Title

MSc Internship

Cyber Security

Onkar Kulkarni

Student ID: X18103693

School of Computing

National College of Ireland

Supervisor: Ross Spelman

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Mr. Onkar Kulkarni
Student ID:	X18103693
Programme:	MSc Cyber Security **Year:** 2018-19
Module:	Internship
Supervisor:	Ross Spelman
Submission Due Date:	12th August, 2019
Project Title:	Prevention of Man-in-the-Middle Attack on Internet Communication Using Blockchain Technology
Word Count:	4722 **Page Count:** 17

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: .....

Date: ……

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Preventing the Man-in-the-Middle Attack on Internet

Communication using Blockchain Technology

Onkar Kulkarni

X18103693

Abstract

Internet communication is getting massive popularity these days. Internet communication and computer technology have become an integral part of everyone's lives. People are exploiting the online ways to chat with each other, find ways of entertainment, put forward their views before the world, shop for themselves, pay the bills, etc. owing to which a large amount of data is generated and transferred over the internet every day. Thus, it becomes necessary to protect the data which is transmitted on the internet. With the advent of HTTPS, the internet communication is believed to have been secured in transferring the data from one computer to another. But, lately some of the issues with HTTPS security have been reported, because of which tampering of data in transit and attacks like MITM attacks are on the rise. One such issue is ability to compromise the single trusted certificate that can represent any website. Thus, this paper thus considers the mentioned issue and tries to prevent the MITM attacks on HTTPS by using notary systems in HTTPS. While doing so, the paper tries to put light on the working of HTTPS and explains how Man-in-the-Middle attack is carried out on HTTPS. The solution is developed leveraging blockchain technologies. The paper also evaluates the system and lists some of the shortcomings of the model and suggest the future work.

1. Introduction

Main motive behind the development of HTTPS is to secure the communication of data flowing from one computer to another through HTTP protocol. It makes use of Secure Socket Layer (SSL) and Transport Layer Security (TLS) protocol to implement the concept of HTTPS. The primary function of SSL/TLS is to encrypt the communication taking place between two computers. TLS/SSL facilitates not just the encryption in the communication but is also responsible for validating the users so that the data flows to the right destination and the integrity of the data is maintained. Following section 1.1 explains in detail about the HTTPS and its working.

1.1 HTTPS

HTTPS technology was developed by Netscape Communications in 1994 for its own web browser Netscape Navigator. Initially, Secure Socket Layer (SSL) protocol was used as an

underlying technology for developing HTTPS. SSL later to Transport Layer Protocol (TLS), due to which it was recognized by RFC 2818 in May 2000 – Wikipedia.org (2019).

Following sub sections explain the working of TLS model, its handshake process, puts some light on certificate model and explains what the vulnerabilities are associated with HTTPS.

1.1.1 TLS

TLS protocol is a cryptographic algorithm which provides complete security to the data transmitted over the network. However, it should be noted that, TLS is not responsible for securing the data in the end systems. It is mostly used for browsing web securely. The website which deploys the TLS communication, would display the green padlock icon in the web browser to identify themselves as secured with HTTPS. TLS combines both asymmetric and symmetric cryptography to avail both performance and security features while sending the data securely over internet. The keys required for cryptography are generated using algorithms like RSA, Ephemeral Diffie-Hellman (DHE), Diffie-Hellman (DH) and also uses Ephemeral Elliptic Curve Diffie-Hellman (ECDHE) and Elliptic Curve Diffie-Hellman (ECDH). TLS not just protects the integrity of the data, but also it authenticates the client who desires to connect to the server by validating the server's public key - internetsociety.org (2019). This validation process is explained in the following sub section.

1.1.2 Handshake

This subsection provides the information about the TLS handshake in brief.

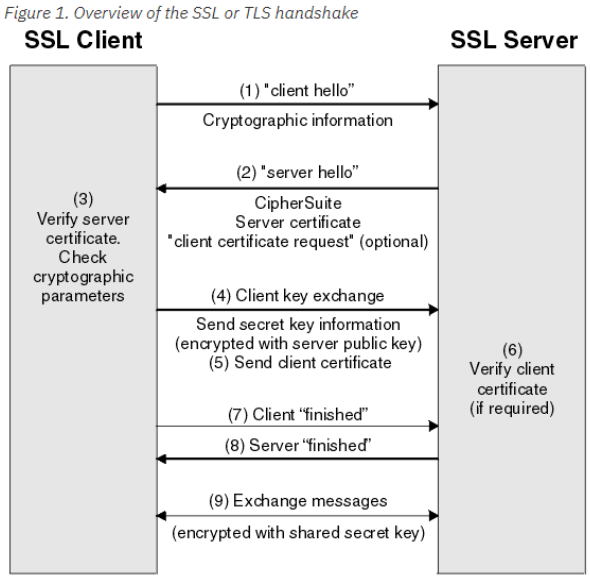

Figure 1: TLS Handshake

The figure shown above shows the process of TLS handshake. As seen from the above figure:

