



5-2024

Understanding student experiences with TLS client authentication

Clay A. Shubert

University of Tennessee, Knoxville, cshubert@vols.utk.edu

Follow this and additional works at: https://trace.tennessee.edu/utk_gradthes



Part of the [Information Security Commons](#)

Recommended Citation

Shubert, Clay A., "Understanding student experiences with TLS client authentication. " Master's Thesis, University of Tennessee, 2024.

https://trace.tennessee.edu/utk_gradthes/11405

This Thesis is brought to you for free and open access by the Graduate School at TRACE: Tennessee Research and Creative Exchange. It has been accepted for inclusion in Masters Theses by an authorized administrator of TRACE: Tennessee Research and Creative Exchange. For more information, please contact trace@utk.edu.

To the Graduate Council:

I am submitting herewith a thesis written by Clay A. Shubert entitled "Understanding student experiences with TLS client authentication." I have examined the final electronic copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Computer Engineering.

Scott I. Ruoti, Major Professor

We have read this thesis and recommend its acceptance:

Scott I. Ruoti, Doowon Kim, Audris Mockus

Accepted for the Council:

Dixie L. Thompson

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

To the Graduate Council:

I am submitting herewith a thesis written by Clay Alan Shubert entitled "Understanding Student Experiences with TLS Client Authentication." I have examined the final paper copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Computer Engineering.

Scott Ruoti, Major Professor

We have read this thesis
and recommend its acceptance:

Dr. Scott Ruoti

Dr. Doowon Kim

Dr. Audris Mokus

Accepted for the Council:

Carolyn R. Hodges

Vice Provost and Dean of the Graduate School

To the Graduate Council:

I am submitting herewith a thesis written by Clay Alan Shubert entitled "Understanding Student Experiences with TLS Client Authentication." I have examined the final electronic copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Computer Engineering.

Scott Ruoti, Major Professor

We have read this thesis
and recommend its acceptance:

Dr. Scott Ruoti

Dr. Doowon Kim

Dr. Audris Mokus

Accepted for the Council:

Carolyn R. Hodges

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

Understanding Student Experiences with TLS Client Authentication

A Thesis Presented for the

Master of Science

Degree

The University of Tennessee, Knoxville

Clay Alan Shubert

May 2024

© by Clay Alan Shubert, 2024
All Rights Reserved.

For my wife, Katie, whose love and support has been invaluable...

Acknowledgements

I would like to express my gratitude to my wife, Katie, who has always supported and believed in me and my academic endeavors. I would also like to thank my advisor, Dr. Scott Ruoti, for providing me with this opportunity, and his guidance and support throughout this process. I would also like to thank my committee members for their time and academic assistance throughout my time at the University of Tennessee-Knoxville. Finally, I would like to thank my friends and family for their encouragement and support. This work is based upon research supported by the National Science Foundation under award CNS-2238001.

Some quotation...

Abstract

This thesis presents a comprehensive investigation into student experiences with TLS client authentication, highlighting the usability challenges and learning curves associated with this long term key management system. We designed a study that required future innovators in technology and security to use modern-day implementations of this certificate-based authentication system. From this study, we analyzed server logs, project reports, and survey responses from students enrolled in the applied cryptography course. We revealed significant hurdles in the initial setup and long-term key management of credentials used in TLS client authentication, emphasizing the gap between theoretical knowledge and practical implementation skills. Through quantitative results, the study quantified the time investment and error rates students face, and provided a System Usability Scale (SUS) assessment that points to the need for improved features and better resources. Qualitatively, this thesis identifies common pain points, resource utilization, and tool effectiveness from the students' perspectives. It further discusses the implications of these findings for design and delivery, suggesting pathways forward to enhance the practical usability and understanding of key management systems.

