Comparative Analysis of HTTPS / TLS Implementations for Healthcare Web Applications

Vladyslav Malanin

V. M. Glushkov Institute of Cybernetics of the NAS of Ukraine, Kyiv, Ukraine
Email: *vladislav.malanin[at]gmail.com*

Abstract: ***In our digital age, keeping patient information safe during its transfer is critically important. HTTPS which relies on the Internet's Transport Layer Security (TLS) protocol, is the primary system used to protect website applications. On the other hand, TLS can provide different outcomes in terms of security, speed or compatibility because not all programs are built the same. This paper discusses the suitability of OpenSSL, BoringSSL, WolfSSL and Microsoft SChannel for healthcare web application use. Exploring the libraries in terms of cryptography, how much extra effort they require, how rules are handled and the way they involve other technologies, this study points out the pluses and minuses of all approaches. To illustrate the way health systems are implemented and the issues that generally arise, real examples and benchmarks are discussed. The recommendations in the findings help developers and architects implant secure, compliant and efficient healthcare web services. It is highlighted that choosing and setting up the appropriate TLS technology is necessary to keep patient data secure and maintain faith in virtual healthcare.***

Keywords: HTTPS, TLS, healthcare security, web application security, OpenSSL, BoringSSL, WolfSSL, certificate management, HIPAA, encryption protocols

1. Introduction

1.1 A background overview of web application security is provided.

Because technology is taking over healthcare, people are now using web technology to store their patient records, make appointments, have remote consultations and manage payments. Since these platforms manage private information and health records, they are easy targets for cybercriminals. Protecting the confidentiality, integrity and accessibility of information is a main issue in today's web application security (Aliyu & Awotunde, 2021).

A range of methods and solutions are used in web application security to avoid unauthorized entry, data losses and problems with internet services. As a result, vital communication protocols such as HTTPS which relies on the TLS protocol, are now essential. Thanks to HTTPS, any data passed between a website and a browser is encrypted, so threats such as MITM attacks, eavesdropping and data alteration can be prevented (Rescorla, 2018).

It ensures confidentiality by encrypting messages, verifies the message's integrity with a hash function and ensures authentication using digital certificates. However, different software may provide a range of TLS security. Using different protocols, algorithms, certificate validation ways and default configurations may lead to various risks. As an example, the Heartbleed bug (CVE - 2014 - 0160) in OpenSSL and the flawed FREAK attack impacted millions of servers running SSL (Durumeric et al., 2015; Beurdouche et al., 2015).

Failure to maintain security in healthcare may result in safety problems for patients, damage to an institution's reputation and breaches of rules. These regulations force businesses in the U. S. and EU to handle data securely and with encryption. There are both monetary fines and legal problems for any violations.

Since the risks are significant, healthcare IT specialists and developers ought to carefully evaluate the positives and negatives of different TLS options. This paper looks at different HTTPS/TLS libraries and reviews their speed, protection offered and level of compliance with healthcare laws and regulations.

1.2 Use of HTTPS/TLS Protocols

To make HTTP secure, HTTPS adds TLS on top of the regular HTTP protocol. The healthcare sector relies on TLS to ensure data is securely shared over the internet. HTTPS is vital for electronic health information as it provides confidentiality, integrity and authentication (Rescorla, 2018). Encrypting patients' medical records, ensuring communication between doctors and patients is secure, protecting passwords and following guidelines such as HIPAA and GDPR are all tasks handled by HTTPS and TLS in healthcare web applications. Senders and receivers of website data are not protected from eavesdropping, tampering and impersonation if HTTPS/TLS is not in use (Beurdouche et al., 2015).

After initiating a handshake, TLS allows the client and server to negotiate which cryptographic details to use, prove their identity through certificates and then build a secure session key using either RSA or Diffie - Hellman techniques. Thanks to the introduction of TLS 1.3, both the safety and the speed of TLS have improved greatly by eliminating weak cryptographic tools and reducing the complexity of the initial exchange (IETF RFC 8446, 2018).

The table screenshot shows the major roles that TLS plays in healthcare web applications.

