*Attacking SSL/TLS Implementations*

Shawn Philip Babu  
Information Systems Security  
Concordia UniversityMontreal, Canada  
shawn0111babu@mail.com

Jemy Amrutia

Information Systems Security

Concordia University

Montreal, Canada.

jemyamrutia@gmail.com

Ekasmeet Gahir

Information Systems Security

Concordia University

Montreal, Canada.

ekasmeet97@gmail.com

line 1: 4th Given Name Surname  
line 2: *dept. name of organization (of Affiliation)*  
line 3: *name of organization (of Affiliation)*line 4: City, Country  
line 5: email address or ORCID

line 1: 5th Given Name Surname  
line 2: *dept. name of organization (of Affiliation)*  
line 3: *name of organization (of Affiliation)*line 4: City, Country  
line 5: email address or ORCID

line 1: 6th Given Name Surname  
line 2: *dept. name of organization (of Affiliation)*  
line 3: *name of organization (of Affiliation)*line 4: City, Country  
line 5: email address or ORCID

*Abstract*—SSL stands for Secure Sockets Layer and, it's the standard technology for keeping an internet connection secure and safeguarding any sensitive data that is being sent between two systems, preventing criminals from reading and modifying any information transferred, including potential personal details. TLS (Transport Layer Security) is just an updated, more secure, version of SSL. We still refer to security certificates as SSL because it is a more commonly used term, but when you are [buying SSL](https://www.websecurity.digicert.com/ssl-certificate?inid=infoctr_buylink_sslhome) from DigiCert you are actually buying the most up to date TLS certificates with the option of [ECC, RSA or DSA encryption](https://www.websecurity.digicert.com/security-topics/how-ssl-works). However, there are undeniable differences between the libraries that implement SSL/TLS protocol and vulnerabilities in these libraries. Hence, the two main questions asked are: what’s the difference between TLS vs SSL? And is it something we need to worry about? In this report, we summarize some of the limitations by considering implementations of each along with review of past protocol-based and software-based vulnerabilities.

# Introduction

The Secure Sockets Layer (SSL) protocol is a security technology that offers online privacy. The protocol enables secure communication between client and server applications. Clients can choose to authenticate themselves, while servers are always authenticated. There is a big demand right now for SSL certificates. The encryption landscape has undergone a substantial transformation since Google launched its "HTTPS Everywhere" campaign. Because not all servers provide web interfaces for managing SSL, OpenSSL can be your only choice for some systems to import and setup your certificate. OpenSSL is quite helpful if you want to speed up the entire process or don't have access to an online administration panel. With just a few OpenSSL lines, it is possible to generate the Certificate Signing Request and the private key, combine files, review the certificate's specifics, and address any potential issues. An outdated cryptographic technique that encrypts network interactions between two computer applications is called Secure Sockets Layer, or SSL. Transmission Layer Security) certificates are what we mean when we talk about SSL certificates. OpenSSL is a cryptographic tool that uses command lines to control SSL/TLS certificate creation, installation, and identification.

A web server (i.e., a website) that is SSL-secured is attempted to be contacted by a browser or server. The browser/server asks for the web server's identity. The browser or server is then sent a copy of the web server's SSL certificate. This check determines if a browser or server trusts the SSL certificate. If so, a message alerts the web server. The web server sends back a digitally signed acknowledgement to start an SSL encrypted session. Eventually, encrypted data is exchanged between the web server and browser/server. Website owners must get an SSL certificate from a certificate authority and install it on the web server (often a web host can handle this process). An impartial third party that may attest to the legitimacy of the website owner is known as a certificate authority. They maintain records of the certifications they issue. Online payments conducted via a credit card or another way are included in the SSL apps. Intranet-based traffic includes, but is not limited to, internal networks, file sharing, extranets, and database access. Webmail servers include Exchange, Office Communications Server, and Outlook Web Access. The connection between an email client like Microsoft Outlook and a mail server like Microsoft Exchange.

**SSL Benefits and Drawbacks: -**

Benefits: -

1.Improving Data Security: -

Stronger data security is one of the major advantages of SSL certificates for websites. They protect the data in transit while securing connections between the browser and server thanks to cryptography. Even if there is a data breach, the attacker won't be able to fully decrypt and comprehend all of the data because of how highly and intricately encrypted it is.

