International Journal of Applied Information Systems and Cybersecurity
Vol. 1 No. 1 (2026) pp. 19-35
Online ISSN: XXXX-XXXX

Original Research

Evaluation of SSL/TLS Configuration in Academic Information Systems: A Case Study

Murni Marisma[1], Ermadi Satriya Wijaya[1]✉, Mukhlis Prasetyo Aji[1], Agung Purwo Wicaksono[1]
[1] Department of Informatics Engineering, Universitas Muhammadiyah Purwokerto, Indonesia
✉ ermadi_satriya@yahoo.com
10.65840/ijaisc.vxix.xx

Article Info
Submitted:
09/21/2025
Revised:
11/10/2025
Accepted:
12/30/2025
Available Online:
01/15/2026

Abstract. *The security of web-based academic information systems has become increasingly critical as sensitive data is continuously exchanged over public networks. One of the primary mechanisms for securing such communication is the implementation of Secure Sockets Layer (SSL) and Transport Layer Security (TLS). However, the effectiveness of these protocols is highly dependent on proper configuration rather than mere adoption. Misconfigurations can introduce significant vulnerabilities, even in systems that utilize modern encryption standards. This study evaluates the SSL/TLS implementation of an academic information system using a configuration-based security assessment approach. The analysis focuses on key parameters, including protocol support, certificate validity, key exchange strength, cipher strength, and forward secrecy. Data is obtained through an external assessment using the Qualys SSL Labs platform and is analyzed descriptively and comparatively against established security best practices. The results indicate that the system has implemented TLS 1.3 and employs strong cryptographic algorithms, achieving an overall security rating of B. Despite these strengths, the absence of consistent forward secrecy and suboptimal key exchange configurations limit the system's overall security posture. These findings highlight that while the system provides adequate baseline protection, it remains vulnerable to advanced threats such as retrospective decryption and downgrade attacks.*

Keywords: *SSL/TLS; web security; configuration analysis; academic information systems; cybersecurity*

1. Introduction

The rapid expansion of web-based academic information systems has fundamentally transformed how educational institutions manage and deliver services. Platforms that handle student records, course registration, financial transactions, and authentication processes are now almost entirely dependent on web technologies. This shift, while improving accessibility and efficiency, has also introduced a broader attack surface. Sensitive data is continuously transmitted across networks, making secure communication not just desirable, but essential (Rescorla, 2018).

At the core of secure web communication lies the Secure Sockets Layer (SSL) and its successor, Transport Layer Security (TLS). These protocols are designed to provide confidentiality, integrity, and authenticity through encryption mechanisms that protect data in transit. Over time, TLS has evolved significantly, with modern versions such as TLS 1.3 addressing many of the vulnerabilities present in earlier iterations. It removes outdated cryptographic primitives, enforces stronger security defaults, and reduces handshake complexity, thereby improving both security and performance (Rescorla, 2018).

Yet, the mere adoption of TLS does not guarantee a secure system. In practice, many web applications implement SSL/TLS (TLS) in ways that are incomplete, inconsistent, or misconfigured. These misconfigurations—ranging from weak cipher suites and improper certificate chains to lack of forward secrecy—can introduce vulnerabilities that undermine the intended security guarantees. In fact, recent studies have shown that configuration errors remain one of the most common causes of security weaknesses in HTTPS-enabled systems, even when modern protocols are in use (Zhang et al., 2022).

The risks associated with improper SSL/TLS configuration are not trivial. Attack vectors such as man-in-the-middle (MITM) attacks, downgrade attacks, and session decryption can exploit weaknesses in protocol negotiation or key management. In environments where sensitive data is exchanged regularly, such as academic systems, these vulnerabilities may lead to unauthorized access, data leakage, or long-term compromise of user privacy (Sherry et al., 2015). What makes this particularly concerning is that such weaknesses often remain undetected, as systems may appear secure on the surface—especially when HTTPS is enabled—while still harboring latent configuration flaws.

Despite the growing body of research on web security, much of the focus has traditionally been placed on application-layer vulnerabilities, such as injection attacks or authentication flaws. Comparatively less attention has been given to SSL/TLS configuration as a standalone area of evaluation, particularly within the context of academic information systems. This gap is significant. Academic platforms often operate under diverse constraints, including legacy infrastructure, varying administrative practices, and limited security resources, all of which can contribute to inconsistent or suboptimal configurations.

This study addresses that gap by focusing specifically on the evaluation of SSL/TLS implementation in an academic information system. Rather than conducting intrusive penetration testing, the research adopts a configuration-oriented approach, examining how cryptographic mechanisms are deployed and whether they align with current security best practices. The analysis considers key aspects such as protocol support, certificate validity, key exchange mechanisms, cipher strength, and forward secrecy.

By isolating SSL/TLS as the primary subject of analysis, this study aims to provide a more precise understanding of the system's security posture. The goal is not only to identify weaknesses, but also to interpret their implications in a broader security context. In doing so, the research contributes to a growing recognition that

secure communication is not solely about enabling encryption, but about implementing it correctly and consistently.

This study contributes to the existing literature in three key aspects. First, it provides a focused evaluation of SSL/TLS configuration as a standalone security layer, which is often overlooked in broader vulnerability assessments. Second, it applies a configuration-based analytical approach to identify subtle security gaps that may not be detected through traditional testing frameworks. Third, it offers practical insights into improving SSL/TLS implementation in academic information systems, particularly in environments with constrained resources. To provide contextual understanding, the overall architecture of the target system and its SSL/TLS-secured communication flow are illustrated in Figure 1.