Table of Contents

1	Introduction	1
2	Background	4
2.1	Public and Private Keys, Algorithms, and Key Strength	4
2.2	Transport Layer Security (TLS)	6
2.3	What exactly is TLS client authentication?	7
3	Related Works	10
3.1	Usable key management	10
3.2	Secure email	11
3.3	Bitcoin	13
4	Methodology	15
4.1	Assignment One: Setup	15
4.2	Assignment Two: Extended Usage and Replication	17
4.3	Server setup and log collection	20
4.4	Consent gathering	21
4.5	Participants	22
4.6	Analysis methodology	22
5	Key Management Part One	25
5.1	Quantitative Results	25
5.2	Qualitative Results	34

6	Key Management Part Two	37
6.1	Quantitative Results	37
6.2	Qualitative Results	42
7	Discussion	47
7.1	Comparing Key Management Part One and Key Management Part Two	47
7.2	Developers Perspective	49
7.3	Pathways Forward	50
7.4	Future Work	52
8	Conclusion	54
	Bibliography	56
	Appendix	60
	A. Study materials	60
	B. User interfaces	78
	Vita	84

Chapter 1

Introduction

As the volume of data exchanged over the Internet exponentially increases, alongside the rise of connected devices, the role of cryptography in safeguarding this data against unauthorized access and tampering has never been more critical. It is the what encrypts and protects financial transactions, personal communications, and sensitive government information, serving as the first line of defense. Cryptographic protocols ensure the confidentiality and integrity of information as it traverses the network. When breached, the consequences can be far-reaching and highly damaging to organizations and individuals alike.

Many of today's encryption schemes utilize public key cryptography such as secure email, bitcoin, and commit signing. Although these schemes are essential to both confidentiality and integrity, the real-world deployment of advanced cryptographic schemes rely on usable key management to be effective. [Ruoti et al. \(2018\)](#) and [Ruoti and Seamons \(2019\)](#) have already identified designs that help users manage a single key over a short period of time. However, there are important and serious concerns on how key management system designs can help users manage a large number of keys over a large period of time without a centralized platform for key management. In addition, key synchronization and recovery is an important ability for the deployment of advanced cryptographic schemes. In this thesis, we investigate the Transport Layer

Security (TLS) protocol encryption scheme TLS client authentication and what its use can teach us about the state of usable key management and encryption in modern day implementations.

The Transport Layer Security (TLS) protocol is the standard method for securing communication between a client's internet browser and HTTP web servers. TLS plays the integral role of authenticating and encrypting, providing confirmation that a server is legitimate, and that data being transmitted is secure. TLS client authentication involves not only authenticating the server to a user, but also a user to the server. This bilateral authentication is often referred to as mutual TLS (mTLS). There have been many studies confirming the advantages of this method and the attack surface that it reduces including man-in-the-middle, replay, spoofing, and impersonation attacks. [22; 6; 1; 10; 8]. In order to realize these benefits there is a need for long term usable key management including synchronization across devices and a recovery process.

We orchestrated a study consisting of three parts where participants, representative of future cybersecurity and technological professionals, engaged in real-world tasks involving the generation, management, and application of cryptographic keys and certificates. In the first part, participants reflected on setting up TLS client authentication from scratch, emulating the experience of configuring a modern day implementation. The second part asked the participants to reflect on using their keys and certificate through the semester to access a web server. The third part asked the participants to reflect on replicating or adapting this setup across a new device by either synchronizing their certificate and keys or regenerating new credentials, thereby illuminating the complexities of key management in a multi-device context. Throughout the study, we monitored server logs to capture the certificate and key usage of our participants over time.

Our study aimed to answer several questions about student experiences with the TLS client authentication system. First, we want to know how users interact with key management and their experiences doing so. In addition, we want to identify the

main challenges and easements that users perceive when generating and handling cryptographic keys. Furthermore, we want to understand how these challenges and easements are affected by the use of multiple devices and over a period of time. Lastly, we aim to find strategies or tools could improve the usability and manageability of TLS client authentication for users with varying degrees of technical expertise.

This thesis adds important context to the current state of usable key management and encryption by taking a look at current practices, habits, and understanding of future innovators in computer technology and security. We also look at past scholarly works on usable encryption and key management schemes and compare them to the requirements of future designs. We answer the main difficulties identified with long-term key management, synchronization, and recovery. In addition, we contribute data providing a System Usability Score (SUS) [16] for TLS client authentication amongst undergraduate seniors and graduate students. Lastly, we identify several gaps between theoretical security and practical implementation of a long-term, large quantity, decentralized usable key management scheme including potential solutions to these gaps.

