM2 System 4](#_Toc32139097)

[2.1 Illustration 4](#_Toc32139098)

[2.2 Implementation 4](#_Toc32139099)

[3. Attack Scenarios and Countermeasures 4](#_Toc32139100)

[3.1 Man-in-the-Middle on Menu Information 4](#_Toc32139101)

[3.2 Eavesdropping on Day-Closing Information 5](#_Toc32139102)

[3.3 Non-Repudiation of Day-Closing Information 6](#_Toc32139103)

[3.4 Encrypting Static Day-Closing Information 7](#_Toc32139104)

[4. Proposed System 7](#_Toc32139105)

[4.1 New features 7](#_Toc32139106)

[4.2 Complete System 8](#_Toc32139107)

[4.3 Illustration 8](#_Toc32139108)

[4.4 Additional Considerations 9](#_Toc32139109)

[5. Conclusion 10](#_Toc32139110)

[6. Miscellaneous 10](#_Toc32139111)

[6.1 Planned Task Allocation 10](#_Toc32139112)

[6.2 Reflection 10](#_Toc32139113)

# 1. Introduction

## 1.1 Overview

This report was written to discuss the security vulnerabilities and countermeasures for the SPAM2 system.

The SPAM2 (Singapore Polytechnic Automated Menu 2) system was met with overwhelming demand and hence has decided to expand by setting up additional outlets outside SP, using public WIFI such as Wireless@SG.

As the system is planned to be deployed in a public environment with an unsecured wireless connection (Wireless@SG), it is crucial to plan security implementations in order to ensure the security and privacy of our data.

## 1.2 Assumptions

This proposal was created with the following factors in mind:

Menu-of-the-day information to be protected with integrity in mind during transit;

Day-closing information to be protected with confidentiality and non-repudiation;

Data at rest to be secured for confidentiality;

Wireless@SG connection is open and unsecured.

## 1.3 Contents of Document

SECTION 2: SPAM2 System

This section details the current implementation and processes of the SPAM2 System, as well as an illustration to better, depict the system’s implementation

SECTION 3: Attack Scenarios and Countermeasures

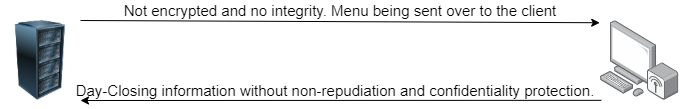
This section discusses various attack scenarios, attacker motivations and capabilities, the potential impact of that attack as well as possible countermeasures to combat the threat.

SECTION 4: Proposed System

This section sums up the final proposed system with an overview of the new features implemented, the complete system configurations, additional considerations as well as an illustration.

# 2. SPAM2 System

## 2.1 Illustration



## 2.2 Implementation

The current SPAM2 system does not contain any security features, and merely covers the basic processes of sending menu and day closing information between server and client.

This is very dangerous as attackers can exploit this lack of security and do malicious actions such as stealing business data, alter menu information and many more.

It is pivotal for the success of the SPAM2 system that we implement sufficient security features to protect the well-being of the system and the business.

# 3. Attack Scenarios and Countermeasures

## 3.1 Man-in-the-Middle on Menu Information

The SPAM2 system sends menu-for-the-day information when the connection between server and client socket is first initiated. This leaves the information vulnerable to integrity attacks like Man-in-the-Middle attacks. The attacker could intercept this unencrypted data and alter it to his own liking. He could inject his own contents or carry out a replay attack.

This can cause panic or jeopardize business operations by sending false menus or by sending an invalid menu. The attacker’s motivation could be as a business competitor who wants to see the SPAM2 system fail or become unusable by customers.

Should this happen, it would be catastrophic to the business operations of the SPAM system as it is not that hard to do if the attacker is well-trained and well-resourced.

The management has also stated the necessity of the menu-of-the-day information requiring integrity protection in transit. Hence, countermeasures have been devised to solve this issue and fulfil the requirement.