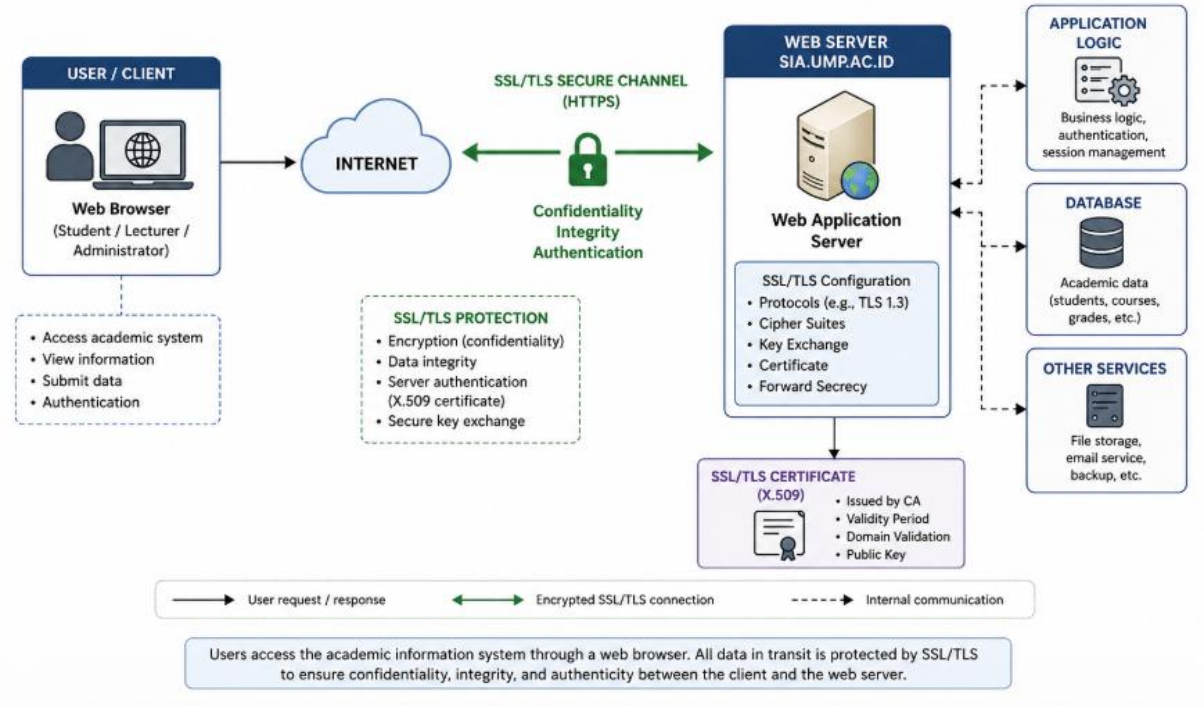

Figure 1. Overview of the target academic information system and SSL/TLS communication flow

Security assessment of web-based systems has long relied on structured methodologies designed to uncover vulnerabilities in a systematic and repeatable manner. Among the earlier frameworks, the Information Systems Security Assessment Framework (ISSAF) has been widely recognized for its comprehensive approach, covering stages from information gathering to exploitation and post-assessment analysis. ISSAF emphasizes technical depth, particularly in penetration testing scenarios, where the objective is to simulate real-world attacks in order to identify exploitable weaknesses (Open Information Systems Security Group, 2018). While effective in exposing vulnerabilities, its focus is largely operational—centered on how systems can be broken—rather than on how securely they are configured at a cryptographic level.

In parallel, more contemporary practices have been shaped by the Open Web Application Security Project (OWASP), which provides widely adopted guidelines

such as the OWASP Testing Guide and the OWASP Top 10. Unlike ISSAF, OWASP adopts a broader and more flexible perspective, combining both technical testing and secure development principles. It highlights common vulnerabilities such as injection flaws, broken authentication, and security misconfigurations, offering a practical reference for both developers and security analysts (OWASP Foundation, 2021). However, similar to ISSAF, OWASP frameworks tend to treat SSL/TLS as one component within a larger security landscape, rather than as a focal point of analysis.

This distinction is important. Much of the existing literature has approached web security from a holistic vulnerability perspective, where SSL/TLS is assessed alongside application logic flaws and network-level weaknesses. As a result, dedicated analysis of SSL/TLS configuration—particularly as an independent research focus—remains relatively limited. Yet, recent studies suggest that configuration-level weaknesses in TLS deployments are among the most prevalent and impactful security issues in modern web systems.

For instance, large-scale measurement studies have demonstrated that a significant portion of web servers, including those supporting HTTPS, suffer from misconfigurations such as weak cipher selection, incomplete certificate chains, or improper protocol negotiation policies (Hu et al., 2021). These issues often persist even in systems that have adopted the latest TLS versions, highlighting a gap between protocol availability and correct implementation. Similarly, research by (Hu et al., 2021) shows that inconsistencies in TLS configuration across different server environments can lead to unpredictable security behavior, especially in systems that rely on distributed infrastructure.

Another line of work has focused on the real-world deployment challenges of TLS, particularly in institutional and enterprise environments. Studies indicate that administrators frequently face trade-offs between compatibility and security, which can result in partial adoption of best practices such as forward secrecy or strict transport security (Acar et al., 2016). These trade-offs are often driven by the need to support legacy clients or maintain system availability, but they inadvertently introduce exploitable conditions.

More recent contributions have also explored the evolution of TLS ecosystems, emphasizing how rapid adoption of newer standards such as TLS 1.3 does not automatically translate into stronger security. Instead, the effectiveness of these protocols depends heavily on configuration choices and policy enforcement (Holz et al., 2020). In many cases, systems that technically support modern protocols still exhibit weaknesses due to inconsistent configuration or incomplete feature utilization.

Despite these advancements, there remains a noticeable gap in studies that specifically examine SSL/TLS implementation within academic information systems. Such systems are unique in that they manage sensitive personal and institutional data while often operating under resource and policy constraints. The combination of high data sensitivity and variable security maturity makes them particularly relevant for focused SSL/TLS evaluation.