Table 1*:* Key Functions of TLS in Healthcare Web Applications

Function	Description	Healthcare Relevance
Encryption	Protects data in transit using symmetric cryptography	Secures patient records, diagnoses, and prescriptions
Authentication	Validates server (and optionally client) identity using digital certificates	Prevents impersonation of healthcare providers or portals
Integrity	Ensures data has not been tampered with during transmission	Protects against MITM attacks and data alteration
Forward Secrecy	Ensures session keys are not compromised even if long - term keys are	Limits damage from future key leaks (important for EHRs)
Session Resumption	Improves performance by reusing session parameters securely	Enhances user experience in patient portals and mobile apps

By implementing HTTPS/TLS correctly, healthcare providers can significantly reduce their risk surface and build trust with patients and regulators. Conversely, misconfigured or outdated TLS versions—such as TLS 1.0/1.1, which have known vulnerabilities—can expose organizations to breaches and legal liability (NIST, 2020).

1.3 Unique Security Concerns in Healthcare

What makes healthcare security unique?

In healthcare, there are specific cybersecurity issues as the data involves sensitive, high - volume and crucial information. While most web applications manage information that is not urgent, healthcare systems handle information such as patient medical records, unique identification traits and instant watch data. When patient data is not managed properly, this can lead to problems with patient privacy, risks for patients and damages to a hospital's reputation (McLeod & Dolezel, 2018).

1) **Healthcare data is extremely valuable.**
   The sale of healthcare records is greatly sought after by those in the black market, outpacing the value of financial information. Since an EHR contains name, social security number, medical profile and insurance plan, they often sell for hundreds of dollars on the black market (Ponemon Institute, 2023).
2) **There are many regulations and laws involved in transporting goods.**
   Healthcare organizations are required to follow strict rules for protecting data. Three major organizations in the U. S. and Europe, HIPAA and GDPR, make sure that data standards are maintained and protected at all times. If organizations do not properly set up TLS and let data experience insecure transmission, they may receive hefty fines and damage their image (U. S. HHS 2022; European Commission 2018).
3) **Systems running on legacy technology and the interoperability of different systems**
   Many institutions operating in the healthcare industry still work with systems that do not have effective encryption security. It's not always easy to integrate TLS with these platforms which raises the possibility of errors that can result in a switch to less secure TLS 1.0 and SSL 3.0 (Alasmary et al., 2020).
4) **Always require up - to - the - minute data from IPC.**
   These applications must have constant data transmission that is rapid and well protected. If the handshake process in a TLS system is complicated or not optimized for quick network connections, service can become unavailable or slower (Mavroudis, Bougioukas, & Petridis, 2021).
5) **Insiders and Poor Configuration**
   A lot of health system breaches are triggered by employees mishandling confidential information or improper installation of controls. A system using HTTPS can still be vulnerable if the TLS certificates are mismanaged, if weak ciphers are used or if software libraries are not patched. Because of these issues, healthcare web apps should always run TLS in a well - secured, well - maintained and proper way. Simply turning on HTTPS is not enough; the effectiveness and correctness of TLS and its configuration must be tested as well.

2. Evolution from HTTP to HTTPS

Due to the need for secrecy, security and integrity in sensitive information, Internet sectors like healthcare motivated the change from HTTP to HTTPS. In the early days, HTTP was designed with the aim of sending information quickly and simply over the internet. Nevertheless, HTTP sends data in normal text, so it can easily be viewed, changed or falsified by anyone who intercepts traffic (Rescorla, 2018). To make the system more secure, encryption was applied to the HTTP protocol. At first, most solutions used Secure Sockets Layer (SSL) to secure the data transmission between a web client and the server. Despite how useful SSL 2.0 and 3.0 were, it did not take long before they were exposed to serious problems, like susceptibility to downgrade and padding oracle attacks (e. g., POODLE) (Beurdouche et al., 2015).

Instead of using SSL, the modern practice today is Transport Layer Security (TLS). With every update to TLS, security was improved and the program ran more efficiently. Both TLS 1.2 and newer TLS 1.3 come with tough encryption algorithms, defense from famous attacks and less communication required during an SSL session setup. Besides other advantages, TLS 1.3 is the best and most secure TLS version now, due to not supporting old cryptographic methods, relying on AEAD ciphers and having forward secrecy turned on (IETF RFC 8446, 2018). Since essential web applications in healthcare deal with constant, interactive and private details (for example, EHR access and teleconsultations), the latest versions of HTTPS and TLS are absolutely required due to regulations. It is required by HIPAA and GDPR to secure data that is sent between two places. The use of TLS 1.0 and TLS 1.1 is not allowed in any real environment, especially when handling Protected Health Information.

