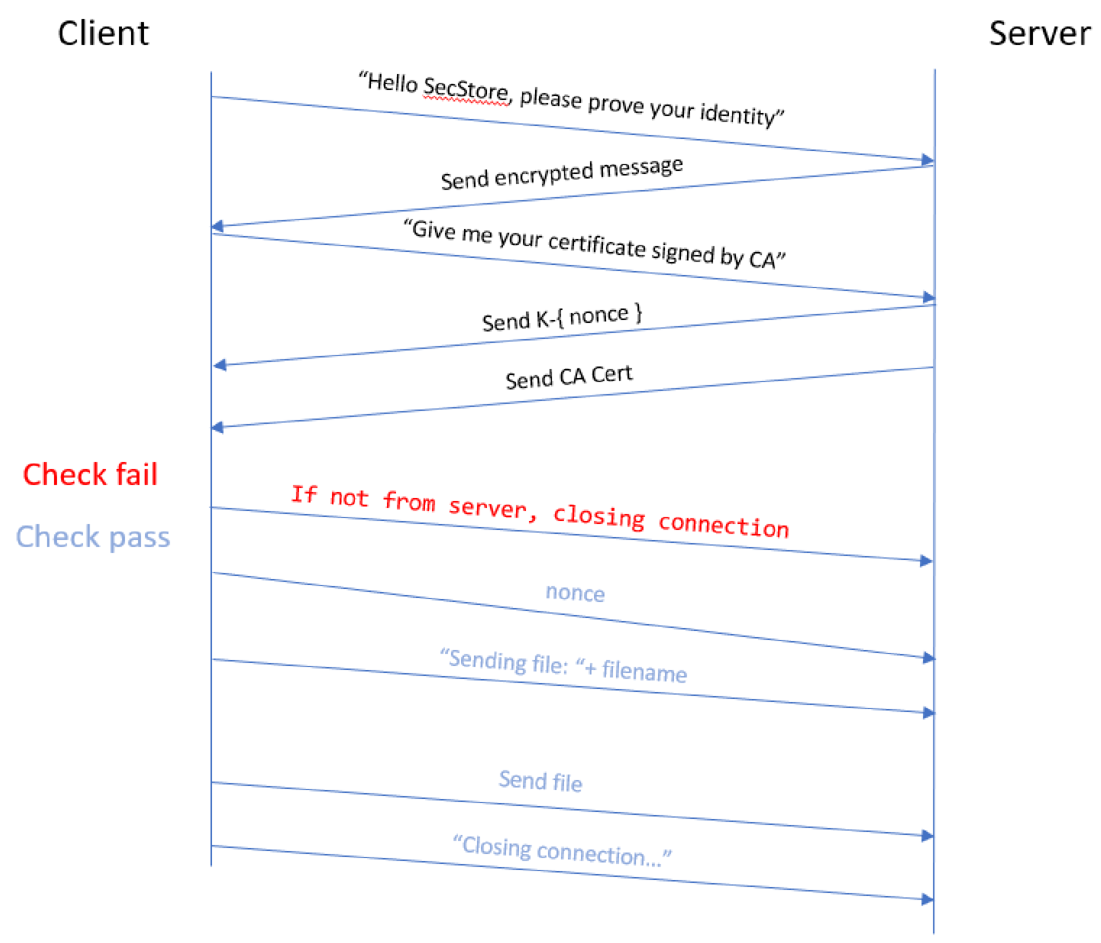
Programming Assignment 2

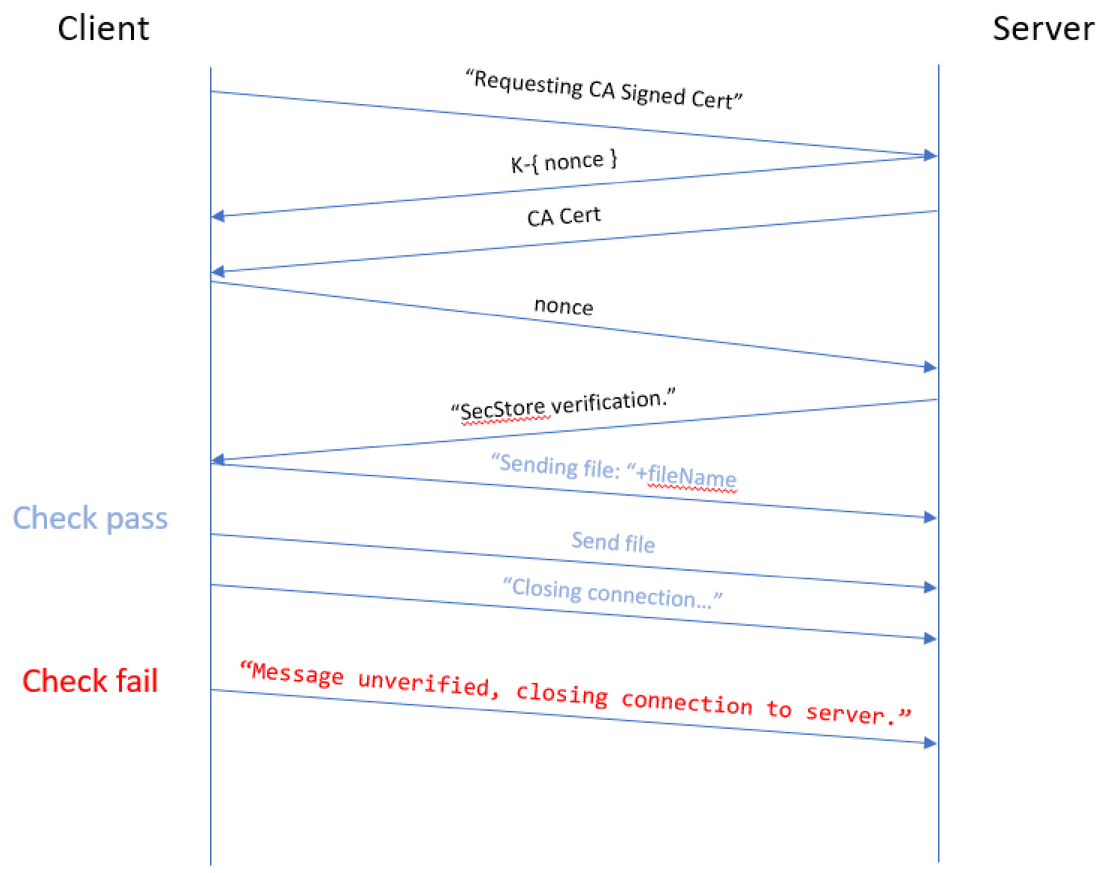
Assignee	
Created	@Jan 20, 2021 11:10 PM
Created By	Ⓞ Oei Kai Xun
Done	☐
Due	@Apr 21, 2021 11:59 PM
Edited	@Apr 22, 2021 1:51 AM
Inbox	✓
Kanban - State	To Do
Kanban - Tag	
Next Due	
Parent Task	
Priority	🧀 Medium
Project	💻 50.005 Computer Systems Engineering 2021
Recur Interval (Days)	
Start	
State	🔴
Sub-Tasks	
Type	⏳One-Time

Specification

CP1


Client
Server
"Hello SecStore, please prove your identity"
Send encrypted message
"Give me your certificate signed by CA"
Send K-{ nonce }
Send CA Cert
Check fail
Check pass
If not from server, closing connection
nonce
"Sending file: "+ filename
Send file
"Closing connection..."

CP2


Client
Server
"Requesting CA Signed Cert"
K-{ nonce }
CA Cert
nonce
"SecStore verification."
"Sending file: "+fileName
Check pass
Send file
"Closing connection..."
Check fail
"Message unverified, closing connection to server."