- The TLS Client transmit a “client hello” message which contains cryptographic information viz TLS version and specifies the CipherSuites which the client supports in a preferred order. The message also provides a byte string which is required for successive computations. The protocol also includes data compression functions which are included by the client
- In response to the message transmitted by the client, the TLS Server “server hello” message that comprises of the CipherSuite which it selects from the list sent by the client, some other random byte string and Session ID. The message also includes server certificate. Depending on whether the digital certificate is authenticated with the clients certificate, it sends out the “client certificate request” which consists of a list specifiying which certificates are accepted and names of valid Certificate Authorities (CA).
- The TLS client scrutinizes the digital certificate using asymmetric and symmetric cryptography.
- The TLS client shares the random byte string to the server so as to compute the secret key that encrypts the messages sent between client and server. Public key of the server is used to encrypt the random byte string.
- If the server initiates the “client certificate request”, the client encrypts encrypts the random byte string with its private key and sends it across to the server along with its digital certificate or a “no digital certificate alert”. The alert is nothing but just a warning, but if client authentication is mandatory the handshake might fail as well.
- The certificate is verified by the server using symmetric and asymmetric cryptography.
- The “finished” message is sent out by both client and server as soon as their part in the handshake process gets over.
- They can now exchange messages using the shared secret key – ibm.com (2019).

1.1.3 Certificates

Certificates used in TLS protocol verify the identity of the client. They give assurance of the establishment of a trusted and secured connection. SSL Certificates consists of public and private key pair. These keys are used together to create an encrypted connection. The certificate also incorporates “subject” which can identify a legitimate owner.

Whenever a browser tries to access website which is TLS protected, the TLS handshake is initiated to establish connection between web server and browser. The TLS

handshake is although not visible to the user, it is the process that executes in the background. In this process three keys are used viz session keys, private keys and public keys -digicert.com (2019).

1.2 Problem Definition

All though the TLS systems provides security to the data in transit, the trustworthiness of these systems is heavily dependendent on Certificate Authority (CA) systems. There have been many incidences in the past wherein it became possible to perform the Man-in-the-Middle Attack. These attacks have been reported from across the globe. This is because of the weak certificates which are easy to forge and make the server believe that those certificates are valid ones and thus establish the connection. The Man-in-the-Middle attack on HTTPS is on the rise.

Thus this paper defines the problem statement as follows

Internet Communication is vulnerable to Man-in-the-Middle attack.

1.3 Solution

Thus in this paper, we take into account the issue of compromising TLS using Man-in-the-Middle attack and present a system called Persistent and Accountable Domain Validation Extended (PADVAE) to prevent the issue of Man-in-the-Middle Attack on HTTPS. The system is based on previous research conducted by Szalachowski, P. (2018). Szalachowski, P.(2018) attempts to improve the existing notary based systems. Notary based systems were introduced in HTTPS earlier, but it had been discovered that there are some issues with the notary based systems like not having enough accountability and the system also introduces privacy issues in the working of HTTPS security.Thus, to implement a more secured and more accountable notary systems, Szalachowski, P. (2018) brings in blockchain technologies into recognition. The Ethereum technology was used in the implementation. We propose to use Nebulas blockchain technology. The nebulas blockchain was developed in 2018 and is the advanced version of ethereum blockchain. The nebulas blockchain works on some of the drawbacks of ethereum blockchain and like issues with Smart Contracts and PoW which are the important parts of blockchain development and its application in the HTTPS security.

Thus the above introduction gives an overview of how the TLS Technology works and discusses in brief about the possible drawback of the TLS technology. The below section “Related Works” attempts to put more light on the problem statement and some of the solutions in the recent times which were developed against the issue.

2. Related Works

The issue with the TLS Security has been debated over a long time. Since a long time there were discussions regarding the root cause of the issue and possible solutions for mitigating this issue. This section puts light on some of those solutions by referring to the work of researchers and experts.

A research was conducted in year 2013 by Clark, J (2013) to understand the problem in the TLS technology. Clark, J. (2013) conducted a research on HTTPS security and found out some issues in the TLS protocol. Some of which are listed below:

- Weak Encryption, Signature Key Lengths and Hash Functions

  Many of the encryption algorithms found in ciphersuites of initial versions of TLS are not secured enough to protect the modern day attacks. Any symmetric keys having encryption schemes of 40, 64 or 56 bit lengths can be broken using brute force attacks. Weak hash functions can be responsible for collision attacks.