Table 2: Evolution from HTTP to HTTPS with TLS

Protocol	Introduced	Security Features	Status	Use in Healthcare
HTTP/1.0	1996	None (plaintext communication)	Deprecated	Not compliant with any healthcare standards
HTTP/1.1	1997	None by default; used with SSL/TLS later	Still in use	Requires HTTPS wrapper for compliance
HTTPS (SSL 3.0)	1999	Encryption, basic certificate validation	Deprecated	Vulnerable (e. g., POODLE attack)
TLS 1.0/1.1	1999/2006	Improved over SSL; basic encryption	Obsolete (NIST, 2020)	Non - compliant with HIPAA since 2020
TLS 1.2	2008	Strong encryption, support for AEAD ciphers	Widely adopted	Compliant; supports HIPAA/GDPR requirements
TLS 1.3	2018	Forward secrecy, faster handshakes, removes legacy features	Emerging standard	Ideal for real - time healthcare systems

Key Takeaways for Healthcare Web Applications:

1) **Compliance**: Regulatory bodies now mandate TLS 1.2 or higher for systems handling sensitive medical data.
2) **Performance**: TLS 1.3 improves latency and connection times, ideal for real - time healthcare services like telehealth and IoMT (Internet of Medical Things).
3) **Interoperability**: While TLS 1.3 is superior, not all legacy systems support it. Organizations may need to bridge compatibility gaps during transitions.
4) **Security**: TLS 1.3 enforces best practices by default, reducing the risk of misconfiguration—a common cause of healthcare data breaches (Verizon DBIR, 2023).
5) As threats continue to evolve, the use of outdated HTTPS/TLS implementations can expose healthcare providers to both cybersecurity risk and legal liabilities. Therefore, a timely migration to modern TLS standards is essential for any healthcare organization aiming to ensure patient safety, data integrity, and trustworthiness in digital services.
6) **Key Features of TLS (Handshake, Encryption, Certificates)**
7) TLS is meant to secure information sent or received over networks that may be insecure. Confidentiality, integrity and authenticity of transmitted data are protected by several important features in a network. TLS handshake, different kinds of encryption and certificates are the main features of TLS.

a) TLS Handshake

Before starting communication, the TLS protocol requires a client and server to negotiate their security settings first. During phase three, both parties select cryptographic requirements such as the version of the protocol and the encryption methods and also establish the keys used for encryption. The process of a handshake can be summarized into three main steps (Dierks & Rescorla, 2008):

- The client and server negotiate the highest version of TLS that they agree on.
- Both agree on which algorithms to use for exchanging keys, encrypting messages and message authenticity.
- Presentation of Certificate: The server shows its digital certificate provided by a trusted Certificate Authority (CA) to be recognized.
- Both users exchange cryptographic keys to create a session key that is secure. TLS today uses ephemeral key exchanges such as ECDHE, to achieve forward secrecy.

If the verification process is successful, the session keys are turned on and the connection becomes secure. By using this handshake, they can ensure only allowed parties join the communication and protect their data with encryption.

b) Encryption

After the handshake, TLS relies on symmetric encryption as it uses less computing power. In 2018, Advanced Encryption Standard (AES) and ChaCha20 - Poly1305 were identified as typical algorithms that encrypt data and also safeguard it from various cyber - attacks (Rescorla, 2018). When data is encrypted, it becomes extremely difficult for anyone unauthorized to access patient files and images while they are being sent.

c) Certificates and authority of documents

A digital certificate matches the public key of an entity with its identity and it is typically provided by a trusted CA. In healthcare, certificates prove to clients that the communication is authentic and not faked (Mavroudis et al., 2021). Certificates can participate in certificate chains and support functions that verify if they should be kept or replaced (such as OCSP). Certificates should be properly managed and verified, especially where the rules for compliance and data accuracy are very strict.