The outcomes of this thesis have significant implications for the design and application of future advanced cryptographic schemes. As we chart the course for a future where cybersecurity becomes increasingly paramount, understanding the user experience with advanced key management is not just an academic exercise, but a necessity to ensure that security measures are not just theoretically sound, but practically viable and widely adopted.

Chapter 2

Background

In this section we describe the role of public and private keys, the algorithms that can be used, and the relative strength of different keys in cryptographic schemes. Knowledge of these concepts provides a foundational understanding cryptographic protocols, offering a glimpse into the intricate mechanisms that safeguard our digital interactions.

2.1 Public and Private Keys, Algorithms, and Key Strength

Public and Private Keys In the realm of cryptography, the concept of public and private keys forms the backbone of secure communication over the internet. This system is known as asymmetric cryptography and involves a pair of keys: one public and one private. The public key is openly distributed and can be shared with anyone, while the private key is kept secret by the owner. The dual-key mechanism ensures that anyone can encrypt a message using the public key, but only the holder of the paired private key can decrypt it, thereby enabling secure and private exchanges.

Public and private keys also facilitate digital signatures, where a sender can sign a document with their private key to prove authenticity. Recipients can then use the

sender's public key to verify that the signature is valid. This two-key structure is integral to numerous security protocols, including Transport Layer Security (TLS), where it establishes trust and confidentiality between parties.

Algorithms Cryptographic algorithms are mathematical functions used for encryption, decryption, and various security applications. These algorithms are categorized into two primary types: symmetric and asymmetric. Symmetric algorithms use the same key for both encryption and decryption, whereas asymmetric algorithms, like those used in public-private key cryptography, use different keys.

Common asymmetric algorithms include RSA (Rivest–Shamir–Adleman) [24], ECC (Elliptic Curve Cryptography) [17], and DSA (Digital Signature Algorithm) [15]. RSA, one of the first public-key cryptosystems, is widely used for secure data transmission. ECC offers similar functionality to RSA but is more efficient, enabling it to use smaller keys for the same level of security. DSA is used primarily for digital signatures. Most algorithms are viewed as a black-box in cryptography which can and do lead to security oversights resulting in compromise. A great example of this in recent times involved DSA. DSA relies heavily on the uniqueness, secrecy, and entropy of a random signature value, k , even leaking a few bits of k in several signatures could reveal the private key. In 2010, Sony, a popular electronics company, had their private key used to signed game console software revealed by a hacking group, because they failed to generate different k values for each signature [5]. This example reveals the challenges that face cryptographers when developing secure algorithms.

Key Strength Key strength is a measure of how resistant a cryptographic key is to being cracked. It is determined by several factors, including the algorithm used, the size of the key, and the computational power required to break the encryption. Generally, a longer key provides greater security, as the number of possible keys—and, therefore, the difficulty of guessing the correct one—increases exponentially with key length.

For example, in RSA, a key length of 2048 bits is currently considered the minimum for secure communications, with longer keys providing even stronger protection. In ECC, due to its efficient algorithm, a shorter key length can provide equivalent security to a longer RSA key. Key strength is not static; it evolves as computational capabilities grow and cryptographic analysis techniques improve. Ensuring that key lengths stay ahead of computational advances is crucial to maintaining the security of encrypted information.

2.2 Transport Layer Security (TLS)

Transport Layer Security (TLS) is an encryption protocol that provides secure communication over a computer network. Developed as the successor to Secure Sockets Layer (SSL), TLS is the standard means for creating an encrypted link between a web server and a browser, ensuring that all data passed between them remains private.