- Flaws in the Implementation and Attacks Related to it

  The author in this section explains how the misconfigurations and implementation flaws may lead to some of the serious attacks. He states that, some of the values in the TLS protocol are generated randomly. A capable Pseudo Random Number Generator (PRNG) is the basic requirement for generating secured and unpredictable random values. The writer claims that the Netscape Browser during the initial versions, was dependent on PRNG algorithm with weak keys which made the TLS Session Keys predictable. The author mentions about another important attack called as remote timing attacks. This attack is being performed against optimzed type of RSA decryption. The decryption makes use of the secret keys, which are to be used for long term to take branching decisions. Owing to this, the difference in execution time increases, TLS Handshakes leaking data about the keys.

- Protocol Level Attacks

  The author explains about the ciphertext and version downgrade attacks. The ciphersuites are used during the client and server negotiation process. It was possible to downgrade the ciphersuite strength to weakest possible for both the parties. This was possible during the SSL 2.0. But in the advanced versions of SSL this attack was fixed stated author Clark, J. (2013). The version downgrade is also influenced by the man-in-the-middle attack, which attempts to initially downgrade to SSL 2.0 and then to the ciphersuites. Although TLS implementations works to prevent this attack, the author claims that there may have been some vulnerabilities in the client which makes the attack still possible.

The author further explains in detail some of the issues with the TLS due to which the attacks may have been possible.

Certification

Author says that validation service of HTTPS can be automated by initiating the emails to an email id related to the top level domain of Common Name (CN) or an entry from CN's WHOIS records. Success of both the methodologies depends on DNS records. If the DNS records are altered, both the mechanisms can fail asserts Clark, J. (2013). The author states, that a few issues of automated issues are taken care off by the EV certificates. However, downgrading attack can be performed on the EV certificates by using man-in-the-middle attack and replace it with DV certificates.

Thus the author claims weakest of the CA certificates could be the target of the man-in-the-middle attack and brings to the readers notice, the two CAs namely Comodo and Digitar Notar were compromised in the year 2011 Clark, J. (2013).

There have been consistent efforts made to address the issue. Thus to address this issue, author Conti, M. (2016) takes the above research forward and focusses on man-in-the-middle attack on HTTPS and takes a note of the solutions which were implemented to resolve this issue. The author broadly classifies the solutions in the following types:

- Determination of Forged Certificate

  This is one of the solutions employed to determine the forged certificate. In this solution audit logs of HTTPS certificates are created and are verified and monitored independently. The author says that this technique introduces a ICSI Notary Service and collects the certificates real time from the internet sites and logs it into the centralised database. It employs the concept of Crossbear or Observatory techniques states the author. These concepts leverage the third party audit systems but provide a more detailed analysis states Conti, M. (2016).

- Certificate Pinning

  The authors state that these solutions compare the certificates provided by the client with the certificate of the server to check on the MITM attack. In this concept, the servers publish their certificates and public keys, whose reference would be needed in the future for TLS handshakes. The users then scrutinize if there are any changes made in the certificates. The author mentions some of the ways of implementing this solution. Server based methods which focusses on storing the pins on the server. DNS based where the public key is pinned in their DNSSEC record.

- Multipath probing solution

  This method the author explains also introduces the notary based systems which uses a voting based approach to MITM attack. Depending on those votes whether to accept or reject the certificate is decided. This solution as per the author was developed as a firefox plugin.

- Forcing the SSL/TLS connections

  This method uses the ISAN-HTTPS Enforcer that leverages the Javascript API to enforce the redirection to HTTPS.

- Friendly MITM

  Friendly MITM is a method as described by the author, is developed to take on the MITM attack on mobile device. This method tries to behave like a MITM and assess the certificate everytime the new request is made by the client to connect. After the scrutiny the server takes a call on connecting with the user as claimed by Conti, M. (2016).

2.1 Notary Based Systems

A research was conducted in 2012 in analyzing Notary based systems by author Holz, R. (2012) and assess the existing called as Crossbear. It is ready for usage, and around 150 hunters have already been implemented by Testbed as stated by Holz, R. (2012). Crossbear as explained by the author employs umpteen "hunters" on the internet. To test the systems the author implements two kinds of hunters. One is a Firefox add-on which is employed for both detection and localization and other one only performs the localization. The Crossbear stores a list of MITM affected servers. The hunters always refer to this list at regular intervals of time. They then connect with the server that were reportedly attacked by the MITM and pull out the certificates that the server sends and determines the IP route leading to the server leveraging traceroute and stores all this information in the centralised database. Apart from the information gathered using the hunters, the crossbear also gets additional information in geo-IP database, also pull out WHOIS information states Holz, R. (2012). The author states that the aggressive attackers can leverage the open nature of Crossbear to their advantage. Many of these attacks are inevitable and have to be dealt with in a different way. Hunters are not required to register themselves and also they are not given any IDs. One drawback of this method, feels the author is, it is easily possible for the attackers to send the forged certificate to the crossbear server. If an attacker deploys a 'malicious hunter' the can have difficulties in detecting such injections. Initially the attacker makes sure that all the connections of honest hunters are dropped and then forged certificate is sent across by the attacker thus by passing the security and MITM attack becomes