The countermeasure proposed for this problem with integrity is to implement hashing in SHA-256. SHA-256 can be used to verify the integrity of messages by using a one-way function to create a hash of a message. Although it provides no encryption for confidentiality protection, the hash can be used for integrity protection, which is required in this scenario.

The menu information would be hashed server-side before being sent to the client, where the client will recalculate the hash when received to verify if the data sent has been altered.

This helps prevent the integrity of the menu information from being lost due to a Man-in-the-Middle attack or injection attack.

## 3.2 Eavesdropping on Day-Closing Information

At the end of the day, the client socket sends Day-Closing information back to the server socket. The information sent is unencrypted, which is very dangerous as attackers can easily eavesdrop on it. The information being sent is sensitive and contains very important business data that we cannot afford to lose or be discovered by business competitors. Furthermore, management has also stated that day-closing information is required to have confidentiality protection.

Competitors who want to eavesdrop and steal our business information can easily do it by opening Wireshark and sniffing the packets sent. As the SPAM2 system is connected to an unsecured Wireless@SG connection, the packets sent are not encrypted and its contents can be seen in plaintext. It is not hard to carry out and extremely detrimental to the SPAM2 system should it occur.

If this information is stolen or eavesdropped on, competitors can use this business data to devise strategies to become better than us. This goes against the interest of the SPAM2 system and its management. It is important that this problem is countered adequately in order to prevent the likelihood of it happening.

In order to counter this issue, encryption can be used to ensure confidentiality of the day-closing information sent. Before data is sent, it should be encrypted on the server-side using Rivest–Shamir–Adleman (RSA) encryption algorithm in order to ensure the data cannot be read.

RSA is a very secure public-key encryption algorithm that uses two keys, one public and one private. The private key is only known to the owner and the public key is known to everybody. The private key is used to decrypt messages while the public key is used to encrypt messages. It helps ensure secure communications without having to trust a Key Distributing Centre with your key, leaving you to be the one responsible for your own private key. It is also easy to encrypt and decrypt for the intended sender and receiver, and the decryption key is hard to find. RSA helps to ensure the secrecy and confidentiality of messages by encrypting and decrypting.

The server and client will have their own private key, and their public key will be known to each other. When day-closing information from the client is going to be sent, the data will be encrypted using the server’s public key before sending it over the connection. Once the server has received the encrypted data, it will use its own private key to decrypt it.

This helps ensure that even if the data is stolen or eavesdropped on, the attacker would have to decrypt the plaintext to read it. It is also very hard for the attacker to decrypt it as the server’s private key is only known to itself and very hard to deduce even with clues from the ciphertext.

## 3.3 Non-Repudiation of Day-Closing Information

At the end of the day, the client system will have to send Day-Closing information to the server. During the process, a hacker with malicious intent can send false Day-Closing information which might cause a problem to the SPAM2 system. Alternatively, the server can ensure that the right file is sent by the right sender, the client. The SPAM2 system cannot afford to record false sales and wrong information as it might affect future developments and plans for the system.

To address this issue, Non-repudiation will be implemented into the Day-Closing information. The purpose of implementing it is to make sure that the correct Day-Closing information is received by the server from the correct sender.

For this situation, we will be using digital signatures to ensure that non-repudiation is achieved. Before the client system sends the Day-Closing information through the sockets, the message will be hashed with SHA-256 and encrypted before being appended to the original message. This will create a unique digital signature. When the data is sent to the server, the server will validate the signature by hashing the message and encrypting it. If the server compares the keys and they match, the server will send back an acknowledgement with its own digital signature appended at the back. Once the client receives the acknowledgement and verifies the signature, it will be able to use that acknowledgement together with its signature to vouch that the correct Day-Closing information has been sent and received by the server.

Digital signatures are unique as the messages will go through a hashing function and an encryption function which includes a key. Only the intended recipient will know the key and create the same digital signature to compare. Therefore, this way non-repudiation is achieved as now, the server cannot deny that the client did not send the correct information as the client will have the acknowledgement from the server with the server’s signature appended to the end of it.