In light of this, the present study positions itself at the intersection of configuration analysis and applied security evaluation. Rather than adopting a full penetration testing framework, it isolates SSL/TLS configuration as the primary

subject of analysis, aiming to provide a more granular understanding of how cryptographic mechanisms are implemented in practice. This approach complements existing frameworks such as ISSAF and OWASP by addressing a layer of security that is often assumed to be correctly implemented, yet frequently is not.

2. Method

2.1. Research Approach

This study adopts a qualitative security assessment approach that focuses on analyzing SSL/TLS configuration from an external perspective. The approach emphasizes configuration-level evaluation rather than active penetration testing, which aims to identify misconfigurations and weaknesses that could be exploited by attackers. Configuration analysis has been widely recognized as an effective method for discovering security gaps in web services because many real-world breaches are caused by insecure or inconsistent configuration rather than flaws in the underlying protocol itself (Altulaihan et al., 2023).

2.2. Data Source and Assessment Tool

The primary data source is obtained from an external evaluation using Qualys SSL Labs (ssllabs.com), a trusted and widely used platform for analyzing SSL/TLS configurations of web servers. SSL Labs performs a series of automated tests that examine how the server handles TLS connections, negotiates cryptographic parameters, and presents its certificate chain to clients (Holz et al., 2016).

The assessment is complemented by observational analysis of the system's behavior in establishing secure connections under different client scenarios and configuration conditions.

2.3. Evaluation Parameters

The SSL/TLS configuration is evaluated based on five key parameters that reflect the security quality of the implementation. These parameters align with widely accepted evaluation metrics in recent studies and guidelines(Holz et al., 2020; Rescorla, 2018) *Table 1* summarizes the parameters and their evaluation focus.

Table 1. SSL/TLS Evaluation Parameters

Parameter	Description
Protocol Support	Evaluates the highest protocol version and the availability of secure protocols (e.g., TLS 1.3). Higher versions provide stronger security guarantees (Rescorla, 2018).
Certificate Validity	Assesses the validity, trust chain, and proper configuration of the server certificate to ensure identity authenticity (ENISA, 2021).
Key Exchange Strength	Examines the key exchange mechanisms used by the server, including the use of ephemeral key exchange (ECDHE/DHE) to enhance session security (Beurdouche et al., 2017).
Cipher Strength	Measures the cryptographic algorithms and key lengths used for encryption. Strong cipher suites provide resistance against brute-force and cryptanalytic attacks (Rescorla, 2018).
Forward Secrecy	Determines whether session keys remain secure even if the server's long-term private key is compromised, ensuring protection of past communications (Bhargavan et al., 2019).

2.4. Research Procedure

The research procedure consists of several systematic steps, as illustrated in *Figure 2*.

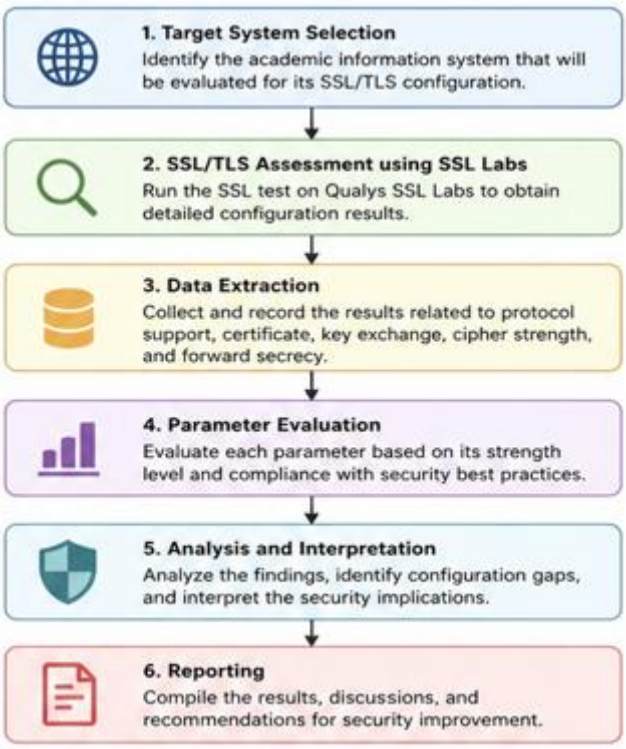

Figure 2. Research methodology for SSL/TLS configuration analysis

2.5. Data Analysis

The data obtained from SSL Labs is analyzed descriptively and comparatively. Each parameter is interpreted against industry best practices and recommendations from standards organizations to determine the security maturity of the system (Holz et al., 2020; Rescorla, 2018). The analysis focuses on identifying configuration weaknesses and their potential impact on confidentiality, integrity, and user privacy.

3. Results and Discussion

3.1. SSL Configuration Result

The SSL/TLS configuration of the target academic system was evaluated using a standardized external assessment tool, focusing on protocol support, certificate validity, key exchange mechanisms, and cipher strength. The analysis reveals that the system achieves an overall rating of B, indicating a moderate level of security maturity. While this rating reflects the presence of fundamental protections, it simultaneously signals that certain advanced security mechanisms are either partially implemented or absent.

A structured summary of the configuration findings is presented in *Table 2*, which consolidates the key technical parameters derived from the SSL evaluation process.

Table 2. SSL/TLS Configuration Summary of the Target System

Parameter	Result
Overall Rating	B
Protocol Support	TLS 1.3
Certificate Status	Valid
Cipher Strength	Strong
Key Exchange	Moderate
Forward Secrecy	Partially Supported

As shown in *Table 2*, the system has successfully implemented TLS 1.3, the latest version of the Transport Layer Security protocol. This is a non-trivial strength. TLS 1.3 introduces several security improvements over its predecessors, including reduced handshake complexity and the elimination of obsolete cryptographic algorithms, thereby minimizing the attack surface associated with legacy configurations (Rescorla, 2018). More importantly, it enhances resistance against downgrade attacks and significantly improves performance without compromising security.