TLS has three main core principles: Encryption, Authentication, and Integrity. For encryption, TLS uses asymmetric cryptography for the initial handshake between the client and a server, followed by symmetric encryption for continuous communication. This approach ensures that sensitive data cannot be intercepted nor understood by eavesdroppers. Through the use of digital certificates, TLS facilitates authentication between the communicating parties. Most TLS implementations utilize server-side authentication in which servers are required to present a valid certificate to the client, verifying that the server is indeed who it claims to be. TLS provides message integrity checks using Message Authentication Codes (MACs), allowing data to be verified during transmission.

The TLS handshake is the system process that establishes the core principles for the TLS connection. During this handshake, the following steps occur:

1. Protocol Version Negotiation: The client and server agree on the version of the TLS protocol to use.

2. Cipher Suite Negotiation: They also agree on the cipher suite, which includes the encryption algorithm, key exchange algorithm, and MAC algorithm.
3. Public Key Exchange: The client and server exchange public keys through certificates. The client verifies the server's certificate against a list of trusted certificate authorities (CAs).
4. Key Generation: The client and server generate session keys for encryption and decryption, ensuring that they are the only parties able to read the communication.
5. Authentication and Finalization: Once the handshake is completed and the encryption is established, the secure transmission of data can begin.

Figure 2.1 visualizes the three-way handshake that occurs in most standard and accepted TLS connections.

The critical role of TLS in modern security cannot be overstated. It is integral to protecting user data across countless applications. Recognizing its importance, many browsers and search engines have started to enforce a "secure by default" policy, marking non-TLS encrypted websites as untrustworthy. This shift underscores the internet community's consensus on the necessity of TLS for maintaining a secure and trustworthy digital ecosystem.

2.3 What exactly is TLS client authentication?

TLS client authentication, or mTLS, is a security mechanism designed to authenticate both the client and server to each other in a communication session. This contrasts with the more common practice where only the server is authenticated (via its SSL/TLS certificate) while the client remains unauthenticated. In this encryption scheme, both parties generate, exchange, and validate digital certificates. The general steps are as follows.

1. Client connects to server.
2. Server presents TLS certificate to the client.
3. Client verifies server's certificate against list of trusted certificate authorities.
4. Client presents TLS certificate to the server.
5. Server verifies the client's certificate against a list of trusted certificate authorities.
6. Server grants the client access.
7. Client and server exchange information over encrypted TLS connection.

Figure 2.2 illustrates this process of the TLS three-way handshake [7; 2]. Upon successful validation, the server grants access to the user, establishing an encrypted, mutually authenticated communication channel. TLS client authentication has been around for quite some time [11], and is often used in internet of things (IoT), banking, healthcare, and government applications [19; 14; 9].

Our study simulates a CA that users must interact with to obtain a signed certificate. Our server then validates the client based on the CA's public key, in which the server trusts. TLS client authentication is particularly relevant when there is a limited number of clients where connecting to a specific confidential web service is required. This means in practical applications it is not implemented organization wide, but on a subset of specific users, possibly admins.

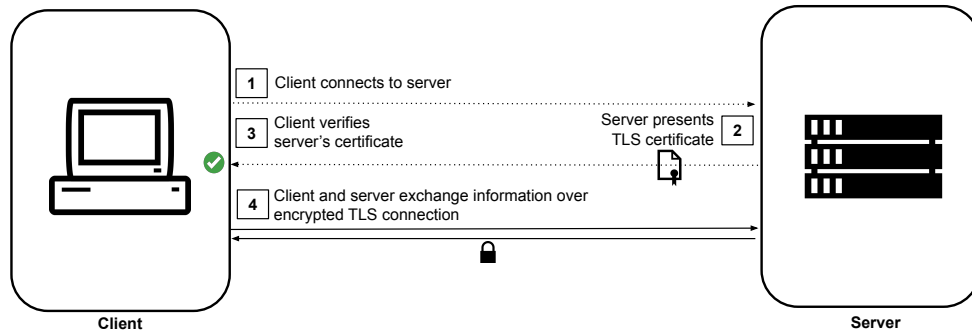


Figure 2.1: Diagram of the standard TLS three-way handshake.