d) Many websites prefer to use TLS 1.2 and the new version TLS 1.3

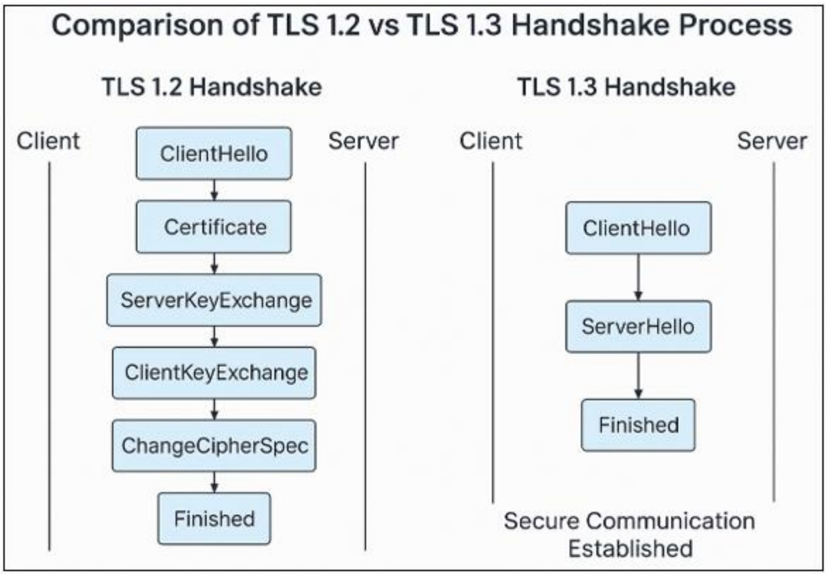
Both the security and performance of TLS have improved as new versions have been released. Hosting web applications today usually involves using TLS 1.2 or 1.3, both having different effects on healthcare web application security.

TLS 1.2: Introduced in 2008, TLS 1.2 included strong cipher suites, supported AES with GCM and offered users the choice of several hash algorithms used in message authentication (Dierks & Rescorla, 2008). Java is popular thanks to its ability to work with many platforms.

Yet, using TLS 1.2 increases the time it takes for connections to be established due to its involvement in several trips which could influence telemedicine or sending emergency data.

TLS 1.3: TLS 1.3 came out in 2018, operating under a new structure to ensure both greater security and a faster performance. By using only one round trip, the handshake can be completed much faster (IETF RFC 8446, 2018). TLS 1.3 does away with older and insecure options such as the use of RSA for keys and weaker ciphers.
This means that each session in TLS 1.3 uses forward secrecy, making it impossible for attackers to read past information even if they access session keys later.

Diagram: Comparison of TLS 1.2 vs TLS 1.3 Handshake Process



How Web App Security can be used in the healthcare sector

TLS 1.3 does away with weaker cryptographic methods and ensures better security, thus limiting risks and attacks. Because of this, healthcare web applications are able to protect the privacy and accuracy of patient data (Mavroudis et al., 2021). By making latency in TLS 1.3 less noticeable, users of telehealth and monitoring applications experience better service. Today’s standard for banks: Following many of the regulations’ guidelines, most financial institutions now use TLS 1.2 or the newer TLS 1.3.

3. Security Requirements in Healthcare Applications

Healthcare web applications must adhere to stringent security requirements due to the sensitivity of the data they handle and the potential impact of security breaches on patient safety and privacy. This section explores the regulatory landscape**,** the types of data transmitted, and the threat models and risks that are especially relevant in the healthcare domain.

1) Regulatory Requirements

Healthcare applications operate under strict regulatory frameworks designed to protect patient information and ensure its confidentiality, integrity, and availability. The most prominent regulations include:

a) **Health Insurance Portability and Accountability Act (HIPAA):** A U. S. federal law that mandates standards for protecting sensitive patient health information. HIPAA requires covered entities to implement technical safeguards, such as encryption for data in transit and at rest, and access controls to prevent unauthorized disclosures (U. S. HHS, 2022).

b) **General Data Protection Regulation (GDPR):** The European Union’s comprehensive data protection regulation imposes strict rules on the processing and transmission of personal data, including healthcare information. GDPR emphasizes data minimization, explicit consent, and the right to be forgotten, requiring robust security measures in web applications handling EU residents’ data (European Commission, 2018).

c) **Other Regional Laws**: Various countries have additional laws (e. g., Canada’s Personal Health Information Protection Act, Australia’s Privacy Act) that impose similar data security and privacy requirements.

These regulations collectively mandate the use of strong encryption protocols**,** secure authentication mechanisms**,** detailed logging**,** and regular security assessments for healthcare web applications.