## 3.4 Encrypting Static Day-Closing Information

After a day passes and the Day-Closing information is sent from the client to the server, the information will be decrypted by the server for verification and storage purposes. The server then stores the information into a day-closing file in plaintext. This will leave the information vulnerable and allow hackers or business rivals to discover the information should they be able to steal the file.

To prevent this from happening, before the data is stored into the server storage system, it will go through a round of encryption: The Advanced Encryption Standard(AES). Therefore, in that case, the data will not be stored in the file as plaintext, reducing the chances that a person infiltrating the system will be able to decrypt the data and gain access to the Day-Closing information.

The AES is a block cipher capable of handling 128-bit blocks, using key sized at 128,192 and 256 bits. The AES is a symmetric key cipher (Also known as secret-key ciphers) uses the same key for encrypting and decrypting. All key lengths are deemed sufficient to protect classified information requiring either 192- or 256-bit key lengths. There are 10 rounds for 128-bits keys,12 rounds for 192-bit keys and 14 rounds for 256-bit keys. A round consists of several processing steps that include substitution, transposition and mixing of the input plaintext and transforming it into the final output of ciphertext.

As the AES is designed to resist known attacks and its large key size makes it resistant to brute force attack, there is a very low chance that the attack will be able to crack the key and decrypt the ciphertext if the attackers manage to steal the data file.

This adds confidentiality protection to day-closing information being stored in a file. The AES standard is highly secure and very resistant to attacks. Thus, it justifies the use of AES encryption to protect and encrypt the day-closing information before it is stored in the file.

# 4. Proposed System

## 4.1 New features

New security features implemented into SPAM2 System:

Integrity protection of menu-of-the-day information;

Encryption of Day-closing information in transit;

Non-repudiation protection for day-closing information in transit;

Encrypting stored static day-closing information;

## 4.2 Complete System

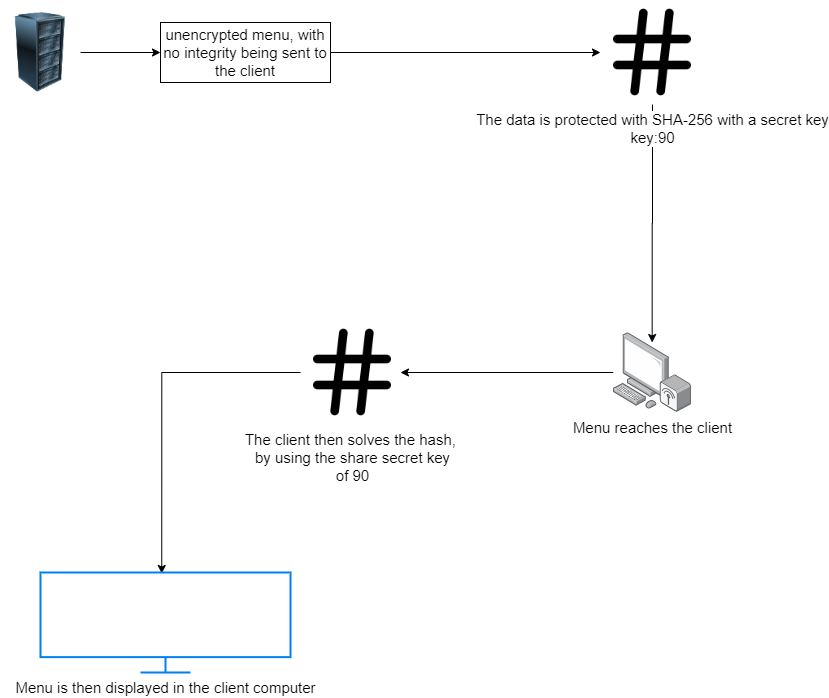
During the initiation of the socket connection between server and client, the server will send over menu-of-the-day information to the client. This information will be hashed server-side before being sent. Upon receipt of the hashed information, the client will perform a recalculation of the hash to ensure the integrity of the menu information is not compromised.