successful asserts Holz, R. (2012). Holz, R. (2012) also says that the Crossbear suffers a single point of failure. The attackers leveraging DDOS attack can flood the Crossbear servers with MiTM reports and try to inflict serious damages to the server feels Holz, R. (2012). Thus Holz, R. (2012) feels that the research should be conducted to let the Crossbear monitor the events continuously. Taking this research ahead, Szalachowski, P. (2018) came up with a solution to make the notary based systems stronger. It aims to improve the way notary systems behaved and develop a concept called Persistent and Accountable Domain Validation (PADVA). Szalachowski, P. (2018) argues that the notary based systems by providing their perspectives about the public keys of the domains can help in improving the security of the system. Notary can keep an eye on the public keys of the domains and we can observe the key continuity of those keys. Notary systems can perform better and give a better understanding of the fake certificates believes Szalachowski, P. (2018). Szalachowski, P. (2018) states that the notary systems were always neglected due to some issues with privacy, availability and security. To improve the systems the Szalachowski, P. (2018) urges to focuss on persistence, auditability, privacy and availability. According to Szalachowski, P. (2018), the keys should be authenticated consistently and periodically. This can improve the security level as the attacker would have to tackle persistent validation everytime he would send a request for connection. The notary systems should be open to audit by anyone. The author believes that the trust issues with the CA ecosystems and notary systems could be addressed in a better way. Szalachowski, P. (2018) also believes that the notary systems should be able preserve the privacy of the clients and should be always available. Taking the above mentioned things into account, Szalachowski, P. (2018) presented a system called Persistent and Accountable Domain Validation (PADVA). This system have been developed leveraging the blockchain technology. There are three main pillers of this system viz Server, Notary and Requester. The blockchain platform is being used to make all the transactions transperant and the smart contract concept of blockchain is used to automate the notary system which would ensure that the notary systems are available and are accountable by implementing the SLA states Szalachowski, P. (2018).

2.2 Blockchain based Solution

Just like above solution was developed to implement notary systems using blockchain technologies, few solutions have been implemented using the blockchain. A recent survey by (Karaarslan, E. and Adiguzel, E.) (2018) gives an overview of the blockchain technologies and their usage in securing the PKI implementations. The blockchain system (Karaarslan, E. and Adiguzel, E.) (2018) the author states is developed as a peer to peer network on nodes working on the same protocol. Each transaction is recorded and stored in a chain of blocks called as a

ledger. The system can work against tampering states (Karaarslan, E. and Adiguzel, E.) (2018). The nodes are connected to each other and their security is maintained by hash functions of cryptography viz SHA Algorithm. To validate the new blocks the all peers are required to connect and agree to it. Proof of Work (PoW) is used on a large scale. (Karaarslan, E. and Adiguzel, E.) (2018) states that the blockchain technologies are transparent and can provide good amount of security to the PKI. There are some issues the author states with respect to scalability as the protocol like PoW uses more resources states (Karaarslan, E. and Adiguzel, E.) (2018). (Karaarslan, E. and Adiguzel, E.) (2018) believes that there is a huge scope for trying out new solutions. In a similar research, Dykcik, L. (2018) presented a similar solution to prevent the MiTM attack on notary-based systems. Dykcik, L. (2018) used Ethereum blockchain technology. This system has been divided into four parts. Requesters, CA, Clients and Web Servers. This system mainly focusses on the resilience, automation and transparency parameters of notary services. The blockchain technology, Dykcik, L. (2018) states would provide the above features. Dykcik, L. (2018) Ethereum technology is the second most popular performing technology and mainly aimed at running smart contracts platform. A research was conducted by Atzei, N. (2016), on Ethereum smart contracts. Atzei, N. (2016) states that there are some vulnerabilities present in the Ethereum smart contracts. Atzei, N. (2016) says that the Ethereum should be executed properly and correctly to ensure that no one can tamper with the data. The author lists down some of the vulnerabilities of Ethereum Smart contracts. Call to unknown is a vulnerability in which some of the functions used by Solidity to call and shift may have an adverse impact of calling the fallback function. Atzei, N. (2016) also highlights that the Smart contracts may be vulnerable to other flaws like Exception disorder, Gasless send which may be responsible to the attacks like DDOS. It may also give rise to attacks like DAO, Man-in-the-Middle attack asserts Atzei, N. (2016). Apart from these attacks based on Smart Contracts, Atzei, N. (2016) also mentions about the low level attacks which may be possible on Ethereum Smart Contracts. Atzei, N. (2016) believes that these problems should be looked at very seriously so as to prevent the attacks happening the Ethereum smart contracts.