However, the presence of TLS 1.3 alone does not guarantee optimal protection. A closer inspection reveals that forward secrecy is not consistently supported across all access scenarios, particularly when accessed through certain clients or browser configurations. This limitation is critical. Forward secrecy ensures that even if long-term private keys are compromised, past communication sessions remain secure. Without it, encrypted traffic could potentially be decrypted retrospectively, exposing sensitive academic data such as authentication credentials or student records (Bhargavan et al., 2014).

The system also demonstrates strong cipher strength, indicating the use of robust encryption algorithms that are resistant to brute-force and cryptanalytic attacks. This aligns with current best practices recommended by international security standards, which emphasize the importance of secure cipher suites in protecting data-in-transit (IETF, 2018). Still, cipher strength alone cannot compensate for weaknesses in key exchange or session management.

Interestingly, the key exchange mechanism is categorized as moderate, suggesting that while secure methods are in place, they may not fully leverage ephemeral key exchange techniques in all cases. This again ties back to the incomplete implementation of forward secrecy. In modern secure systems, ephemeral Diffie-Hellman (DHE or ECDHE) is typically preferred to ensure that session keys are not derived from static parameters (Beurdouche et al., 2017). The absence or inconsistent use of such mechanisms reduces resilience against advanced threat models.

From a broader perspective, the observed configuration reflects a system that is functionally secure but not optimally hardened. This distinction matters. In real-world deployments, attackers rarely exploit obvious vulnerabilities; instead, they target subtle misconfigurations, protocol fallbacks, or inconsistencies in implementation. Several recent studies have shown that even systems with relatively high SSL ratings can remain susceptible to exploitation if advanced features such as forward secrecy or strict transport security are not uniformly enforced (Holz et al., 2020).

Another point worth noting is the implicit indication that the system may be operating behind a proxy or content delivery layer, as suggested by the abstraction of its original network identity. While this adds a layer of protection by masking backend infrastructure, it also introduces dependencies on external configuration policies. Misalignment between proxy-level and server-level SSL configurations is a known source of security gaps (Durumeric et al., 2017).

In practical terms, the implications are clear. The current SSL/TLS implementation provides adequate baseline protection, sufficient to prevent common attacks such as passive eavesdropping or basic interception. Yet, it stops short of delivering the level of assurance expected in high-security environments, particularly those handling sensitive academic data. Incremental improvements—specifically enforcing forward secrecy across all sessions, optimizing key exchange mechanisms, and aligning configurations with modern security benchmarks—would likely elevate the system to an "A" rating and significantly strengthen its defensive posture.

3.2. SSL/TLS Configuration Visualization and Interpretation

The visual representation of the SSL/TLS assessment provides a concise yet revealing overview of the system's security posture. Unlike tabular data, which isolates parameters, the graphical summary captures proportional relationships between critical components—certificate reliability, protocol support, key exchange robustness, and cipher strength—allowing a more intuitive interpretation of strengths and weaknesses within the configuration.

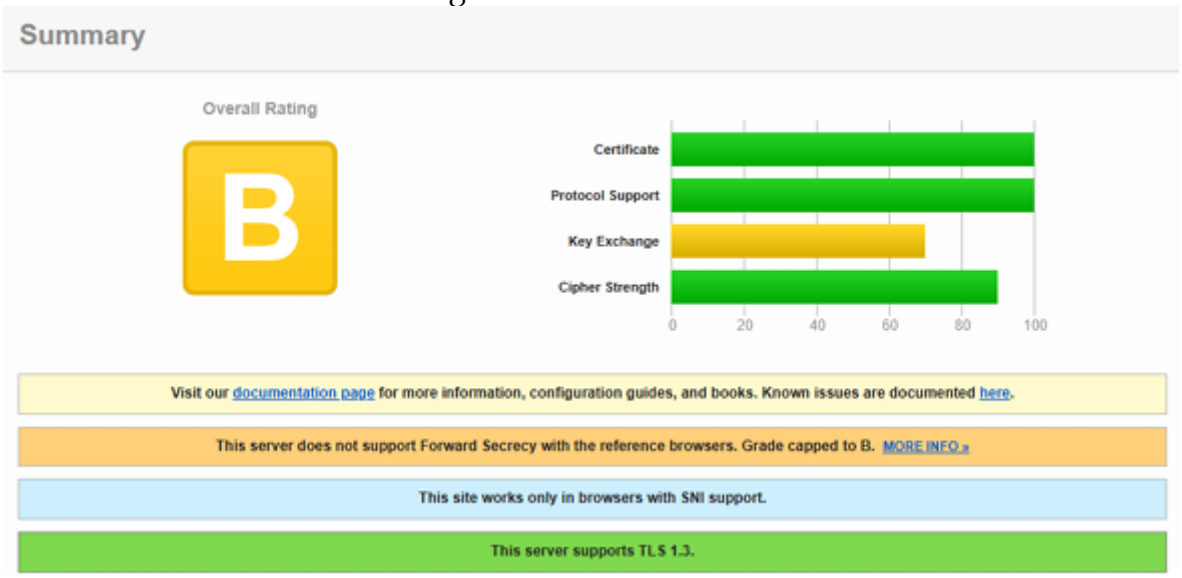

Figure 3. SSL/TLS configuration summary showing overall rating and security components

As illustrated in *Figure 3*, the system demonstrates strong performance in certificate validity and protocol support, both reaching near-maximum levels. This suggests that the digital certificate is properly issued, trusted, and correctly configured, which is essential for establishing secure communication channels. Proper certificate management remains a cornerstone of web security, as misconfigured or expired certificates can directly undermine trust and expose users to man-in-the-middle attacks (Clark & van Oorschot, 2013).