2.Identity Verification and Authentication: -

Data that is transmitted via the internet is moved between different organisations with a significant potential of ending up in the hands of intruders or other unauthorised third parties. The fact that SSL confirms and authenticates the parties' identities is another important benefit.

3. Avoid a Variety of Attacks: -

The ability of SSL certificates to protect websites and users from a variety of threats is another significant advantage. As it encrypts all data in transit, it helps prevent eavesdropping, impersonation, data theft, identity theft, and Man-in-the-Middle attacks.

Drawbacks: -

1.Cost: -

Free SSL certificates are offered, however they do not provide the necessary level of security and encryption. One must pick a CA who can offer the highest levels of protection while adhering to industry standards and rigorous compliance standards. This CA must have the necessary infrastructure, knowledge, and reputation. There is a price for this. Your fees rise if you have many domains and sub-domains.

2. Expiry: -

For continuous security, expired SSL certificates must have their expiration date monitored and renewed. In the absence of sight, the process of monitoring and renewal may be more difficult. Reputable suppliers like Entrust from Indusface offer cutting-edge Certificate Management Systems (CMS) to administer SSL easily and effectively in order to address this SSL drawback.

3.SSL-Related Vulnerabilities: -

Although SSL has many advantages for the website, it also has several security flaws. There are numerous vulnerabilities in the older SSL/TLS protocols, including POODLE, BEAST, Heartbleed, CRIME, and TLS 1.0 or TLS 1.1. Browsers label websites with certificates that use these antiquated protocols as unsafe.

**Why the developers decided on new TLS implementation instead of already using open SSL library?**

First off, developers can require particular features or components that are absent from the libraries that are currently available. Creating a custom TLS implementation from scratch allows developers to create a solution that is tailored to their needs. Second, developers can be concerned about the safety or calibre of the current libraries. Developers can have more control over the security and quality of the codebase with a new TLS implementation, which may be essential for applications that need security. Finally, developers may elect to employ a new TLS implementation as a teaching tool or as a means of better understanding the underlying technology. The process of developing a new implementation can help developers learn more about how TLS works and progress their careers.

**TLS OVER SSL**

TLS offers a more secure way to handle authentication and message exchange. TLS uses the more secure Key-Hashing for Message Authentication Code (HMAC) to ensure that a record cannot be changed during transmission over an open network like the Internet, whereas SSL uses keyed message authentication.TLS specifies the Improved Pseudorandom Function (PRF), which generates key information with the HMAC using two hash methods. By prohibiting data from being altered if only one algorithm is compromised, two algorithms strengthen security. As long as the second algorithm is not compromised, the data is secure.

TLS uses PRF and HMAC values in the message to provide a more secure authentication technique than SSL, which both send messages to each node to verify that the exchanged communications were not tampered with. The TLS protocol specifies the sort of certificate that must be sent between nodes in order to guarantee more consistency. Moreover,

TLS offers more detailed alerts concerning session issues and keeps track of when specific alarms are given.

They were identical for a while to make the switch from SSL to TLS easier. The only differences between SSL v3 and TLS 1.0 were a few minor protocol features. This made it simple to keep using SSL, as we had for so many years, rather than encouraging people to use the new TLS language. Some individuals began to use the names interchangeably. The harm had already been done when TLS 1.1 (and later) replaced SSL as the only reliable protocols when it was phased out (deprecated).

**What is the difference between HTTP and HTTPS?**

HTTPS stands for "secure" and has a S in it. HTTP over SSL/TLS is just HTTPS. Traffic to and from a website with an HTTPS address is authenticated and encrypted using the SSL/TLS protocol, and that website has a valid SSL certificate that was issued by a certificate authority. Several web browsers have started to flag HTTP pages as "not secure" or "unsafe," in an effort to persuade users to switch to the more secure HTTPS protocol. Hence, HTTPS has become crucial for establishing user trust in addition to being necessary for keeping consumers safe and user data secure.

**What is TLS**

An encryption system called Transport Layer Security was created to provide end-to-end security for web-based communications. To prevent manipulation and eavesdropping, the Internet Engineering Task Force (IETF) adopted TLS as the default protocol.