TLS 1.3

TLS 1.3 was developed in 2018 and is expected to provide a better security than the earlier versions, against known attacks like man-in-the-middle attack, spoofing attacks, etc. But according the latest recent release of information, the nccgroup found out some vulnerabilities in TLS libraries. They were discovered in August 2018 – nccgroup.trust (2019). These vulnerabilities have been found to be applicable to all the versions of TLS 1.3. A research was conducted by Ronen, E. (2019) to test the TLS

versions including TLS 1.3 against some of the popular attacks like MiTM. Ronen, E. (2019). To perform the testing Ronen, E. (2019) implemented TLS in about 9 different ways. Ronen, E. (2019) softwares like Amazons2n, OpenSSL, MozillaNSS, WolfSSL, etc. The author simulated MiTM attacks and performed padding Oracle attacks on all the mentioned nine implementations of TLS and got negative results for 7 out of the 9 implementations. After performing all the tests Ronen, E. (2019) concludes that even the latest version TLS 1.3 is not safe and is still vulnerable to padding oracle attacks which can lead to Man-in-The-Middle attack. Necessary steps need to be taken to improve the security of TLS 1.3 believes Ronen, E. (2019).

Thus from the above literature survey, it is evident that even latest technologies like TLS 1.3 are still vulnerable to Man-in-the-Middle attack. Thus this paper presents another way to tackle this issue.

3. Methodology

This paper presents a system called as Persisten and Accountable Domain Validation Extended (PADVAE). This system is based on an already implemented project called as Persistant and Accountable Domain Validation (PADVA) developed by Szalachowski, P. (2018). But instead of Ethereum blockchain, this paper presents Nebulas blockchain.

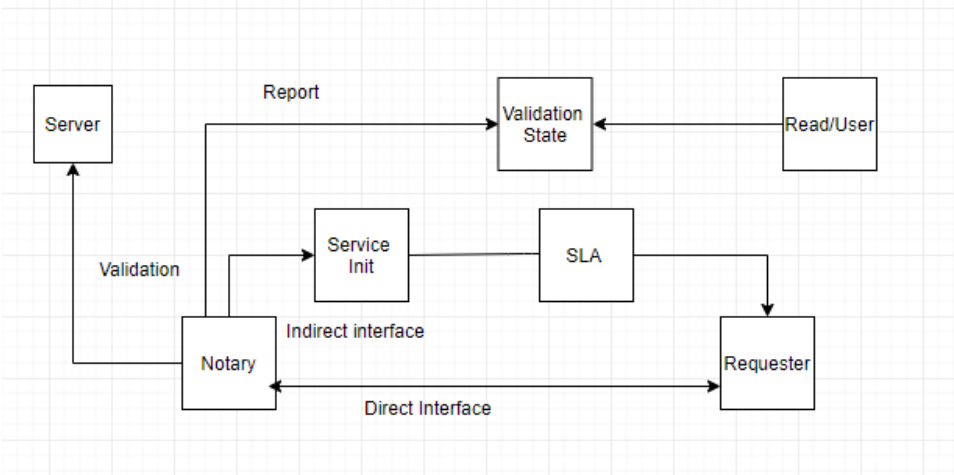

Figure 2: Block Diagram

This system is mainly divided into three categories:

- **Server**

It uses the TLS connection to render its service

- **Notary**

  It works as a third party auditor. It has to keep checking the servers and verifying their public keys.

- **Requester**

  It is more interested to check public keys of the servers. It can operate the server or and make the PADVAE service work from the notary.

The figure above shows the PADVAE system. As seen from the above diagram the Service init, validation state and the SLAs are the part of the blockchain technology. It works in the following way:

- The first step is about the notary service getting the requests from the requester upon transmitting the transaction fee to the notary service which incorporates the details like domain name to be verified along with its genuine public key and a transaction fee.
- The notary service takes a note of the service and sets up a service for serving the request. To ensure the SLA the notary deposits the fees as well. The notary by choice can be willingly refuse to serve the requests by citing approriate reasons like inacccessible server or misconfiguration.
- Each time after period T of validation, the notary pings the server to seek the validated and latest information about public key of the server. If the server notices any change in the output of validation, it publishes this information in the blockchain. Any user using blockchain can access this information. Else there is no need for the notary to publish the information. This implies the earlier state has not changed yet.
- To validate the correctness of the state the requester can connect with notary using direct interface. If the notary does not provide the requested information, the requester takes the indirect route, wherein the notary is forced to perform the work within the time period which is predefined. And if the server fails to do so the SLA gets executed and the contract of service terminates.