The visualization also highlights that TLS 1.3 is fully supported, which significantly strengthens the system's cryptographic foundation. TLS 1.3 not only removes insecure legacy algorithms but also enforces more secure handshake mechanisms by default. In practice, this reduces latency and limits opportunities for attackers to exploit protocol negotiation weaknesses. Recent empirical analyses have shown that widespread adoption of TLS 1.3 correlates with a measurable decrease in exploitable downgrade vectors and handshake manipulation attacks (Holz et al., 2020; Rescorla, 2018).

However, the chart subtly exposes an imbalance. While certificate and protocol metrics appear optimal, the key exchange component lags behind, positioned in a mid-range category rather than at full strength. This disparity is not merely cosmetic. Key exchange mechanisms play a pivotal role in ensuring that session keys remain ephemeral and independent from long-term secrets. When these mechanisms are only partially optimized, the system becomes more vulnerable to retrospective decryption scenarios, particularly in adversarial environments where traffic interception is feasible (Bhargavan et al., 2014; Rescorla, 2018).

Another notable aspect is the strong cipher strength, which indicates that encryption algorithms in use are sufficiently robust against contemporary cryptanalytic attacks. This is reassuring, yet somewhat deceptive if viewed in isolation. Encryption strength must be considered alongside key management and protocol enforcement. A system can employ strong ciphers and still be vulnerable if session key generation or exchange processes are not equally secure. This layered dependency has been emphasized in recent security studies, which argue that modern web security failures are increasingly driven by configuration inconsistencies rather than algorithmic weaknesses (Holz et al., 2020; Rescorla, 2018).

The most critical insight, however, lies beneath the visual summary: the system's inability to fully support forward secrecy, as indicated by the grading limitation. Although not directly visualized in the bar chart, this constraint explains why the overall rating is capped at "B." Forward secrecy ensures that each session uses a unique encryption key, preventing attackers from decrypting past communications even if long-term keys are later compromised. Its absence—or inconsistent implementation—introduces a latent risk that is often underestimated in operational environments (Alashwali et al., 2020).

It is also worth reflecting on the broader implication of achieving a "B" rating. In many operational contexts, this level may be considered acceptable, particularly for systems that are not classified as high-risk. Yet, from a research and best-practice standpoint, it represents a transitional state rather than a final benchmark. Contemporary guidelines increasingly advocate for configurations that consistently achieve "A" or higher, especially in systems handling sensitive or personally identifiable information (Rescorla, 2018).

Taken together, the visualization reinforces a nuanced conclusion. The system is not insecure—it successfully implements modern protocols and strong encryption—but it is not fully optimized either. The gap between "adequate" and "robust" security is often defined by subtle configuration details, and in this case,

forward secrecy and key exchange optimization emerge as the primary areas requiring attention.

3.3. Security Analysis and Interpretation

The findings obtained from the SSL/TLS assessment point toward a system that is technically sound, yet not fully optimized from a security engineering perspective. An overall rating of B positions the system in what could be described as a "functional but transitional" security state—adequate for general protection, but not sufficiently hardened against more sophisticated threat models.

At first glance, the implementation of TLS 1.3 stands out as a clear strength. This protocol version has been widely recognized for its streamlined handshake process, mandatory use of secure cryptographic primitives, and built-in resistance against downgrade attacks. In practical terms, it eliminates entire classes of vulnerabilities associated with earlier TLS versions, such as insecure renegotiation and weak cipher negotiation (Rescorla, 2018). More recent analyses further confirm that TLS 1.3 significantly reduces the attack surface in real-world deployments, particularly in environments exposed to passive and active network adversaries.

Yet, security is rarely determined by protocol choice alone. What emerges more subtly—but critically—is the incomplete support for forward secrecy, which ultimately constrains the system's overall rating. This is not a minor technical detail. Forward secrecy ensures that session keys are ephemeral and independent of long-term private keys, meaning that even if a server's private key is compromised in the future, previously recorded communications remain secure. Without this property, encrypted traffic may become retrospectively vulnerable, especially in scenarios involving long-term data collection by adversaries (Albrecht et al., n.d.).

The implications become more tangible when considering realistic attack scenarios. In academic systems, where authentication sessions and personal data exchanges are frequent, the absence of consistent forward secrecy increases exposure to session replay and decryption risks. Attackers who obtain server-side secrets—whether through misconfiguration, insider threats, or unrelated vulnerabilities—could potentially decrypt historical traffic. While such attacks require significant capability, they are no longer purely theoretical, particularly in high-value or institutionally targeted environments (Amann et al., 2013).

Another dimension worth emphasizing is the moderate classification of key exchange mechanisms. This suggests that while secure methods are in place, they may not consistently enforce ephemeral key exchange (e.g., ECDHE) across all connections. The distinction matters. Modern security guidelines strongly recommend ephemeral key exchange as a baseline requirement, not an optional enhancement. When this is only partially implemented, it introduces inconsistency—arguably one of the most common and dangerous characteristics of real-world security failures (Bhargavan et al., 2014; Rescorla, 2018)

Interestingly, the system's strong cipher configuration does provide a degree of compensatory protection. Robust encryption algorithms—when correctly implemented—remain highly resistant to brute-force and cryptanalytic attacks. However, this strength should not be overinterpreted. Encryption is only one layer in a multi-layered defense model. If key exchange and session management are not

equally robust, the overall security posture remains constrained. As highlighted in recent large-scale TLS ecosystem studies, misalignment between cryptographic strength and configuration practices is a recurring issue across web infrastructures (Sheffer et al., 2022).

There is also a broader architectural implication hinted at by the findings. The difficulty in identifying the system's original network endpoint suggests the use of an intermediary layer, such as a reverse proxy or content delivery network (CDN). While such mechanisms enhance resilience and obscure backend infrastructure, they can also introduce configuration fragmentation. SSL/TLS policies may be applied at multiple layers—edge, proxy, and origin server—leading to inconsistencies that are not immediately visible in surface-level scans (Calzavara et al., 2018). This layered complexity often explains why systems with modern protocols still fall short of optimal security ratings.