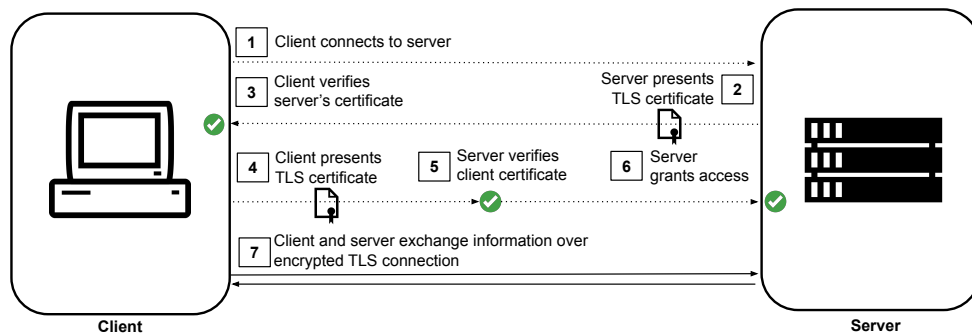


Figure 2.2: Diagram of the mTLS three-way handshake.

Chapter 3

Related Works

There is a wealth of literature that relates to the theoretical benefits of key management schemes. However, there is a lack of exploration on how user's view and practice key management in their workflow. We first lay some background on TLS client authentication at a high level. Then we transition to the importance of key management in this encryption scheme. Finally, we examine relevant work on usable key management and encryption in implementations such as secure email and bitcoin.

3.1 Usable key management

Key management is the process of generating, storing, and distributing cryptographic keys. In the context of advancing usable key management within Public Key Infrastructures (PKIs), the work by Carl M. Ellison (1999) [12] serves as a pivotal reference. He critiqued the traditional assumptions underlying early PKI designs, notably the reliance on global directories and their security implications. He argued for a reevaluation of PKI foundations, emphasizing the need for a usable PKI that accommodates the realities of digital communication and identity verification on the internet. Ellison's insights contribute significantly to the discourse on making PKI systems more accessible and practical for users. This work stands as a critical

junction point, encouraging a shift towards PKI systems that prioritize usability without compromising security, thereby addressing a fundamental challenge in digital cryptography and key management practices.

There are many existing security protocols and authentication methods that utilize key management. In the following subsections, we discuss related studies that involve user experiences with key management schemes. These serve as comparison to the discussion of key management in the context of TLS client authentication.

3.2 Secure email

Secure email has been widely studied and is firmly implemented across several industry applications. However, the most challenging part of widely adopted usable secure email arises directly from key management. [29; 26]. Ruoti et al. (2018) have already provided a comparative analysis of various secure email tools, emphasizing the usability implications of different key management approaches. By evaluating systems like Pwm, Tutanota, and Virtru, their work uncovered the preference for integrated solutions that seamlessly blend with existing email accounts, thereby minimizing the learning curve and enhancing adoption rates. This preference underscores the importance of designing integrated systems in TLS client authentication that are found to be intuitive for end-users.

Ruoti et al. (2019a) also delved deeper into the usability of secure email tools, employing a paired-participant methodology to mimic real-world usage scenarios more closely. This innovative approach shed light on the user experience from both senders' and receivers' perspectives, revealing that tutorials and integrated solutions significantly impact users' ability to successfully navigate secure communication tools.

There have been several investigations into automating the key management process in secure email. Namely, Mueller and Michalek (2024) [20] focus on automating the creation and management of S/MIME certificates for email users, significantly enhancing usability and security. Their implementation of an extended

ACME protocol within the Thunderbird email client demonstrates a substantial reduction in both the time and complexity involved in obtaining S/MIME certificates. This advancement is crucial for the widespread adoption of encrypted email communications.

In reality if TLS client authentication were to be implemented it would require users to potentially maintain and manage several different keys and certificates. This is already common in the case of users utilizing secure email methods whom also own several email accounts. Banday and Sheikh [3] conducted a study on the usability of S/MIME when users were required to manage multiple email address certificates. They created three email accounts each with a unique email certificate and mail server. They then asked participants to send and receive emails from each of the accounts. They found that users committed many errors when sending and viewing signed/encrypted emails to recipients who had more than one email address associated with them. They also found that specifically, users found the process of selecting the correct certificate for sending/viewing S/MIME emails to be difficult and tedious.