Szalachowski, P. (2018) uses the Ethereum blockchain technology. But as seen from the above literature there are some issues with the Scalability and Smart Contracts of Ethereum, we decided to implement the Nebulas blockchain.

Nebulas Blockchain

Nebulas blockchain was developed in the year 2018. Nebulas is a new blockchain technology which is based on Ethereum blockchan – Nebulas Technical White Paper. (2018). According to Nebulas Technical White Paper. (2018), the Nebulas tries to solve the problem of PoW regarding the ability to be able upgrade itself. The nebulas technology brings in scalability which was missing in Ethereum and the feature of self-evolving technology which will help improve the upgrading ability of the blockchain. The Nebulas also focusses the vulnerabilities in the Smart Contracts in Ethereum and tries to better it states Nebulas Technical White Paper. (2018).

4. Implementations

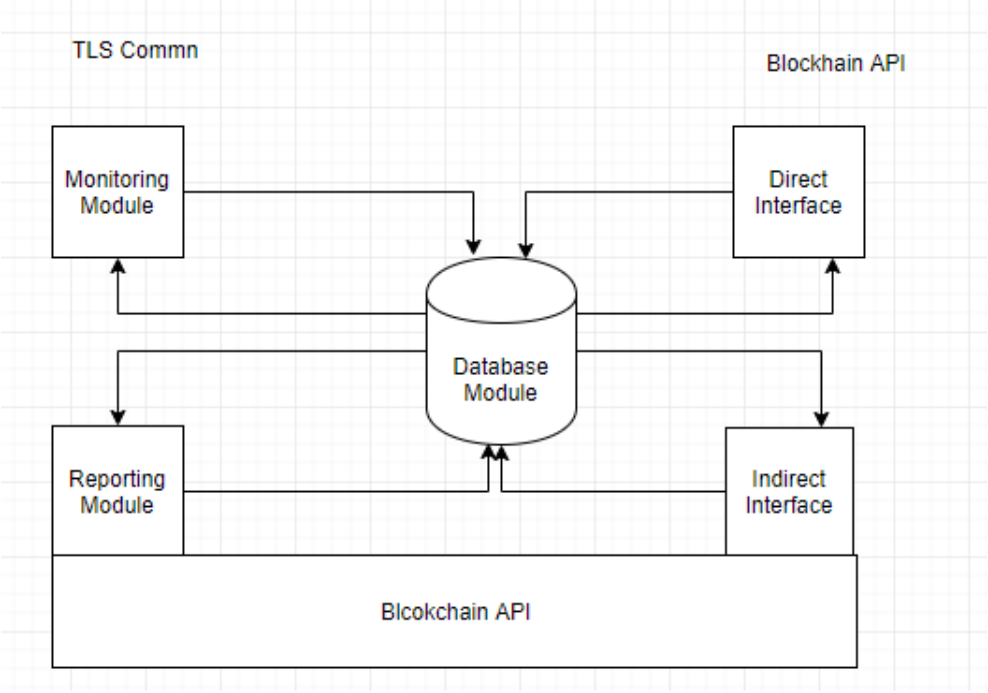

Figure 3: Implementation

Above diagram illustrates implementation. We followed the following the steps to implement the project:

1. Blockchain Implementation
2. Database Module
3. Direct Interface

4. Monitoring Module
5. Reporting Module
6. Indirect Interface

- **Blockchain Deployment**

Blockchain development is the first part of our coding. It forms the base of the project. Therefore, it was important that we implement the blockchain first. To implement the blockchain we used the JavaScript language. We followed the following steps to implement the blockchain.

