Network Verification Based on TLS Protocal

Yiwen Tang Jiayi Zhang Yawei Zhang Rui Zhang

Carnegie Mellon University

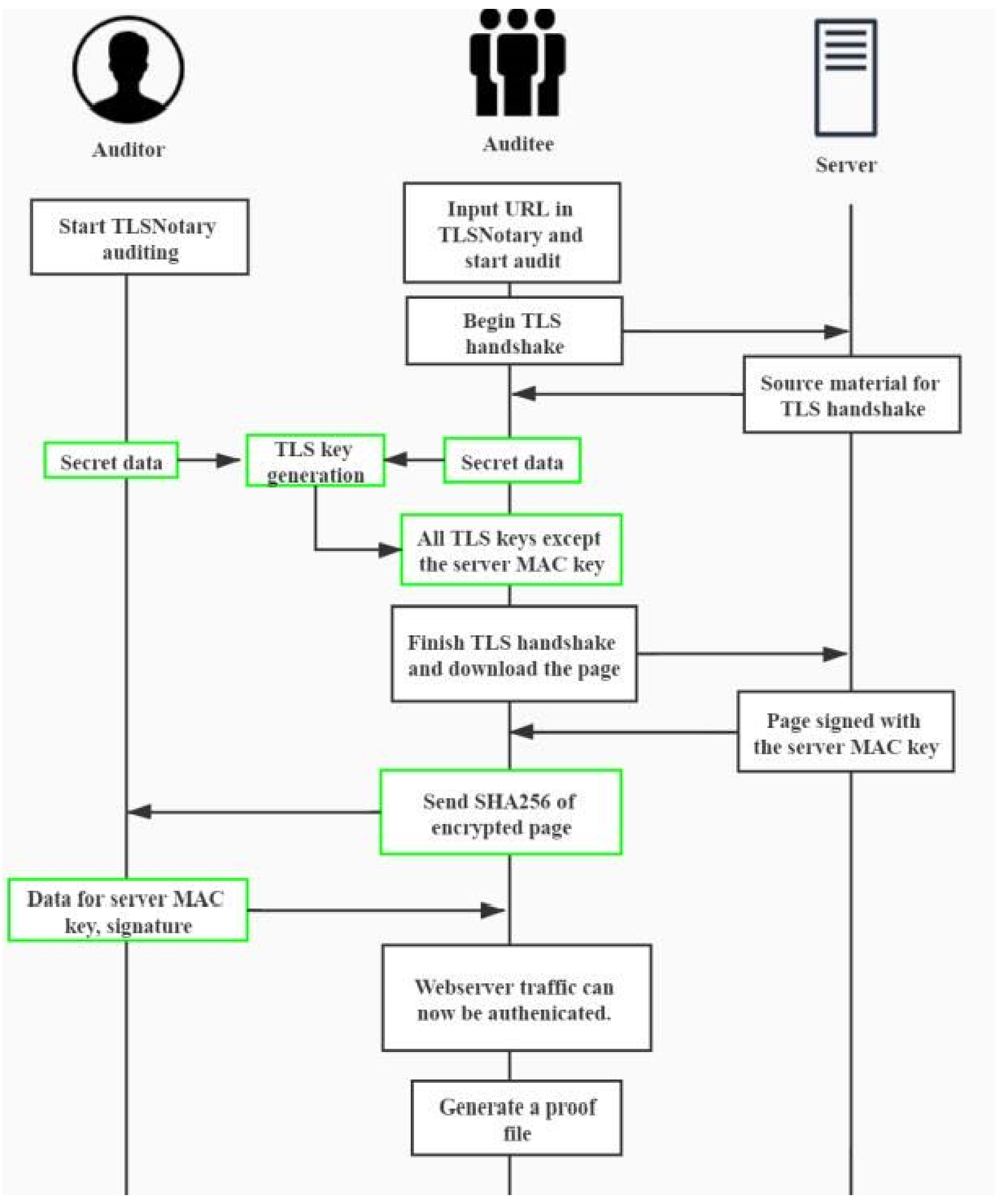
Introduction

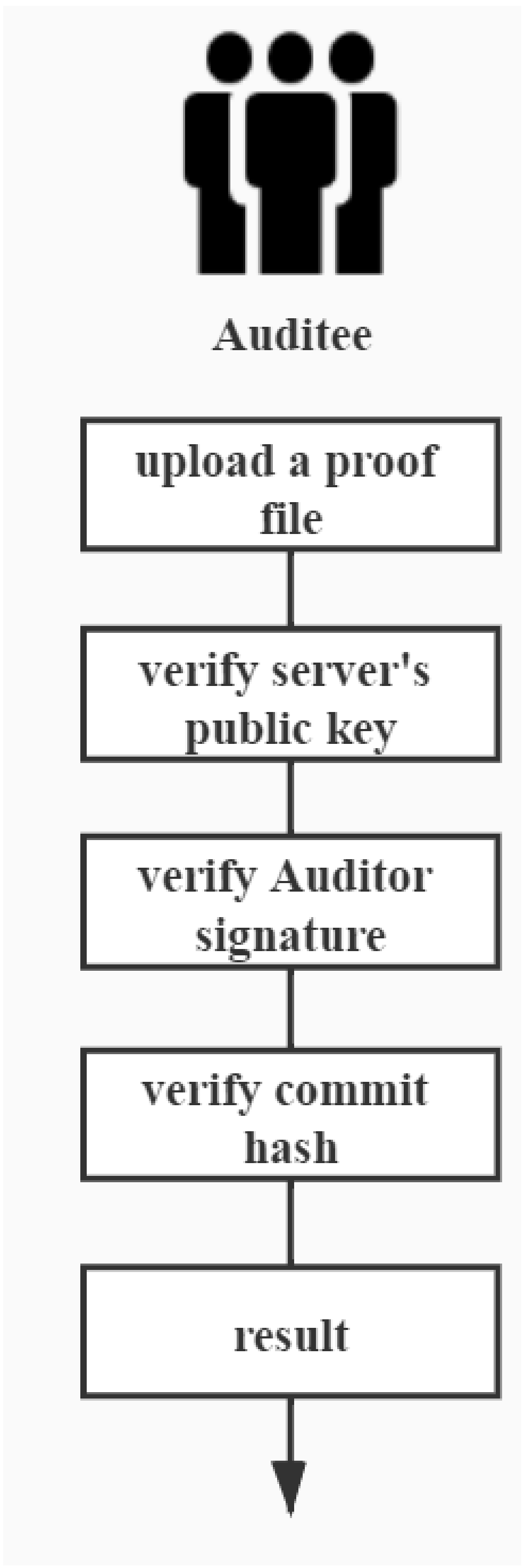
Although TLS protocol is used to encrypt network, there still exists attacks aiming at SSL/TLS. We focus on implementing an additional protection above TLS to retrieve secure data from network. Our project allows a client to provide evidence to a third-party auditor that certain web traffic occurred between him and server.

Mechanism

We separate client site into two parts: Auditee and Auditor.

- Auditee performs data acquisition.
- Auditor is responsible for data verification.