Key management in secure email and TLS client authentication share a fundamental challenge: the complexity of managing keys and certificates for end-users. Both domains require users to handle potentially multiple keys and certificates, which can lead to errors and usability issues. The studies highlighted emphasize the preference for integrated solutions that reduce the learning curve by blending seamlessly with users' existing workflows, such as email clients or web browsers. Moreover, automation of key and certificate management processes, as seen in the implementation of extended ACME protocol for S/MIME certificates in email, points to a similar need for simplifying TLS client authentication. This suggests that strategies improving secure email usability—like integrated systems, user-friendly interfaces, and automation—can be adapted to enhance TLS client authentication, making it more accessible and manageable for users.

3.3 Bitcoin

Bitcoin and blockchain technology have been widely investigated as offering potential avenues for authentication and enhanced decentralized security. The use of this technology has wide applications that have been explored in several studies.

Pal et al. (2021) [23] delves into the application of blockchain technology for key management, proposing a decentralized approach that offers enhanced security and resilience against traditional vulnerabilities. This approach leverages the immutable and distributed nature of blockchain to facilitate secure, transparent key distribution and management mechanisms. Together, these studies underscore the ongoing evolution of key management solutions, emphasizing the importance of usability, automation, and innovative technologies in addressing the challenges of secure digital communication.

Furthermore, Nguyen et al. (2020) [21] explores the application of blockchain technology for authenticating educational certificates in Vietnam. It highlights blockchain's potential to address the issue of fake certificates by ensuring data integrity, transparency, and trust. This approach is particularly relevant to TLS client authentication, as both domains emphasize the importance of secure and verifiable digital credentials. Just as TLS client authentication leverages cryptographic certificates to secure online communications, the proposed blockchain system aims to secure and verify academic credentials, thereby enhancing trust in digital certifications. This study underlines the broader applicability of blockchain for securing various forms of digital authentication and integrity, resonating with the principles underlying TLS client authentication.

Blockchain technology shares a commonality the the goals of TLS client authentication, the potential to securing digital identities. Ruoti et al. (2019) [28] conducted a thorough analysis of blockchain technology, distinguishing it from similar technologies and exploring viable use cases. It systematically addresses common questions about blockchain, such as its definition, capabilities, and applications,

using grounded theory to analyze a broad spectrum of literature. The discussion on blockchain's potential for securing digital credentials is particularly relevant to TLS client authentication, underscoring blockchain's role in enhancing online security and trust through decentralized, verifiable digital identities and transactions. This research contributes significantly to understanding blockchain's applicability and limitations, providing a comprehensive overview beneficial for research and practical applications in secure online communications and authentication methods like TLS.

Of particular symmetry to that of TLS client authentication, bitcoin and blockchain technology have similar key management issues. Eskandari et al. (2018) [13] delves into the challenges and usability issues surrounding key management in Bitcoin, identifying that while Bitcoin introduces innovative solutions to key management, it also inherits many traditional challenges. The analysis presented in the paper underscores the necessity for user-friendly and secure management practices, echoing the broader need in TLS client authentication for approaches that simplify user interaction while ensuring security.

Chapter 4

Methodology

We conducted an IRB-approved user study. We performed quantitative and qualitative analysis on server logs and two project reports that students enrolled in the COSC 483/583 applied cryptography course completed. This section gives an overview of the student demographics, assignments that were completed, types questions asked, and methodology for analyzing the generated data. Appendix A contains copies of all the study materials.

4.1 Assignment One: Setup

The "Usable Key Management (Part 1)" project was designed to immerse students in the practical applications of TLS client authentication. Data for the first part of our study was collected from this assignment. This experience aimed to engage students in the process of establishing a secure communication channel using cryptographic keys and certificates with a server for the first time. The exact assignment page that students were given can be seen in Appendix A.