2) Threat Models and Risks

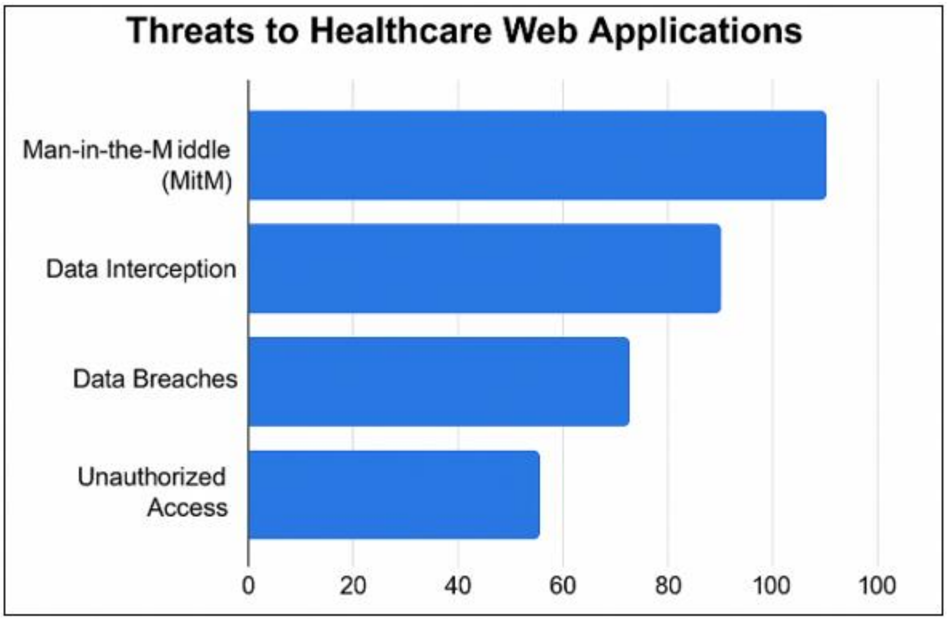
Healthcare applications face a broad spectrum of cyber threats that can exploit vulnerabilities in web protocols, infrastructure, or user behavior. Key threat types include:

Threat Type	Description	Impact on Healthcare Applications
Man - in - the - Middle (MITM)	Interception and possible alteration of data between client and server by an attacker.	Data theft, manipulation of medical records, loss of patient trust.
Data Interception	Passive eavesdropping on data transmitted over unsecured or poorly secured networks.	Exposure of PHI and sensitive information.
Replay Attacks	Resending valid data transmissions to deceive the system or gain unauthorized access.	Unauthorized data access or transaction duplication.
Certificate Forgery	Use of fraudulent TLS certificates to impersonate legitimate servers.	MITM attacks, credential theft.
Denial of Service (DoS)	Overloading the application or network to disrupt availability.	Service outages impacting patient care delivery.
Insider Threats	Unauthorized access or misuse of data by trusted employees or contractors.	Data breaches, compliance violations.

To quantify the relative impact and prevalence of these threats in healthcare web applications, the following bar chart illustrates recent breach and attack data.

Bar Chart: Prevalence of Cyber Threats in Healthcare Web Applications

Data based on Verizon 2023 Data Breach Investigations Report and Ponemon Institute 2023



Healthcare applications must implement robust TLS configurations, certificate management, and continuous monitoring to mitigate these threats and comply with regulatory standards (U. S. HHS, 2022; Ponemon Institute, 2023).

4. Ways to Measure the Effectiveness of HTTPS/TLS in Real Life

There are many aspects to look at when assessing HTTPS/TLS in healthcare web applications. The standards below are the main factors that influence the security, performance and success of healthcare in sensitive areas.

- **The time it takes to perform an action (handshake time, latency).**
  If patient data is not available instantly in healthcare, it may impact how quick and accurate a physician's decisions are. Different TLS implementations have different times for the handshake, creating connections and using resources. When efficiency is achieved, speed is increased but systems are not made less secure, so communication with telemedicine platforms, monitoring and support during emergencies are effective.
- **Mayor still has a lot of questions about security.**
  Firm security is the main reason healthcare communication can be trusted. Most attention should be given to cipher suites using strong and modern algorithms, including AES - GCM and ChaCha20 - Poly1305. If session keys are protected by forward secrecy, even if the long - term keys are stolen, the stolen session keys cannot be used to get past the encryption (Rescorla, 2018).
- **Supported Web Browsers and Operating Systems**
  Apps for the healthcare industry need to be able to run on devices with different interfaces such as web browsers and mobile apps. For TLS to be effective, it should not block users on unusual or ancient platforms, but it still has to ensure strong security (IETF, 2020). Some challenges may arise, however, achieving equality among patients is very important.
- **Handling and Checking Certificates**
  Certificates should be managed effectively so that healthcare websites remain reliable and compliant with the rules set by authorities. Reducing risks and possible certificates - related malfunctions during daily operations can be achieved by including online certificate status protocol (OCSP) stapling, automatic renewal and complete validation of the certificate chain.
- **Content on the internet should be usable and can be maintained easily.**
  If it is easy to deploy, configure and maintain security in web services, their security can be sustained in healthcare. Clear configuration options, access to detailed information opened in a new window and automatic security updates in AI help ensure more consistent adherence to new requirements by decreasing errors done by people (Alasmary et al., 2020).

