Phang Teng Fone 1003296
Gerald Lim 1003371

Programming Assignment 2 Report

Fig. 1 below gives the basis of a possible protocol. However, there's one problem with the. What is the problem? Explain it in your handout for submission, and give a fix for the problem.

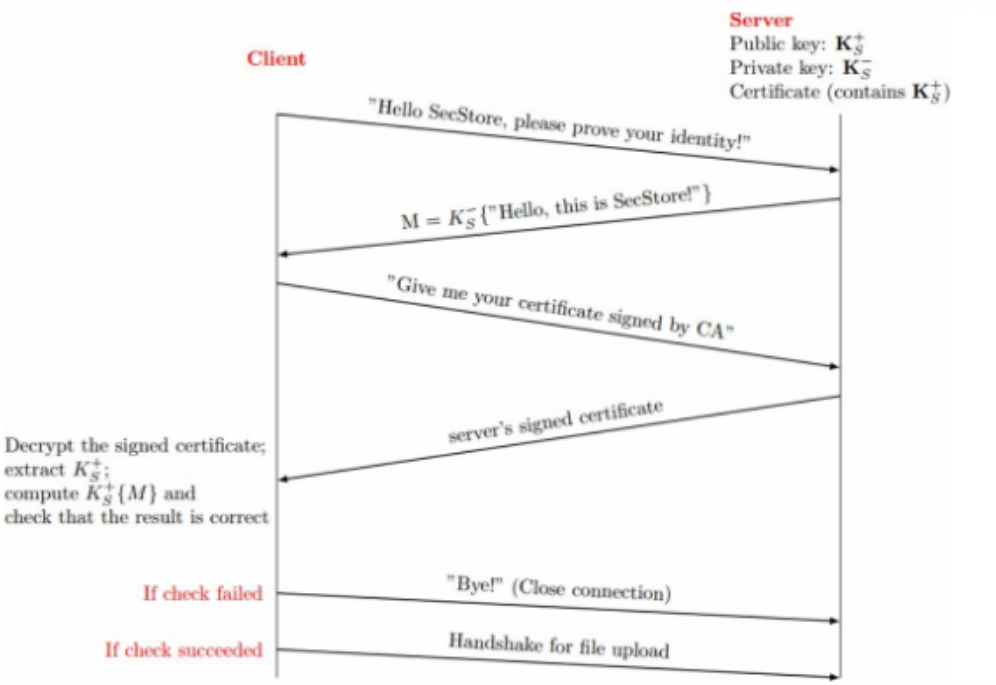
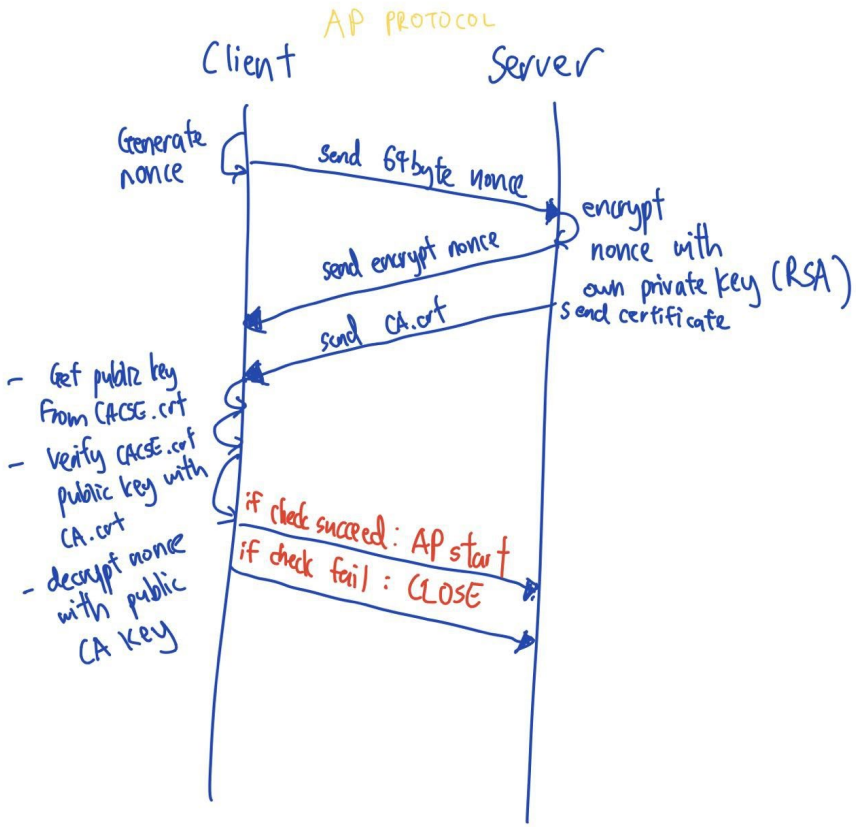