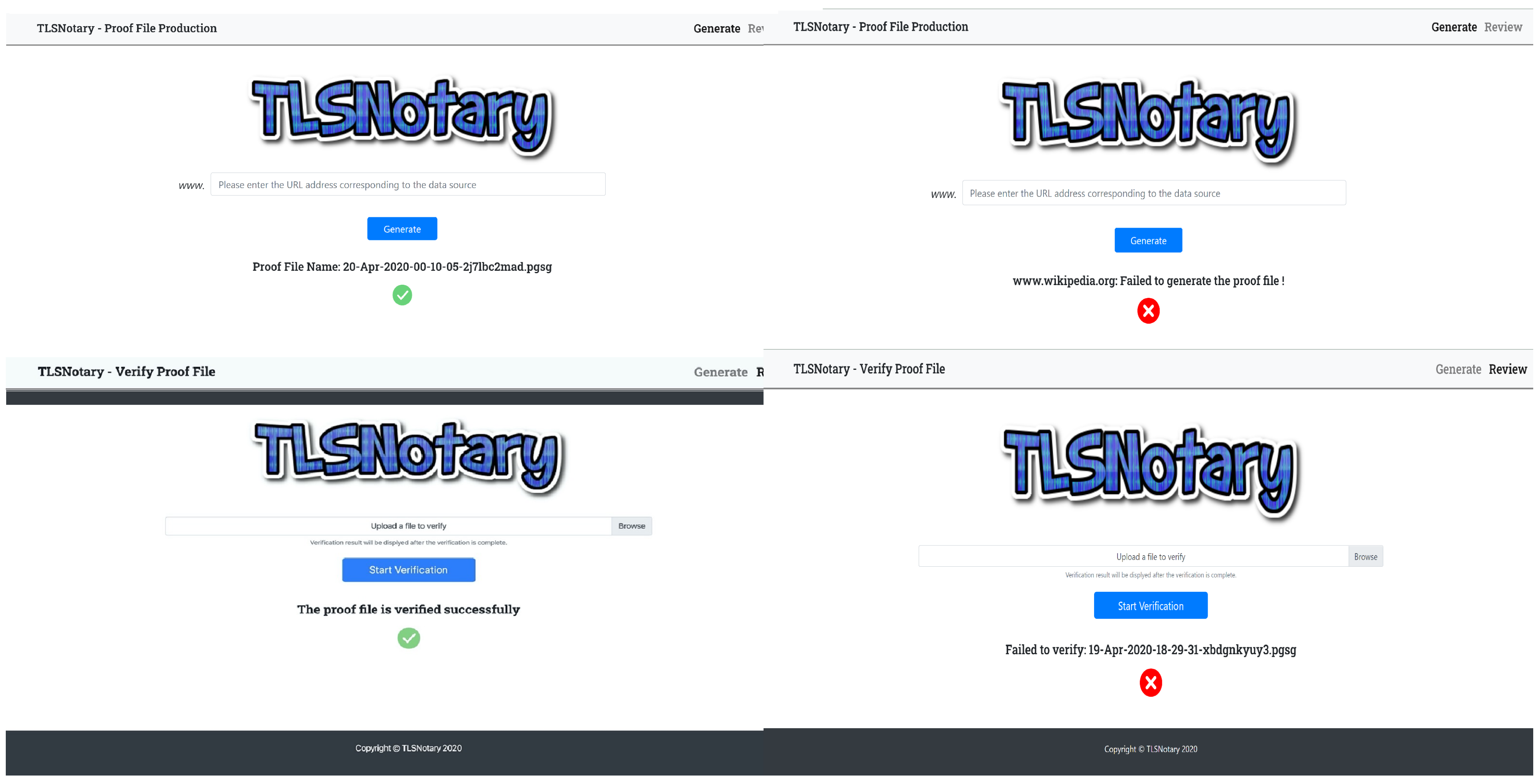

Auditor
Auditee
Server
Start TLSNotary auditing
Input URL in TLSNotary and start audit
Begin TLS handshake
Source material for TLS handshake
Secret data
TLS key generation
Secret data
All TLS keys except the server MAC key
Finish TLS handshake and download the page
Page signed with the server MAC key
Send SHA256 of encrypted page
Data for server MAC key, signature
Webserver traffic can now be authenicated.
Generate a proof file



Implementation

Based on algorithms and mechanisms above, we:

- Implement auditor and auditee using Python.
- Design a website as user interface using Flask.
- Deploy the whole system is on AWS.


TLSNotary - Proof File Production

Please enter the URL address corresponding to the data source

Proof File Name: 20-Apr-2020-00-10-05-2j7lbc2mad.pgsg

TLSNotary - Proof File Production

Please enter the URL address corresponding to the data source

www.wikipedia.org: Failed to generate the proof file !