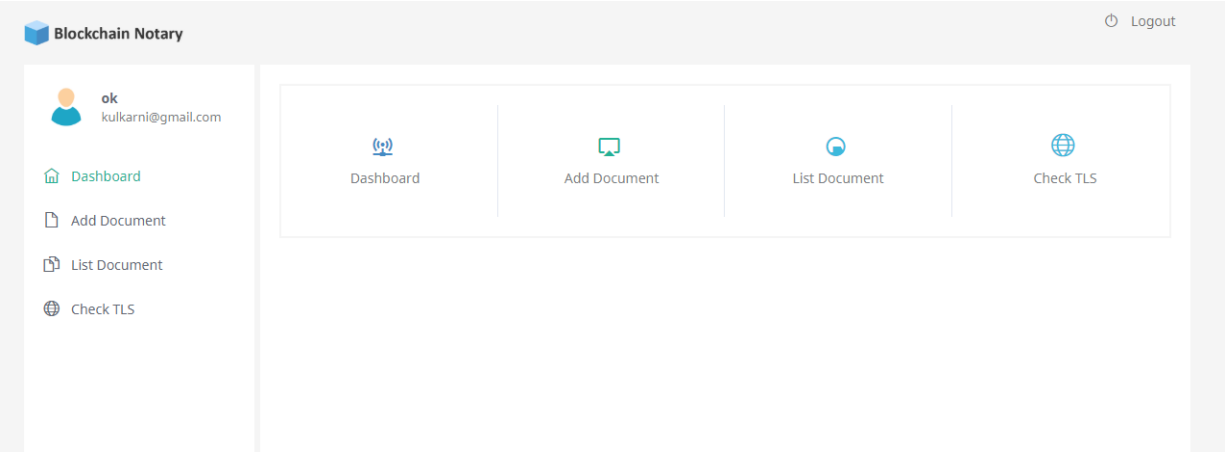

Figure 4: Blockchain Console

- Download, install and configure the NebPay software. The software has been taken from Github.com - GitHub. (2019).
- Generate and Initialize genesis block.
- Install and start like the localhost server.
- Generate address
- Save the private key in JSON format
- Load wallet through Pvt key and send token
- Check the transaction and hide
- Wait for the confirmation to complete

This completes our blockchain implementation.

- **Database Module**

  This module has been implemented after the blockchain, MySql have been used in this module.

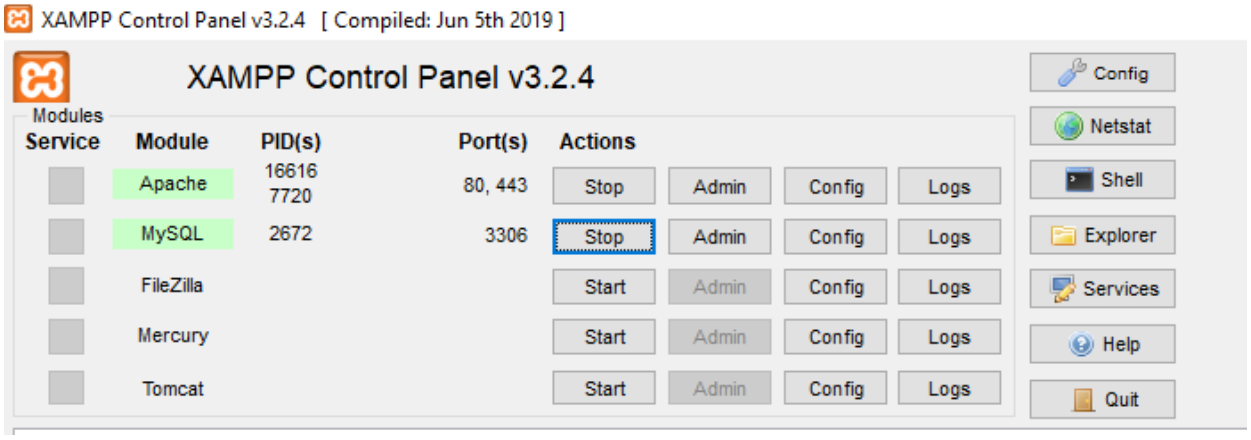

Figure 5: MySQL

- **Direct Interface**

  Direct interface development is the third part after the database implementation. It is responsible for any request that comes from the outside world to connect to the server. Therefore, there are two parts of the development viz client-side coding and server-side coding. Client side is a webpage designed using HTML, JavaScript and AJAX and the server-side coding is done using PHP. While coding the form validations have been implemented for better security of the system.

- **Monitoring Module**

  The monitoring module periodically performs TLS handshakes. To implement this module, we don't require any coding as it is a built-in feature in Blockchain which can be implemented.

- **Reporting Module**

  This module reports any changes in validations states to the blockchain. It is developed using NodeJS and PHP.

- **Indirect Interface**

  It handles SLAs and service initializations. It is developed using PHP, HTML, CSS

- **TLS**

TLS 1.2 was implemented in this project. We used some of the TLS libraries and checked the version of TLS. To implement the TLS, we created our own test certificate by running the bat file and stored in the browses trusted root CA. We can check the version of TLS using the following URL

https://www.howsmyssl.com/a/check

This completes the overall implementation.