From a practical standpoint, the system demonstrates baseline compliance with contemporary security expectations, but not alignment with best-in-class configurations. This distinction is subtle but important. A "B" rating does not indicate vulnerability in the conventional sense; rather, it reflects missed opportunities for hardening. Enforcing forward secrecy across all connections, standardizing ephemeral key exchange, and ensuring consistent policy enforcement across all access paths would likely elevate the system to an "A" rating with relatively minimal architectural changes.

In essence, the analysis reveals a system that has adopted the right technologies but has not fully leveraged their potential. The gap is no longer about capability—it is about configuration precision. And in modern cybersecurity practice, that gap often defines the difference between being secure and being resilient.

3.4. Security Implications

The security implications of the observed SSL/TLS configuration extend beyond a simple grading outcome. They touch directly on how resilient the system is when confronted with realistic, and often evolving, threat scenarios. While the implementation demonstrates a baseline level of protection, several configuration gaps—most notably the incomplete enforcement of forward secrecy—introduce risks that warrant closer examination.

One of the most immediate concerns relates to the absence of consistent forward secrecy. In cryptographic terms, forward secrecy ensures that session keys are ephemeral and cannot be reconstructed even if long-term private keys are later exposed. Without this property, encrypted communications lose an important layer of temporal protection. If an attacker were to obtain the server's private key—through compromise, leakage, or even future cryptanalytic advances—previously recorded traffic could potentially be decrypted. This is not merely a theoretical risk. Modern adversaries, including state-level actors, are known to adopt "store now, decrypt later" strategies, collecting encrypted data with the expectation that it may become readable in the future (Rescorla, 2018; Sheffer et al., 2022).

In the context of an academic information system, the implications are particularly sensitive. Such platforms routinely handle authentication credentials, personal identification data, and academic records. The retrospective exposure of this

data could lead to identity misuse, unauthorized access, or reputational harm to both users and institutions. Even if the probability of key compromise is relatively low, the potential impact remains significant. This asymmetry—low likelihood, high consequence—is precisely why forward secrecy has become a recommended default in modern TLS deployments (Rescorla, 2018; Sheffer et al., 2022).

A second concern lies in the system's potential exposure to downgrade attacks. Although TLS 1.3 is implemented, incomplete configuration or compatibility fallbacks may allow attackers to force connections into weaker cryptographic states under certain conditions. Downgrade attacks exploit negotiation processes between client and server, tricking them into agreeing on less secure protocol versions or cipher suites. While TLS 1.3 was designed to mitigate many of these issues, improper backward compatibility handling can still reintroduce vulnerabilities (De Ruiter & Poll, n.d.).

This risk becomes more pronounced in heterogeneous environments where users access the system through a variety of browsers and devices. Not all clients enforce strict security policies, and some may still support legacy negotiation paths. If the server does not explicitly restrict these pathways, it effectively broadens the attack surface. Empirical studies have shown that downgrade vulnerabilities often persist not because of outdated protocols themselves, but due to permissive configuration practices that prioritize compatibility over security (Somorovsky, 2016).

Beyond technical vulnerabilities, there is a broader and often underappreciated dimension: user privacy. Secure communication protocols are not only about protecting data integrity and confidentiality; they also serve as a trust mechanism between users and digital services. When SSL/TLS configurations fall short of best practices, even subtly, they can undermine this trust. Users typically assume that HTTPS guarantees strong protection, yet this assumption does not always hold in the presence of partial implementations or misconfigurations.

The lack of full forward secrecy, combined with moderate key exchange robustness, suggests that session confidentiality may not be uniformly guaranteed. In practical terms, this means that user activities—login sessions, data submissions, or browsing patterns—could be more exposed than expected under certain adversarial conditions. Privacy, in this sense, is not binary. It exists on a spectrum, and the current configuration places the system somewhere in the middle rather than at the higher end of that spectrum (Holz et al., 2020; Rescorla, 2018; Sheffer et al., 2022).

It is also worth considering the cumulative effect of these issues. Individually, each limitation—whether related to forward secrecy, downgrade resistance, or configuration inconsistency—may appear manageable. Collectively, however, they create a layered vulnerability profile. Attackers rarely rely on a single weakness; instead, they exploit chains of small misconfigurations that, when combined, yield significant access or insight into protected systems. This compositional nature of security risk has been repeatedly observed in recent analyses of web infrastructure vulnerabilities (Altulaihan et al., 2023; Sheffer et al., 2022).

From a mitigation standpoint, the path forward is relatively clear. Enforcing forward secrecy across all connections, disabling legacy protocol fallback mechanisms, and aligning SSL/TLS configurations with contemporary best practices

would substantially reduce the identified risks. Importantly, these improvements do not require fundamental architectural changes. They are, in essence, refinements—yet their impact on security posture would be disproportionately large.

In summary, the system's current SSL/TLS implementation provides a reasonable level of protection against common threats, but it leaves room for exploitation under more advanced scenarios. Addressing these gaps is not simply a matter of improving a rating; it is about ensuring that the confidentiality and privacy of user data remain intact, even under adverse conditions.

3.5. Comparative Discussion with Security Standards

Placing the observed configuration in a broader context helps clarify what the "B" rating actually means in practice. On paper, the system aligns with several contemporary expectations—most notably the adoption of TLS 1.3 and the use of strong cipher suites. Yet, when measured against widely accepted security baselines, it becomes evident that compliance is partial rather than complete.