TLSNotary - Verify Proof File

Verification result will be displyed after the verification is complete.

The proof file is verified successfully

TLSNotary - Verify Proof File

Verification result will be displyed after the verification is complete.

Failed to verify: 19-Apr-2020-18-29-31-xbdgnkyuy3.pgsg

Copyright © TLSNotary 2020

Copyright © TLSNotary 2020

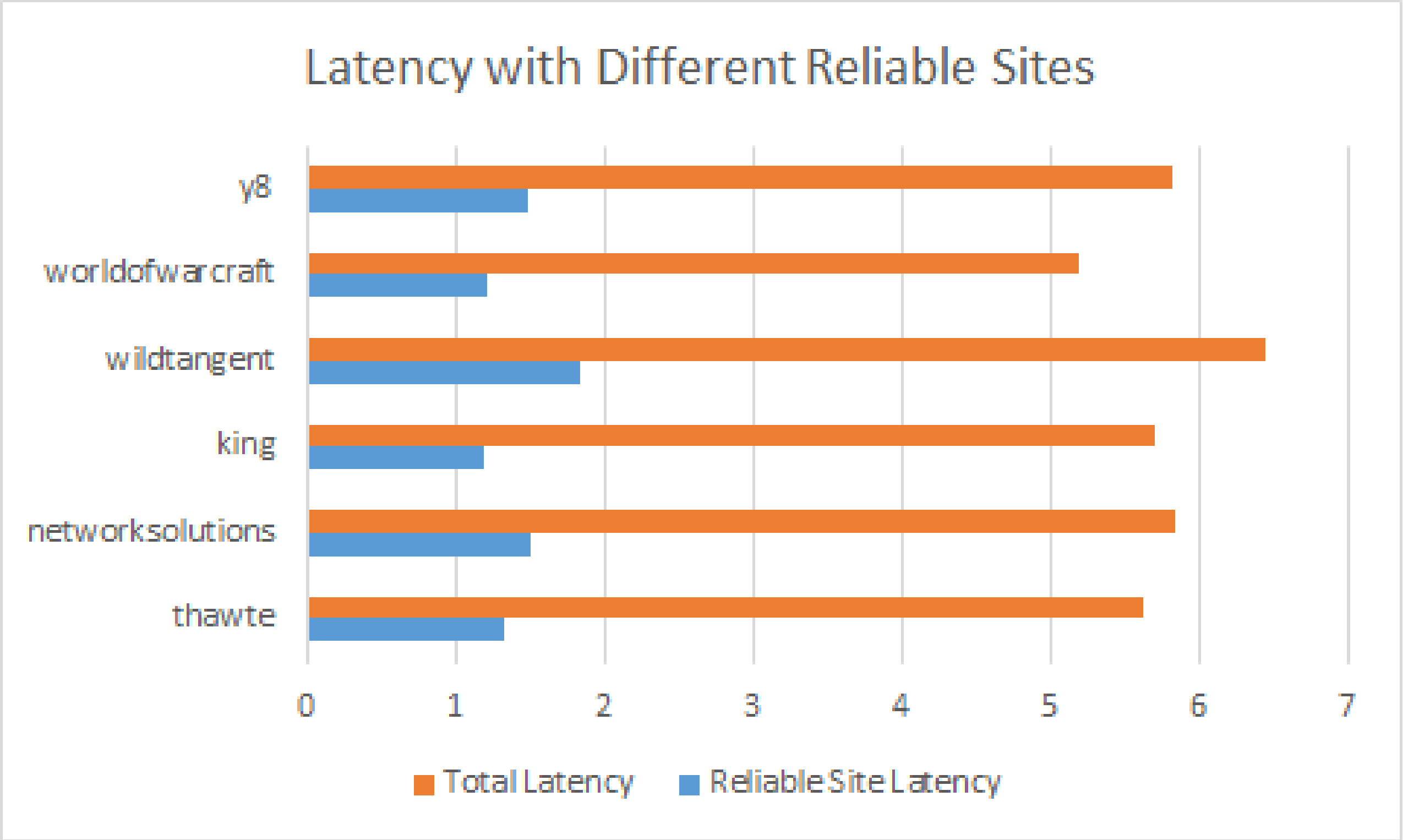
Experiment

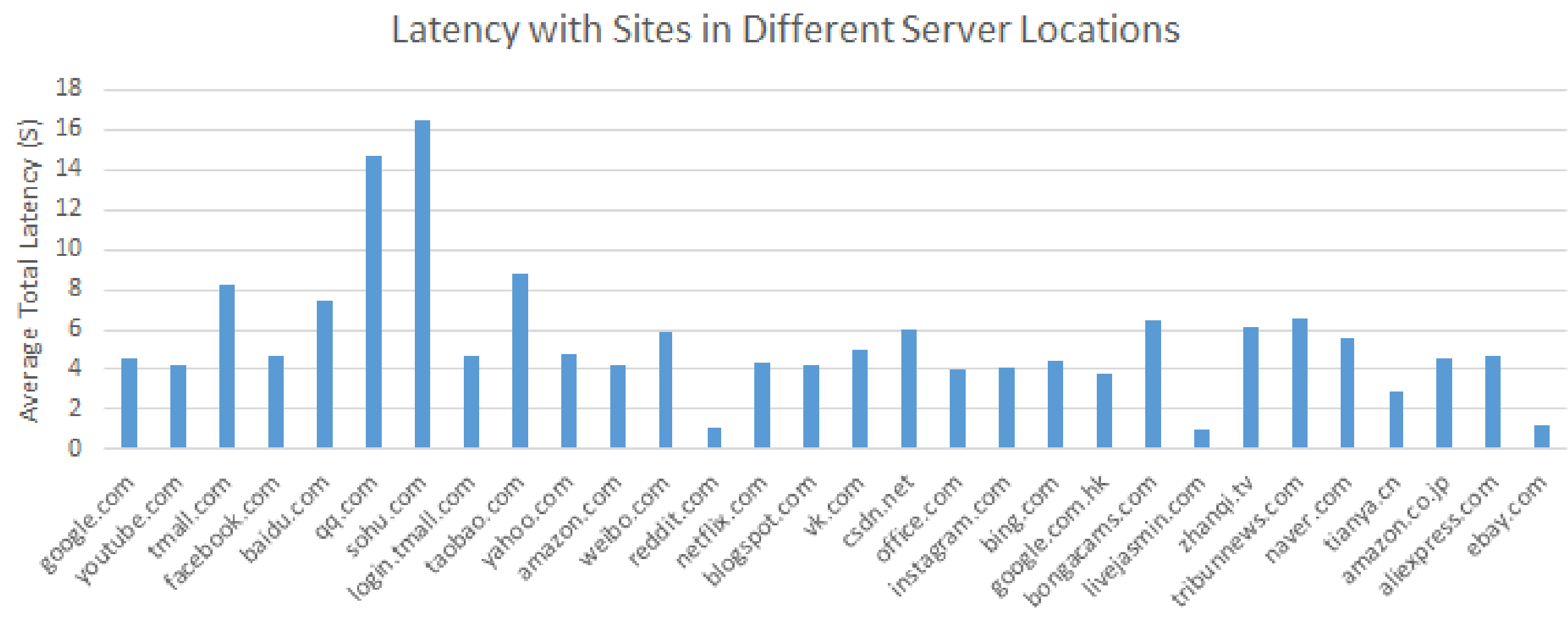
1. Functional Experiment

We repeat the URL requests for 500 websites twice and 614 of the 1000 experiments success.