5. Testing and Evaluation

Performing Man-in-the-Middle Attack on HTTPS:

Using tool called Wireshark, I tried sniffing the traffic and hunt for the credentials and got the following results.

```
        [iRTT: 0.000251000 seconds]
        [Bytes in flight: 594]
        [Bytes sent since last PSH flag: 594]
    [Timestamps]
        [Time since first frame in this TCP stream: 0.003371000 seconds]
        [Time since previous frame in this TCP stream: 0.000825000 seconds]
    TCP payload (594 bytes)
Secure Sockets Layer
    TLSv1.2 Record Layer: Application Data Protocol: http-over-tls
        Content Type: Application Data (23)
        Version: TLS 1.2 (0x0303)
        Length: 589
        Encrypted Application Data: 00000000000000013559e3c2485178ffd90506eb43180adb...
```

As we can see that the Application data is encrypted, which shows that the HTTPS was successful in protecting the data.

6. Drawbacks and Future works

TLS 1.3 is an emerging technology. It has been released in Feb 2018 and Szalachowski, P. (2018) believes that the upgrade to TLS 1.3 sooner is not something we should expect. Szalachowski, P. (2018) states that the TLS 1.3 is different from earlier versions of TLS 1.3 which might affect TLS 1.3, thus it can affect this system. The TLS 1.3 ceases to use GMT protocol in both client as well as server and may partially affect this system. This has been done to stop fingerprinting since the GMT could keep an eye on server and client. Owing to this change it would become difficult for the key usage to be recorded which might affect the system partially.

Another drawback is that there are no enough limitations to test the system. As the Nebulas platform is a new blockchain, there are relatively very limited to test the performance of this blockchain.

Therefore efforts should be made to deploy TLS 1.3 and this system together so that certificate transperancy is maintained. It is necessary to take into consideration that TLS 1.3 no longer supports GMT protocol. Therefore, external timestamping can be used to still make the system PADVAE functional. If TLS 1.3 version is deployed with this system, any of the lower versions could be deployed as external timestamp.

7. Conclusion

Thus as seen from the above findings, the PADVAE has been implemented and it is able to detect the man-in-the-middle attack. There are a few shorcomings in the implementation which needs to be addressed. Although it strives to keep the notaries transperant, auditable and available, a lot work can still be done on this system as discussed in the previous sections.

8. References

En.wikipedia.org. (2019). *HTTPS*. [online] Available at: https://en.wikipedia.org/wiki/HTTPS [Accessed 5 Aug. 2019].

Internet Society. (2019). *What is TLS & How Does it Work? | ISOC Internet Society*. [online] Available at: https://www.internetsociety.org/deploy360/tls/basics/ [Accessed 5 Aug. 2019].

Ibm.com. (2019). *IBM Knowledge Center*. [online] Available at: https://www.ibm.com/support/knowledgecenter/en/SSFKSJ_7.1.0/com.ibm.mq.doc/sy10660_.htm [Accessed 5 Aug. 2019].

DigiCert. (2019). *What Is SSL (Secure Sockets Layer)? | DigiCert.com*. [online] Available at: https://www.digicert.com/ssl/ [Accessed 7 Aug. 2019].

Szalachowski, P. (2018). Blockchain-based TLS Notary Service.

Clark, J. (2013). SoK: SSL and HTTPS: Revisiting Past Challenges and Evaluating Certificate Trust Model Enhancements. *IEEE*.

Conti, M. (2016). A Survey of Man In The Middle Attacks. *IEEE*, 18(3).

Holz, R. (2012). X.509 Forensics: Detecting and Localising the SSL/TLS Men-in-the-Middle. *Springer*.

Karaarslan, E. and Adiguzel, E. (2018). Blockchain Based DNS and PKI Solutions. *IEEE Communications Standards Magazine*, 2(3), pp.52-57.

Dykcik, L. (2018). BlockPKI: An Automated, Resilient, and Transparent Public-Key Infrastructure.

Atzei, N. (2016). A Survey of Attacks on Ethereum Smart Contracts SoK.

Nccgroup.trust. (2019). [online] Available at: https://www.nccgroup.trust/uk/about-us/newsroom-and-events/blogs/2019/february/downgrade-attack-on-tls-1.3-and-vulnerabilities-in-major-tls-libraries/ [Accessed 9 Aug. 2019].

Ronen, E. (2019). Ronen, E. (2019). The 9 Lives of Bleichenbacher's CAT: New Cache ATtacks on TLS Implementations.

Nebulas Technical White Paper. (2018).

GitHub. (2019). *nebulasio/nebPay*. [online] Available at: https://github.com/nebulasio/nebPay [Accessed 10 Aug. 2019].