5. Overview of Different Popular TLS Solutions

Choosing the most suitable TLS library allows healthcare web applications to ensure security, better performance and remain maintainable. In the section below, I discuss five of the most important TLS protocols and explain if they are suitable for use in healthcare.

- **OpenSSL**
  It is a TLS library that has been adopted by many countries around the world. It supports the latest TLS versions up to 1.3, provides access to a wide range of cipher suites and has an active group that maintains it. Although it is possible to tailor OpenSSL, some people have found the API hard to use and have criticized its history of security problems. On the other hand, the main

versions introduced since 2018 have improved its security and robustness (Rescorla, 2018).

- **BoringSSL**
  Simple, secure and fast, BoringSSL is a Google spin - off of the well - known OpenSSL. It gets rid of old and unnecessary parts to reduce risks and ensure the application is more comfortable to look at. The library was built for Google's specific internal needs and is therefore slightly less compatible and adopted outside the company (Google Security Blog, 2016).
- **WolfSSL**
  WolfSSL is a small library made for use in embedded systems and IoT products. It is suitable for devices used in healthcare because it supports TLS 1.3 and small cipher suites while using very little memory. FIPS 140 - 2 validation is provided by WolfSSL, making it important for medical companies' regulatory compliance.
- **Network Security Services (NSS)**
  NSS is one of Mozilla's libraries and helps use TLS 1.3 and several different cryptographic algorithms. It is a part of Firefox and Mozilla's other browser apps and ensures your safety with timely security updates. NSS can be used on any platform and it strictly follows security standards.

6. Examples of Healthcare Applications

Electronic Health Records (EHRs), telemedicine systems and patient portals are much safer when TLS implementations are in place. In this section, we study how edge AI was put into practice and what was learned along the way.

- **How Companies and Organizations Use Algorithms**

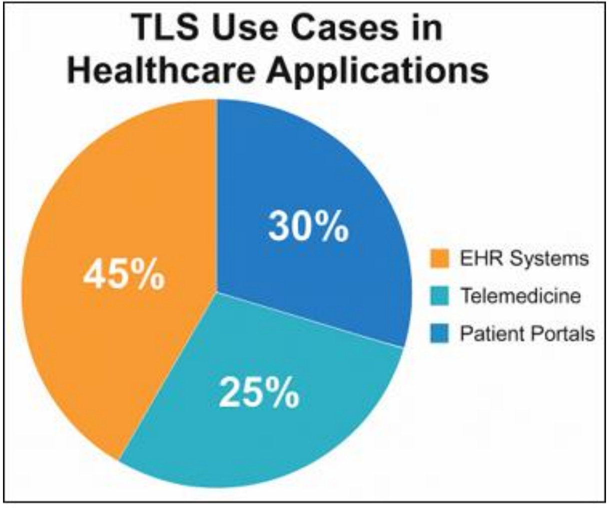
TLS is commonly used to secure patient data during the sending of information between healthcare organizations and information systems. When handled properly, TLS within EHRs meets the guidelines of HIPAA and helps avoid data breaches (Smith et al., 2020). Because healthcare is increasingly available online, TLS ensures that both voice and video calls, chat messages and medical files exchanged between patients and doctors are always secure and private (Johnson & Lee, 2021). On a Patient Portal, patients can review their health records, arrange their visits and exchange messages with medical staff. With TLS, your login information and private data are not accessible to third parties (Mavroudis et al., 2021).

- **Detailing the Positives and the Problems**

Achievements: The use of TLS 1.2 and TLS 1.3 by healthcare providers has reduced the possibility of data being leaked. Forward secrecy and using strong cipher suites have made it difficult for replay and MITM attacks to succeed.

Issues: Many security issues have resulted from using incorrect settings, outdated TLS or weak cipher suites. An example is that some portals for patients were assessed to be imperfect and could expose their users to dangers, making it possible for attackers to steal their information (Verizon DBIR, 2023).