Table 3. Comparison with Recommended SSL/TLS Security Standards

Parameter	Observed	Recommended
TLS Version	TLS 1.3	TLS 1.3 ✓
Cipher	Strong	Strong ✓
Key Exchange	Moderate	Strong (ECDHE)
Forward Secrecy	Partial	Full

The comparison presented in *Table 3* highlights a clear gap between the observed configuration and recommended best practices. While the system successfully adopts modern protocols and strong encryption, it falls short in ensuring consistent forward secrecy and optimal key exchange mechanisms.

Current best-practice guidelines emphasize consistency as much as capability. For instance, modern server-side recommendations advocate not only for the use of secure protocols, but also for the strict enforcement of forward secrecy, elimination of legacy fallback paths, and careful curation of cipher suites. These requirements are not arbitrary. They reflect a shift in threat modeling, where attackers increasingly exploit configuration gaps rather than cryptographic weaknesses themselves (Rescorla, 2018; Sheffer et al., 2022).

From this perspective, the system under evaluation occupies an interesting middle ground. It demonstrates technological alignment with modern standards, yet falls short in configuration completeness. The presence of TLS 1.3 suggests that the system has moved beyond outdated protocol dependencies. However, the inconsistent implementation of forward secrecy indicates that security policies are not uniformly enforced across all communication scenarios. This inconsistency is precisely what prevents the system from achieving a higher security classification.

Comparatively, systems that achieve an "A" rating typically exhibit tight configuration discipline. This includes mandatory use of ephemeral key exchange (e.g., ECDHE), strict disabling of weak negotiation mechanisms, and full support for forward secrecy across all client interactions. In contrast, a "B" rating often reflects environments where secure features are present but not consistently applied. Such

configurations are not inherently insecure, but they introduce predictable weaknesses that can be leveraged under specific conditions (Rescorla, 2018; Sheffer et al., 2022).

It is also important to recognize that security standards themselves are evolving. What was considered a strong configuration five years ago may now be viewed as insufficient. For example, earlier TLS deployments often prioritized compatibility with older clients, allowing fallback to weaker protocols or non-ephemeral key exchanges. Today, this approach is increasingly discouraged. The balance has shifted toward security-first configurations, even at the cost of reduced backward compatibility (Vanhoef & Ronen, 2020).

Another dimension of comparison lies in operational practice. In real-world deployments, SSL/TLS configurations are rarely static. They are influenced by infrastructure layers, including reverse proxies, load balancers, and content delivery networks. This layered architecture can lead to policy fragmentation, where different components enforce slightly different security rules. As a result, the system may appear secure under certain testing conditions while exposing weaker configurations under others. Such inconsistencies have been identified as a recurring issue in large-scale measurements of web security posture (Holz et al., 2020; Sheffer et al., 2022).

Against this backdrop, the evaluated system reflects a common pattern observed in institutional web applications. It adopts modern protocols and achieves a reasonable level of protection, yet stops short of full optimization. This is not necessarily due to a lack of awareness, but often a result of competing priorities—usability, compatibility, and operational constraints. Nonetheless, from a security standpoint, these trade-offs introduce measurable risk.

A more stringent alignment with established standards would require relatively targeted adjustments. Enforcing forward secrecy across all sessions, standardizing key exchange mechanisms, and eliminating any residual fallback behavior would likely elevate the system's rating. More importantly, these changes would enhance resilience against both current and emerging attack vectors.

Ultimately, the comparison highlights a subtle but critical distinction. The system is secure by design, but not yet secure by configuration excellence. And in modern cybersecurity practice, that distinction is often where real-world vulnerabilities emerge.

4. Conclusion

This study set out to examine how effectively SSL/TLS has been implemented within an academic information system, with particular attention to configuration quality rather than mere protocol adoption. The results paint a nuanced picture. On one hand, the system clearly reflects an awareness of modern security requirements, as evidenced by the deployment of TLS 1.3 and the use of strong cryptographic primitives. These are not trivial achievements—they provide a solid baseline against a wide range of common network-based threats.

At the same time, the analysis reveals that security maturity is not solely defined by protocol choice. Subtle configuration gaps, especially the incomplete support for forward secrecy, introduce limitations that directly affect the system's overall resilience. This is where the distinction becomes important. The system is not

insecure in a conventional sense; it performs adequately under standard conditions. Yet it falls short of what would be considered robust or future-proof in environments where data sensitivity and long-term confidentiality matter.

What stands out is how small inconsistencies can carry disproportionate implications. The absence of consistent forward secrecy, for instance, does not immediately expose the system to trivial attacks, but it weakens guarantees over time—particularly in scenarios involving key compromise or retrospective decryption. Similarly, moderate key exchange practices, while acceptable, suggest that not all security mechanisms are being leveraged to their full potential.

There is also a broader takeaway here. In modern web security, the gap between "good enough" and "well-hardened" systems is often defined not by technology, but by precision in configuration. Many vulnerabilities today emerge not from outdated tools, but from incomplete or uneven implementation of otherwise strong mechanisms. The findings of this study align with that reality.

Looking ahead, improving the system's security posture does not require a fundamental redesign. The path forward is relatively straightforward—enforce forward secrecy consistently, refine key exchange configurations, and ensure alignment with current best-practice guidelines. These adjustments may appear incremental, yet their impact would be significant, both in terms of measurable security ratings and actual risk reduction.

In conclusion, the evaluated system demonstrates a solid foundation, but it remains in a transitional state. With targeted refinements and ongoing monitoring, it has the potential to move from adequate protection toward a more resilient and fully optimized security posture.

Acknowledgement

This study has several limitations. The analysis is based on external observation using a single assessment platform, which may not capture internal configuration details or server-side policies. Additionally, the evaluation focuses solely on SSL/TLS configuration and does not include application-layer vulnerabilities or network-level attacks. Future work may extend this analysis by incorporating multi-tool assessments and deeper penetration testing techniques.