Users and online apps frequently run into many potential security issues when accessing the internet. They consist of verifying the other party's identification, data manipulation, and third-party surveillance. TLS uses cryptographic methods to protect users while they browse online, help ensure the integrity of the data being exchanged, and authenticate the client or server in a connection.

TLS is often associated with secure web browsing, which guards against hackers and eavesdroppers during online transactions. The padlock icon at the top left corner of the web browser denotes secure browsing sessions. TLS is also utilised by programmes like email, file transfers, and audio and video conferencing. The vast majority of protocols, including HTTP, SMTP, FTP, XMPP, and many others, are also interoperable with TLS. Users should be aware that only data transported over the internet is intended to be secured by TLS; data on end systems is not.

**How does TLS work?**

To assist provide a secure connection between two or more interacting programmes, assure device interoperability, and run reasonably efficiently, TLS security is designed to use encryption on both the client and server sides.

The first step in client-server communication is choosing whether to use TLS protocols or not. There are numerous methods the client might specify a TLS connection. The client might, for instance, utilise a port that supports the encryption types used in TLS connections. Making a protocol-specific request to transition to a TLS connection is another possible approach.

The TLS protocol definition moves via two layers, the TLS handshake protocol and the TLS record protocol, once the client and server have decided to interact over TLS. Combining symmetric and asymmetric cryptography is a feature of TLS protocols. Asymmetric cryptography produces key pairs, one public (shared by the sender and receiver) and one private, as opposed to symmetric cryptography, which generates keys known to both the sender and recipient.

The TLS handshake protocol defines the standards necessary to exchange an application "message." Depending on the supported cypher suites and the key exchange mechanism in use, a TLS handshake involves a series of exchanges between the client and server that can go something like this:

● A client initiates a connection by sending a "client hello" message that includes a list of supported cypher suites (a group of encryption techniques needed to create a secure connection) and a client random, or random sequence of bytes.

● The chosen cypher suite, the chosen TLS protocol version (1.0, 1.2, etc.), and a random sequence of bytes (referred to as the "server random") are all included in the server's "server hello" message.

● For authentication, the server transmits the client its SSL certificate. When the server requests it, the client can also transmit a certificate for authentication in addition to authenticating the server by checking the SSL certificate.

● The "premaster secret," which is a second string of random bytes, is sent by the client. The premaster secret is first encrypted by the client using asymmetric cryptography to create a public key from the server's security certificate. Only the server has the private key necessary to decrypt the premaster secret.

● With the help of the private key, the server decrypts the premaster secret.

● Client and server both produce session keys using the premaster secret, client random, and server random.

● A "completed" message that has been encrypted using the session key is sent by the client.

● The server replies with an encrypted "done" message using the session key.

● Secure symmetric encryption has been accomplished by the client and server, concluding the handshake and allowing communication to proceed using the pre-agreed session keys.

The TLS record protocol uses symmetric cryptography to create distinct session keys for each connection after the decryption technique is decided upon during the handshake stage. These keys allow for ongoing communication throughout the session. The record protocol also adds a message authentication code based on hashing to any data being delivered (HMAC).

Users should anticipate using some computer power in the process because TLS's encryption methods are complicated. TLS, however, also has internal mechanisms in place to avoid material lags. The performance and load times of online applications shouldn't be significantly impacted by TLS protocols, and most enterprises shouldn't see an increase in computing expenditures.

**TLS Implementation**

1. TLS STACK:

A TLS (Transport Layer Security) stack is a particular way to implement the TLS protocol, which enables secure internet connection. A TLS stack is often made up of several software layers that cooperate to offer secure communication between two endpoints.

A cryptographic library that offers cryptographic operations like encryption, decryption, hashing, and key exchange is a component of a TLS stack at the lowest level. The negotiation and initialization of the TLS session are taken care of by a protocol implementation layer. For secure communication, there is a record layer that offers message framing and fragmentation. The interface between the TLS stack and the application employing secure communication is provided by the application layer, in the end.

2. Cipher Suites:

A collection of cryptographic methods used in the TLS (Transport Layer Security) protocol are referred to as cypher suites.

● Algorithm for Key Exchange: This algorithm is employed to safely transfer cryptographic keys between the two endpoints. example: RSA and Diffie-Hellman (DH).

● Data transferred between the two endpoints is encrypted using an encryption technique, such as Advanced Encryption Standard (AES) or Triple Data Encryption Standard (3DES).