Fig. 1: Basis of Authentication Protocol

The issue with the handout is that there is no way the client can tell that the public key belongs to the server. This is susceptible to the man-in-the-middle attack. Where T(malicious attack) may impose as client to server and impose as server to client. In this case, it is more inclined to client to server impersonation.

T may hijack the conversation between the client and the server in the following manner:

- T hijacks the client's nonce and impersonates as the server
- T encrypts the client's nonce with his own private key (M') and sends it back to the client
- Client requests for a certificate signed by CA and once again got intercepted by T, T then sends his own public key to the Client
- Client decrypts M' with the public key received and compares the decrypted result is the same as the client's original nonce
- Since the results match, client is authenticated with T, now T can receive all of the files sent by the client
- **Confidentiality is breached**

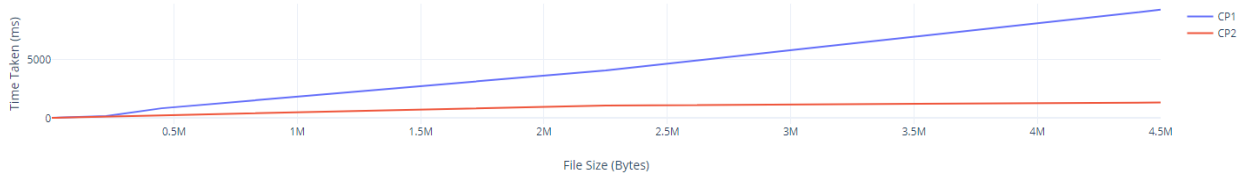


Proposed Solution

- The client first sends a 64 bytes nonce to the server
- The server the encrypts the client's nonce its own private key using RSA encryption and sends it back to the client
- The server also sends its signed certificate along with the encrypted nonce
- The user retrieves the CA public key from the certificate provided by a known organisation
- The user validates the server signed certificate and verifies the CA public key with the server signed certificate
- If the above two conditions pass, authentication is a success and client can begin sending encrypted files, otherwise server may be a malicious attacker

The proposed solution ensures integrity and confidentiality as malicious attacks such as man-in-the-middle attack can be prevented. This can be observed in 2 scenarios:

- Upon comparing the decrypted nonce with the client's original nonce, the client is able to determine whether a malicious attacker has tampered with the nonce while it's being sent the server
- Upon verifying the CA public key with the server certificate, the client is able to determine whether a malicious attacker has an inauthentic certificate



Graph of File Size (Bytes) vs Time Taken (ms)
Blue Line - CP 1
Red Line - CP 2

CP1

File Name	File Size	Time Taken
100.txt	4,500 bytes	26.982311ms
200.txt	9,000 bytes	29.992244ms
500.txt	22,500 bytes	57.022084ms
1000.txt	45,000 bytes	71.034611ms
5000.txt	225,000 bytes	168.586852ms
10000.txt	450,000 bytes	838.779694ms
50000.txt	2,250,000 bytes	4061.229645ms
100000.txt	4,500,000 bytes	9216.593608ms

CP2

File Name	File Size	Time Taken
100.txt	4,500 bytes	25ms
200.txt	9,000 bytes	16ms
500.txt	22,500 bytes	31ms
1000.txt	45,000 bytes	43ms
5000.txt	225,000 bytes	137ms
10000.txt	450,000 bytes	234ms
50000.txt	2,250,000 bytes	1076ms
100000.txt	4,500,000 bytes	1324ms