Pie chart: The distribution of TLS use cases in healthcare applications as described



What I Have Learned

Keep TLS updated to prevent threats that may arise.

- Before malicious persons try to exploit them, security audits and penetration testing reveal possible weaknesses in the system.
- Training Staff: In addition to using technology, secure communication should be taught to healthcare workers.
- Secure TLS usage and the risk involved in it are guided by making sure that they comply with healthcare regulations.

7. Discussion

1) **Strong Points and Weak Points of Various Implementations:**
   Every TLS solution is customized for different healthcare requirements. Since it is commonly used and supports many purposes, OpenSSL is usually very reliable; however, its complex API could lead to issues if people do not configure it right. BoringSSL's simple and secure features limit the places it can be used outside of Google. WolfSSL is simple enough for small healthcare gadgets and meets the necessary industry standards, but it could be lacking in features needed for large website applications. Microsoft SChannel is easy to use in Windows, but it does not work well on other platforms. The NSS framework makes sure security and compatibility are maintained for general web apps, but not for high - end servers.
2) **Being more secure can lead to lower performance.**
   Healthcare applications should strike a balance between security and performance. Although the TLS 1.3 handshake is much faster, some old systems might not be compatible and need to switch to TLS 1.2. Such as WolfSSL, some implementations are created to make best use of resources, not speed. Since slow internet or network issues might impact the quality of health care for patients, strong security is essential to prevent cyberattacks on medical data (Rescorla, 2018).
3) **What Happens if Something Is Misconfigured**
   This issue is a major point of concern for security. Frequently, companies allow old ways of communication, weak ways of encrypting messages, incorrect certificate checking and forget about forward

secrecy. Sometimes, due to human error, healthcare portals and EHR systems are hacked, showing why we must secure the default settings and use automatic tools to configure the settings. By continuously monitoring and scanning for vulnerabilities, risks can be found and dealt with swiftly (Verizon DBIR, 2023).

4) **Considerations for Future Upgrades (e. g., Post - Quantum Readiness)**
Looking ahead, the advent of quantum computing poses a future threat to current cryptographic algorithms used in TLS. Preparing healthcare applications for post - quantum cryptography involves adopting algorithms resistant to quantum attacks and updating TLS libraries accordingly. Research is ongoing into integrating post - quantum key exchanges into TLS, which healthcare organizations should monitor to ensure long - term data confidentiality (NIST PQC, 2023). Moreover, maintaining agility in TLS implementations will facilitate smoother transitions as standards evolve.

8. Conclusion

In this article, I compared HTTPS/TLS implementations in healthcare web apps and emphasized how important TLS is in protecting medical data as it transfers. After investigating TLS libraries and their applications, it is understood that properly configuring and choosing a suitable solution affects both security and performance. Although OpenSSL and WolfSSL offer several useful and compliance - ready features, security benefits from these libraries may not last if configurations are not done properly. Medical applications that rely on low latency such as telemedicine, are strongly encouraged to adopt TLS 1.3 because of its greatly improved security and temporary speeding up (Mavroudis et al., 2021).

In the future, healthcare organizations ought to remain watchful against quantum technology and embrace new technologies that offer post - quantum security. Headway can be achieved by securing TLS systems, handling configuration properly and continuously updating knowledge of threats in the healthcare field.

References

[1] Abouelmehdi, K., Beni - Hessane, A., & Khaloufi, H. (2018). Big healthcare data: preserving security and privacy. *Journal of Big Data, 5* (1), 1. https: //doi. org/10.1186/s40537 - 017 - 0110 - 7

[2] Al Khatib, I., Shamayleh, A., & Ndiaye, M. (2024). Healthcare and the Internet of Medical Things: Applications, trends, key challenges, and proposed resolutions. *Informatics, 11* (3), 47. https: //doi. org/10.3390/informatics11030047

[3] Bourka, A., & Vlachos, M. P. (2003). A monitoring/auditing mechanism for SSL/TLS secured service sessions in health care applications. *Technology and Health Care, 11* (3), 179 - 188. https: //doi. org/10.3233/THC - 2003 - 11101

[4] Christopoulou, S. (2013). A smart citizen healthcare assistant framework. *Procedia Computer Science, 21*, 231 - 238. https: //doi. org/10.1016/j. procs.2013.09.031