Requirements: Students were tasked with gaining access to a pass-off server, which required student to authenticate themselves using TLS client authentication. This required the student's browser to transmit a certificate and a digital signature, which

is created using the private key associated with the certificate, to the server. Upon receipt, the server verifies the certificate's authenticity and grants access.

To achieve access to the pass-off server, students were required to:

1. Generate a cryptographic key pair: This involved creating a private key (kept secret) and a public key (shared publicly).
2. Obtain a signed certificate for the key pair: Utilizing a designated website, students submitted their certificant signing request or CSR file to receive a corresponding signed certificate unique to them.
3. Register their signed certificate and private key with the browser: This step ensures the browser can use the provided certificate for TLS client authentication.

Successful completion of these steps would allow students to log into the pass-off website without encountering any errors.

Reflection Upon completing the project, students were instructed to document their experiences in a detailed report, covering:

- The time taken to complete the project.
- Steps undertaken to achieve the project goals.
- Challenges faced and unsuccessful attempts.
- The utilization of information sources and their usefulness.
- Tools used throughout the process, including those eventually discarded, with an emphasis on the ease or difficulty of use.

Questions and Feedback Students answered questions regarding the easiest and hardest steps in the setup process, suggesting potential improvements. They also completed the After-Scenario Questionnaire (ASQ) [18] and the System Usability Scale (SUS) [16] to reflect on their satisfaction with the setup process, the ease of use, and the overall system usability.

Guidance and Autonomy To simulate real-world cryptosystem deployment experiences, students were encouraged to explore online resources and tools independently without direct instructions from instructors or teaching assistants (TAs). However, guidance was available after significant effort without progress, to ensure students could eventually access the grading server and complete the project.

This project not only aimed to enhance students' technical skills in cryptography but also to foster an environment of self-directed learning and problem-solving, mirroring the challenges professionals face in real-world cybersecurity scenarios.

4.2 Assignment Two: Extended Usage and Replication

The second project was titled “Usable Key Management (Part 2)” took the first project a step further, asking students to access the pass-off website from a new device that had not been used before in the semester. Data for the second and third part of our study was collected from this assignment. To ensure the completability this project we suggested that students visit the University of Tennessee Hydra and Tesla lab rooms. These locations contain public lab machines that students can utilize. We provided them with the option to complete this project by choosing to synchronize their existing credentials or regenerate new credentials, as they did in part one. This allowed us to gain valuable insight into long term key management practices and understand reasoning for the choices made. Finally, students were asked to complete several reflection questions. Part two of usable key management

included an after-scenario questionnaire (ASQ) [18], system usability scale (SUS) questionnaire [16], several self-report, general questions to document the exact steps taken and perceived usability, and thought exercises to demonstrate what students had learned throughout their experience. The exact assignment page that students were given can be seen in Appendix A.

The "Usable Key Management (Part 2)" project was structured to deepen students' understanding of TLS client authentication by reflecting on their semester-long experience and exploring the intricacies of managing client certificates across multiple devices.

Requirements The core requirement for the second assignment involved accessing the pass-off server from a new device. This task required the use of either a different personal computer or on a university Tesla or Hydra lab machine. The successful access from a new device was verified through a specific pass-off link, demonstrating the student's ability to navigate TLS client authentication in a multi-device environment. In real world context, the proper result would be that students would synchronize their existing certificate and keys to a new device rather than the student repeating the steps in the first assignment. Overall, the semester-long reflection and project report was broken down into three main sections.

Multi-device TLS Client Authentication The first part of the reflection questions asked students about their experiences while completing the second assignment. This included:

- Duration and steps taken to complete the project, including unsuccessful attempts.
- The choice between synchronizing an existing certificate to the new machine or obtaining a new certificate, with an explanation of the chosen approach.

- After-Scenario Questionnaire (ASQ) responses reflecting the student’s satisfaction with the ease, time, and support information found when setting up TLS client authentication on a second device.
- Discussion on the easiest and hardest steps in the process, suggested improvements, and any additional feedback.