Problem

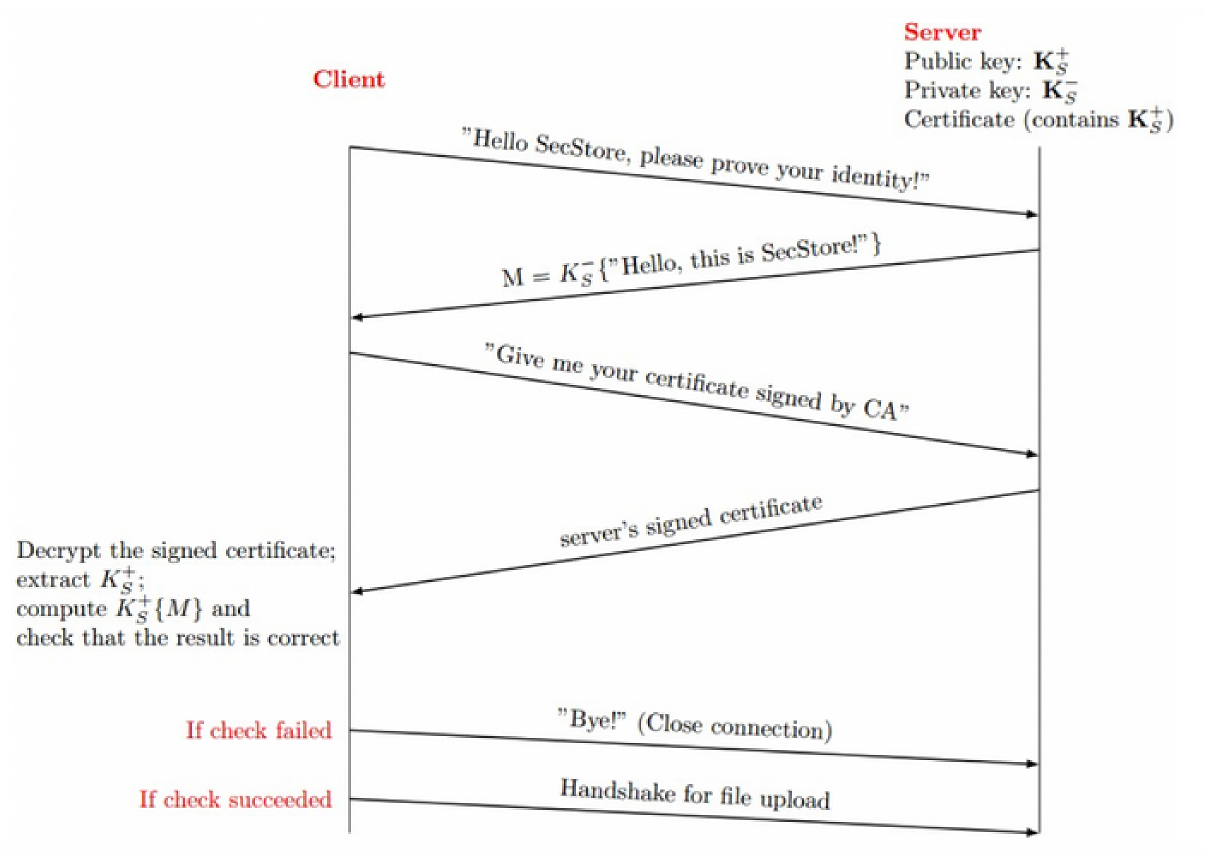

Server
Public key: $\mathbf{K}_S^+$
Private key: $\mathbf{K}_S^-$
Certificate (contains $\mathbf{K}_S^+$)
Client
"Hello SecStore, please prove your identity!"
$M = K_S^-\{$"Hello, this is SecStore!"$\}$
"Give me your certificate signed by CA"
server's signed certificate
Decrypt the signed certificate; extract K_S^+; compute $K_S^+\{M\}$ and check that the result is correct
If check failed
"Bye!" (Close connection)
If check succeeded
Handshake for file upload

Fig. 1: Basis of Authentication Protocol

Problem

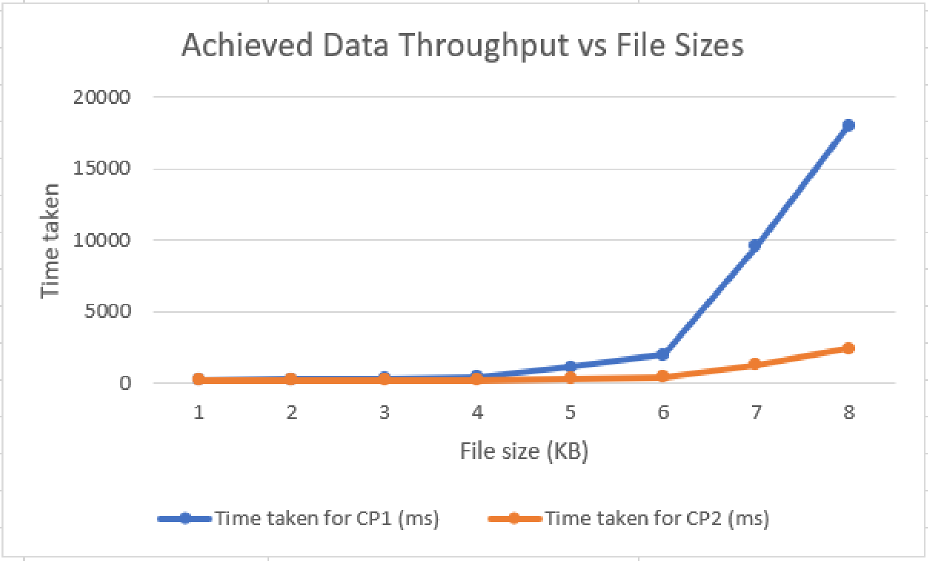
1. The client cannot tell if the public key belongs to the server or not.
2. M is not encryption of the nonce of "Hello SecStore, please prove your identity!" sent by the client.
3. The nonce should be randomly generated for one-time use to prevent malicious users to impersonate the server.

Solution

1. By using a CA-signed public key, we can verify that the public key indeed belongs to the server. This is similar to Solution 2 in case 5 of the handout.
2. When client decrypts M and it will lead to a fail check as the plaintext is not the same as the original nonce. So, the encrypted message sent by the server must be the identical nonce that client initially sends.
3. To generate random nonce, it can be done with Random().

Plots

File name	File size (KB)	Time taken for CP1 (ms)	Time taken for CP2 (ms)
100	5	226.0442	205.5751
200	9	258.3352	211.3772
500	23	308.6892	214.5001
1000	45	421.5876	225.3045
5000	225	1146.7001	321.4253
10000	450	1988.3939	430.6536
50000	2247	9584.5948	1290.4847
100000	4493	18038.3683	2417.9197


Achieved Data Throughput vs File Sizes
Time taken
File size (KB)
Time taken for CP1 (ms)
Time taken for CP2 (ms)