2. Latency Experiment:

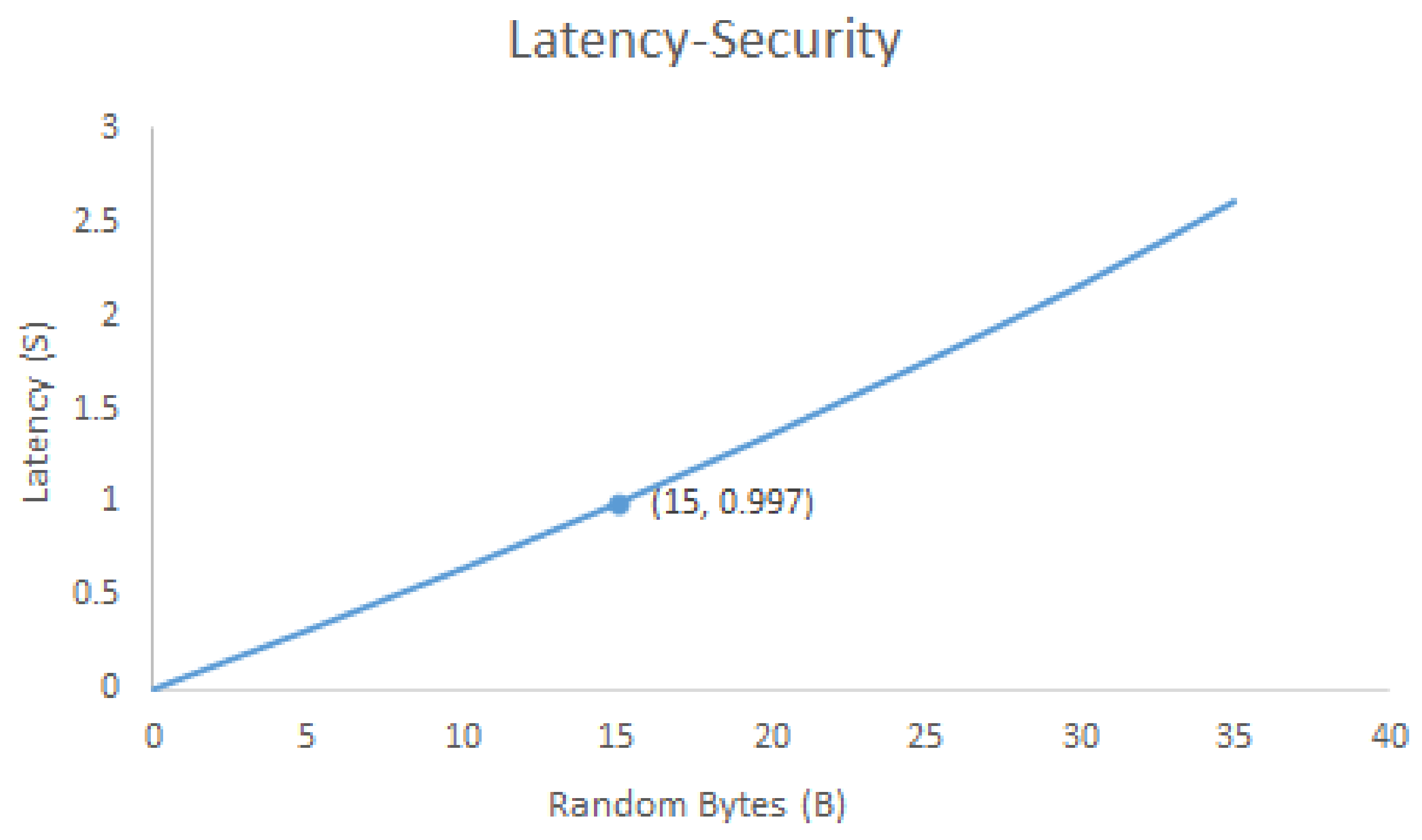
We conduct two experiments on the latency with different reliable sites and with sites in different server locations.





3. Security Experiment:

There is a trade-off between latency and security. In order to ensure that the latency is within an acceptable range, and the number of random bytes reaches the maximum, we choose 15 as our system's random bytes.



Conclusions

At the end of semester, we not only have a deeper understanding of the OSI network model, but also understand the TLS protocol deeply. When implementing the verification system, we get to know the process of the three handshakes of TLS and encryption, and deepen the understanding of RSA.