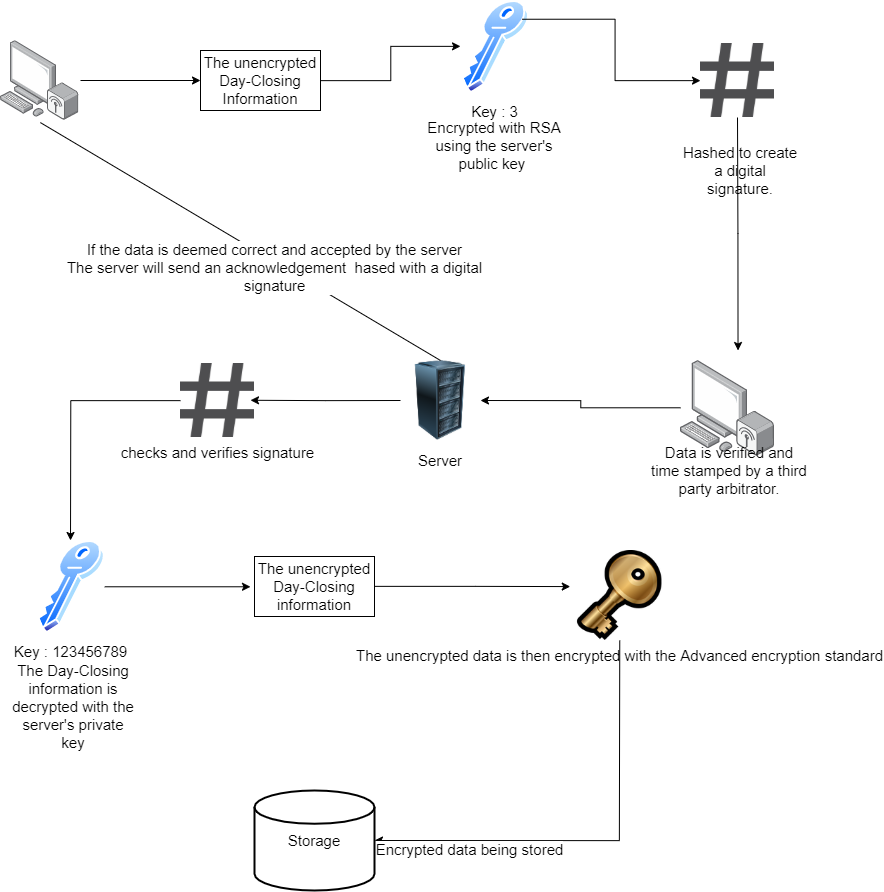
At the end of the day, before the client sends the Day-Closing information to the server the file and its contents will be encrypted with RSA using the server’s public key and hashed to create a digital signature, which will then be time stamped and verified by a third-party arbitrator.

After the server receives the Day-Closing information, the server will then decrypt it using its own private key and verify the contents, before encrypting it with the AES algorithm. The server will then store the encrypted data into a file for storage purposes.

## 4.3 Illustration

This is the process in which the server will be sending the Menu for the day to the client system.





This is the process of the client sending the Day-Closing Information through the third arbitrator and to the server system.

## 4.4 Additional Considerations

For these countermeasures to be effective in securing the SPAM2 System, there are some important pointers to take note.

Firstly, encryption keys should be well protected and not shared without reason. There should be no need to share private keys under any scenario, and should these keys be leaked, it could spell disaster for the system.

Secondly, the connection should be closely monitored to ensure there is no eavesdropping. Even if the data transmitted is protected, it is still unwise to let attackers eavesdrop on our transmission. There should be monitoring involved to ensure there are no eavesdroppers at all times.

Lastly, there should be no tampering with the physical machine itself. If we protect the digital data but not the physical machine, the SPAM2 system is still vulnerable to non-digital attacks.