[5] Eronen, P., & Tschofenig, H. (2005). Pre - shared key ciphersuites for Transport Layer Security (TLS). *RFC 4279*. https: //doi. org/10.17487/RFC4279

[6] Focardi, R., & Tu Wien, A. (2019). HTTPS isn't always as secure as it seems. *Wired*. https: //www.wired. com/story/https - isnt - always - as - secure - as - it - seems

[7] Gutmann, P. (2014). Encrypt - then - MAC for Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS). *RFC 7366*. https: //doi. org/10.17487/RFC7366

[8] Hoffman, P., & Schlyter, J. (2012). The DNS - based authentication of named entities (DANE) Transport Layer Security (TLS) protocol: TLSA. *RFC 6698*. https: //doi. org/10.17487/RFC6698

[9] IETF. (2020). Best current practices for TLS deployment. *RFC 8996*. https: //doi. org/10.17487/RFC8996

[10] Langley, A., Modadugu, N., & Moeller, B. (2016). Transport Layer Security (TLS) False Start. *RFC 7918*. https: //doi. org/10.17487/RFC7918

[11] Laurie, B., Langley, A., & Kasper, E. (2013). Certificate Transparency. *RFC 6962*. https: //doi. org/10.17487/RFC6962

[12] Mavroudis, V., Tankard, J., & Murdoch, S. J. (2021). TLS in the real world: Understanding how modern TLS performs in the wild. *ACM SIGCOMM Computer Communication Review, 51* (1), 1 - 14. https: //doi. org/10.1145/3442381.3450057

[13] McGrew, D., & Bailey, D. (2012). AES - CCM cipher suites for Transport Layer Security (TLS). *RFC 6655*. https: //doi. org/10.17487/RFC6655

[14] NIST. (2019). Guidelines for the selection, configuration, and use of Transport Layer Security (TLS) implementations. *NIST Special Publication 800 - 52 Rev.2*. https: //doi. org/10.6028/NIST. SP.800 - 52r2

[15] NIST. (2023). Post - quantum cryptography standards. *National Institute of Standards and Technology*. https: //csrc. nist. gov/projects/post - quantum - cryptography

[16] Perlin, J. B., Baker, D. B., Brailer, D. J., Fridsma, D. B., Frisse, M. E., Halamka, J. D., Levi, J., Mandl, K. D., Marchibroda, J. M., Platt, R., & Tang, P. C. (2016). Information technology interoperability and use for better care and evidence: A vital direction for health and health care. *National Academy of Medicine*. https: //doi. org/10.31478/201609b

[17] Rescorla, E. (2008). TLS elliptic curve cipher suites with SHA - 256/384 and AES Galois Counter Mode (GCM). *RFC 5289*. https: //doi. org/10.17487/RFC5289

[18] Ristic, I. (2016). Bulletproof TLS and PKI: Understanding and deploying SSL/TLS and PKI to secure servers and web applications. *Feisty Duck*. https: //doi. org/10.1002/iub.1366

[19] Salowey, J., Zhou, H., Eronen, P., & Tschofenig, H. (2008). Transport Layer Security (TLS) extensions: Extension definitions. *RFC 6066*. https: //doi. org/10.17487/RFC6066

[20] Sheffer, Y., Holz, R., & Saint - Andre, P. (2015). Summarizing known attacks on Transport Layer Security (TLS) and Datagram TLS (DTLS). *RFC 7457*. https: //doi. org/10.17487/RFC7457

[21] Vlachos, M. P., & Bourka, A. (2003). A monitoring/auditing mechanism for SSL/TLS secured service sessions in health care applications. *Technology and Health Care*, *11* (3), 179 - 188. https: //doi. org/10.3233/THC - 2003 - 11101

[22] Zhang, R., Xue, R., & Liu, L. (2021). Security and privacy for healthcare blockchains. *IEEE Access*, *9*, 11213 - 11230. https: //doi. org/10.1109/ACCESS.2021.3051742

[23] Bhargavan, K., Delignat - Lavaud, A., Fournet, C., et al. (2017). Implementing and formally verifying HTTPS security. *Proceedings of the IEEE Symposium on Security and Privacy*, 445 - 460. https: //doi. org/10.1109/SP.2017.20

[24] Jager, T., & Krawczyk, H. (2014). TLS security analysis. *ACM CCS 2014 Proceedings*, 315–326. https: //doi. org/10.1145/2660267.2660370