Author Contributions

Murni Marisma, Ermadi Satriya Wijaya: Conceptualization, Methodology, Data collection, Formal analysis, Writing—original draft, Writing—review and editing.
Mukhlis Prasetyo Aji, Agung Purwo Wicaksono: Conceptualization, Methodology, Visualization, Writing—review and editing, Supervision.

Funding

This research received no external funding.

Declaration on the Use of Artificial Intelligence

Artificial intelligence support was used in a limited capacity through ChatGPT (OpenAI), based on the GPT-5 family model, solely for language improvement,

including grammar correction, sentence refinement, clarity enhancement, and overall readability of the manuscript. The AI tool was not used for data collection, data analysis, interpretation of results, generation of findings, reference selection, or scholarly decision-making. All research activities, intellectual contributions, methodological design, critical interpretation, and final manuscript approval were carried out entirely by the authors. The authors take full responsibility for the accuracy, originality, and integrity of the published work.

References

Acar, Y., Fahl, S., & Mazurek, M. L. (2016). *You Are Not Your Developer, Either: A Research Agenda for Usable Security and Privacy Research Beyond End Users*. https://doi.org/10.1109/SecDev.2016.20

Alashwali, E. S., Szalachowski, P., & Martin, A. (2020). Exploring HTTPS security inconsistencies: A cross-regional perspective. *Computers & Security*, *97*, 101975. https://doi.org/10.1016/j.cose.2020.101975

Albrecht, M. R., Paterson, K. G., & Watson, G. J. (n.d.). *Plaintext Recovery Attacks Against SSH*.

Altulaihan, E. A., Alismail, A., & Frikha, M. (2023). A Survey on Web Application Penetration Testing. *Electronics*, *12*(5), 1229. https://doi.org/10.3390/electronics12051229

Amann, B., Sommer, R., Vallentin, M., & Hall, S. (2013). No attack necessary: The surprising dynamics of SSL trust relationships. *ACM International Conference Proceeding Series*, 179–188. https://doi.org/10.1145/2523649.2523665

Beurdouche, B., Bhargavan, K., Delignat-Lavaud, A., Fournet, C., Kohlweiss, M., Pironti, A., Strub, P.-Y., & Zinzindohoue, J. K. (2017). A messy state of the union. *Communications of the ACM*, *60*(2), 99–107. https://doi.org/10.1145/3023357

Bhargavan, K., Delignat-Lavaud, A., Fournet, C., Pironti, A., & Strub, P. Y. (2014). Triple handshakes and cookie cutters: Breaking and fixing authentication over TLS. *Proceedings - IEEE Symposium on Security and Privacy*, 98–113. https://doi.org/10.1109/SP.2014.14

Calzavara, S., Focardi, R., Squarcina, M., & Tempesta, M. (2018). Surviving the Web. *ACM Computing Surveys*, *50*(1), 1–34. https://doi.org/10.1145/3038923

Clark, J., & van Oorschot, P. C. (2013). SoK: SSL and HTTPS: Revisiting Past Challenges and Evaluating Certificate Trust Model Enhancements. *2013 IEEE Symposium on Security and Privacy*, 511–525. https://doi.org/10.1109/SP.2013.41

De Ruiter, J., & Poll, E. (n.d.). *Open access to the Proceedings of the 24th USENIX Security Symposium is sponsored by USENIX Protocol State Fuzzing of TLS Implementations Protocol state fuzzing of TLS implementations*. Retrieved https://www.ssllabs.com/ssltest/

Durumeric, Z., Ma, Z., Springall, D., Barnes, R., Sullivan, N., Bursztein, E., Bailey, M., Halderman, J. A., & Paxson, V. (2017). The Security Impact of HTTPS Interception.

24th Annual Network and Distributed System Security Symposium, NDSS 2017. https://doi.org/10.14722/ndss.2017.23456

Holz, R., Amann, J., Mehani, O., Wachs, M., & Kaafar, M. A. (2016). TLS in the Wild: An Internet-wide Analysis of TLS-based Protocols for Electronic Communication. *23rd Annual Network and Distributed System Security Symposium, NDSS 2016*. https://doi.org/10.14722/ndss.2016.23055

Holz, R., Hiller, J., Amann, J., Razaghpanah, A., Jost, T., Vallina-Rodriguez, N., & Hohlfeld, O. (2020). Tracking the deployment of TLS 1.3 on the web. *ACM SIGCOMM Computer Communication Review*, *50*(3), 3–15. https://doi.org/10.1145/3411740.3411742

Hu, Q., Asghar, M. R., & Brownlee, N. (2021). A large-scale analysis of HTTPS deployments: Challenges, solutions, and recommendations. *Journal of Computer Security*, *29*(1), 25–50. https://doi.org/10.3233/JCS-200070

Rescorla, E. (2018). *The Transport Layer Security (TLS) Protocol Version 1.3*. https://doi.org/10.17487/RFC8446

Sheffer, Y., Saint-Andre, P., & Fossati, T. (2022). *Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)*. https://doi.org/10.17487/RFC9325

Sherry, J., Lan, C., Popa, R. A., & Ratnasamy, S. (2015). BlindBox. *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, 213–226. https://doi.org/10.1145/2785956.2787502

Somorovsky, J. (2016). Systematic Fuzzing and Testing of TLS Libraries. *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 1492–1504. https://doi.org/10.1145/2976749.2978411

Vanhoef, M., & Ronen, E. (2020). Dragonblood: Analyzing the Dragonfly Handshake of WPA3 and EAP-pwd. *2020 IEEE Symposium on Security and Privacy (SP)*, 517–533. https://doi.org/10.1109/SP40000.2020.00031

Zhang, Z., Hamadi, H. Al, Damiani, E., Yeun, C. Y., & Taher, F. (2022). Explainable Artificial Intelligence Applications in Cyber Security: State-of-the-Art in Research. *IEEE Access*, *10*, 93104–93139. https://doi.org/10.1109/ACCESS.2022.3204051