● The integrity of the data transported between the two endpoints is guaranteed by the Message Authentication Code (MAC) Algorithm.

● for instance, Hash-based Message Authentication Code (HMAC)

● The hashing method is used to make sure the data being sent between the two endpoints is legitimate. for example, SHA-256 and SHA-384.

3. TLS Configuration:

The parameters and settings used to create a secure connection between two endpoints using the TLS protocol are referred to as TLS (Transport Layer Security) configuration.

The TLS version that will be applied to the connection. TLS 1.2 and TLS 1.3 are versions that are frequently used. To authenticate endpoints and create trust between them, TLS uses digital certificates. Resuming previously started sessions is supported by TLS, which can boost efficiency and cut down on overhead. There are many other parameters that are part of the TLS setup, such as protocol-level settings, certificate revocation settings, and other security-related settings.

**Purpose of TLS**

Most websites now use TLS encryption as normal procedure to safeguard web apps from data manipulation and eavesdropping. The SSL/TLS protocols were created in response to the growing security risks and the requirement for encryption on both the client and server sides.

TLS is in place to support the protection of user security and privacy. Without TLS, sensitive data being exchanged online, including login credentials, personal data, and credit card numbers, is susceptible to theft. Moreover, emails, browsing patterns, and direct message interactions could all be observed by unidentified third parties.

TLS helps defend online applications against data breaches and distributed denial-of-service (DDoS) assaults in addition to protecting the personal information of individual users. For businesses of all sizes, data breaches and DDoS assaults can prove to be extremely expensive and harm consumer trust irreparably. It is simple to increase security and contribute to the protection of both user and corporate privacy by making sure web browsers are using TLS.

TLS support is now standard in the majority of browsers. Google Chrome, for instance, regularly warns people away from websites that are not HTTPS. User awareness of website security and the need to look for secure data transfer protocols is growing as a result. Organizations and people can contribute to ensuring a minimal level of shared protection for web-based activities by insisting on the required use of TLS in all web-based communications.

That being said, TLS protocols have been breached in the past ten years, including BEAST in 2011, CRIME in 2012, BREACH in 2013, and Heartbleed in 2014. Yet since then, TLS protocols have undergone significant updates aimed at removing problematic code and adding enhancements in security, performance, and privacy. The handshake step is accelerated by the most recent TLS version (1.3), which also speeds up the encryption process. Also, it gets rid of outdated algorithms that led to security holes in earlier iterations.

# Limitation of SSL/TLS Implementations

SHAWN & HARSHITHA

(Times new roman 10)

# How SSL/TLS secure data ?

VENKY & SANDEEP

(Times new roman 10)

# Comparisions of DIfferent versions of SSL/TLS

VENKY & SANDEEP

(Times new roman 10)

# Vulnerabilities in SSL/TLS-Protocol based

VANI & SAUMYA

(Times new roman 10)

# Vulnerabilities in SSL/TLS-Software based

PRANATHI & SARAT

(Times new roman 10)

# Concluding Thoughts

##### References

1. Ćurguz, Jelena. (2016). Vulnerabilities of the SSL/TLS Protocol. Computer Science & Information Technology. 6. 245-256. 10.5121/csit.2016.60620.
2. Implementing TLS with Verified Cryptographic Security Karthikeyan Bhargavan∗, Cedric Fournet ´ †, Markulf Kohlweiss†, Alfredo Pironti∗, Pierre-Yves Strub‡ ∗INRIA Paris-Rocquencourt, {karthikeyan.bhargavan,alfredo.pironti}@inria.fr †Microsoft Research, {fournet,markulf}@microsoft.com
3. SoK: SSL and HTTPS: Revisiting past challenges and evaluating certificate trust model enhancements Jeremy Clark and Paul C. van Oorschot School of Computer Science Carleton University, Canada {clark,paulv}@scs.carleton.ca
4. https://www.cloudflare.com/learning/ssl/how-does-ssl-work/
5. Lee, Kwang-Hyoung & Park, Jeong-Hyo & Jun, Moon-Seog. (2014). Key-pair(Public key, Private key) conflict analysis using OpenSSL. Journal of the Korea Academia-Industrial cooperation Society. 15. 5294-5302. 10.5762/KAIS.2014.15.8.5294.