# 5. Conclusion

With these countermeasures in place, it adds many required protections in order to ensure the smooth operation of the SPAM2 system. The menu-of-the-day information has been integrity protected, the day-closing information has been protected with confidentiality and non-repudiation protection in transit and has confidentiality protection at rest.

# 6. Miscellaneous

## 6.1 Planned Task Allocation

Risk Assessment - Yi Terng

Proposal - Yi Terng and Sean

Code - Sean and Yi Terng

## 6.2 Reflection

Sean:

This assignment is challenging as it requires us to be able to implement and apply the different encryption techniques that were taught in the lectures. It also includes a wide variety of topics; such as digital signatures, encryption with different algorithms and hashing. It also posed a great opportunity to read and talk a look back at the cryptography notes. It was very interesting to search and read up more about the different techniques that could be used and how safe each of them were.

Yi terng:

This assignment was very fun and exciting as we had the chance to apply what we learnt in lectures in a practical and real-life setting. It was challenging as application required research and good understanding of the topic at hand. This was a real eye-opener as learning about the topic is one thing, but learning how to apply it is another. It was thrilling to learn the different encryption and hashing algorithms in order to protect information and it makes you relate and think about how it applies to real-world implementation today, in the every-day applications that we use, such as Twitter

*End of Report*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Risk** | **Likelihood** (eg High, Medium, Low) | **Severity**  (eg High, Medium, Low) | **Response Strategy** (eg Avoid, Transfer, Mitigate or Accept) | **Actions required** | **Due Date** |
| Man in the middle attack. If an attacker were to intercept the menu and alter it before sending to the client from the server, it would compromise the integrity of the menu and jeopardize operations | High | High | Mitigate | Use Hashing to guarantee message integrity. | Feb |
| Ensure non-repudiation of server acknowledgement to prove that the server received the valid day-closing information from the client | Low | Medium | Mitigate | Upon receipt of valid day-closing Infomation from the client, the server sends an acknowledgement to the client signed with its digital signature to prove that it received the data. | Feb |
| The file was stolen. The data stored in the day-closing file needs to be encrypted to prevent deciphering should the file be stolen, this would compromise the confidentiality of the contents. | Medium | Medium | Mitigate | The data that is to be stored in the day-closing file will be encrypted using RSA before being saved into the file. This ensures that even if the file is stolen, the attacker will still have to break through the cipher. | Feb |
| An attacker could compromise the non-repudiation of the day-closing information in transit by sending his own copy of false day-closing information to the server | Medium | High | Mitigate | To combat non-repudiation. The contents of the file can be signed with an arbitrated digital signature. Therefore, it can be called trusted as there is a trusted third party involved. | Feb |
| Eavesdropping. An attacker could intercept the day-closing information and read its content and compromise its confidentiality | High | High | Mitigate | Before sending the day-closing information through the connection, encrypt it first using AES and decrypt it only after received by the server. | Feb |
| Login information of the users such as their username and passwords are stored in plaintext in text files. This is very dangerous as if the files were to be stolen, it would be easy for the attacker to find out credentials of the users | High | High | Mitigate | Encrypt all user login data using RSA or AES before storing it | Feb |

References:

Vandyke.com. (2020). *Data Integrity Information File Transfer-Data Encryption Security*. [online] Available at: https://www.vandyke.com/products/securefx/data\_integrity.html [Accessed 5 Feb. 2020].

SearchSecurity. (2020). *What is Advanced Encryption Standard (AES)? - Definition from WhatIs.com*. [online] Available at: https://searchsecurity.techtarget.com/definition/Advanced-Encryption-Standard [Accessed 6 Feb. 2020].

Vandyke.com. (2020). *Data Integrity Information File Transfer-Data Encryption Security*. [online] Available at: https://www.vandyke.com/products/securefx/data\_integrity.html [Accessed 5 Feb. 2020].