Semester-long Reflection Next, we asked students to reflect on their experiences using TLS client authentication to access the pass-off server throughout the semester. This included asking several questions about their experience such as:

- An overview of the browser used, authentication steps, and management of the certificate file and private key.
- System Usability Scale (SUS) questions to gauge perceptions of TLS client authentication’s complexity, integration, and usability.
- Reflection on the ease or difficulty of using TLS client authentication and proposed changes for enhancing the experience.

Thought Exercises The last part of the reflections was dedicated to determining what students had learned throughout this process and if they fully understood the concepts involved with the project sequence. We asked questions targeting the:

1. Exploration of security benefits or drawbacks of TLS client authentication compared to password-based authentication.
2. Security implications of synchronizing an existing certificate versus obtaining a new one for a second device.
3. Steps and security concerns involved in regaining access to the pass-off server after losing a certificate or private key, or if the private key is stolen.

4. Consideration of how a Certificate Authority (CA) and pass-off server might detect and respond to a stolen certificate scenario.

We also determined acceptable and expected responses for each of our thought exercises.

1. TLS vs. Password Authentication: TLS client authentication eliminates the risk of password theft and phishing attacks by using cryptographic certificates. However, it requires users to manage certificates and keys, which might be complex for some.
2. Synchronizing vs. New Certificate: Synchronizing certificates is convenient but poses a risk if the device is compromised. Obtaining a new certificate for each device enhances security by limiting the impact of a single device compromise.
3. Stolen/Loss of Certificate or Key: Losing a certificate typically requires obtaining a new one, while losing a private key necessitates revoking the current certificate and starting the process anew to maintain security.
4. Detecting and Responding to Certificate Theft: The CA and server might use anomaly detection to identify unusual access patterns indicative of certificate theft. Revocation lists or real-time certificate status protocols can prevent the use of stolen certificates.

This comprehensive project aimed to solidify students' grasp of TLS client authentication by challenging them to apply their knowledge in a real-world context, emphasizing self-directed learning and problem-solving within the confines of modern cryptographic systems.

4.3 Server setup and log collection

A Certificate Authority (CA) was created in order to sign certificates for the students based on the validity of their certificate requests. The CA was given strict

requirements making sure that each field provided in the request was valid and formatted properly. This would be required in real world implementations of TLS client authentication. The requirements to receive a signed certificate based on the request included: User ID (Student NetId), Common Name (firstname lastname), Organization (University of Tennessee), Organizational Unit (Department of Electrical Engineering and Computer Science), Locality (Knoxville), State (Tennessee), Country (US), and the cryptographic key must have at least 128-bit equivalent security (2048 bit or greater sized key).

The pass-off server was a web application that students in the course would use to check the accuracy of intermediate values obtained from code in other projects throughout the semester. There was an nginx reverse proxy serving this application that had "*ssl_client_verify*" enabled which required the students to present a certificate during the TLS handshake. This proxy also stores and references the CA certificate in order to verify that the client certificate being presented was signed by a trusted source. This follows present-day and best practice implementations of TLS client authentication.

4.4 Consent gathering

An essential step was the gathering of informed consent from students for the use of their assignment reports in research. This process was facilitated through a detailed Informed Consent form, which was distributed via email to students at the end of the semester. The study materials and consent form were approved by our Institutional Review Board and can be found in Appendix A. Participants were required to acknowledge their voluntary participation, their age of consent, and their understanding that they could terminate participation at any time. Reflection and log data was then parsed to remove the data of students who did not provide their informed consent at the time of analysis.

4.5 Participants

The participants of this study consisted of students enrolled in the Tickle College of Engineering department of Computer Science and Electrical Engineering (EECS). These students are majoring in Computer Science, Computer Engineering, or Electrical Engineering and enrolled in the COSC 483/583 applied cryptography course. They are representative of future innovators in technology and security. There were a total of 34 students who enrolled in the course, 22 of whom provided informed consent and allowed us to qualitatively analyze their two reports reflecting on their experiences. The age of our study's participants are reflective of college undergraduate and graduate students ranging from 18 years old to 24 years old.

4.6 Analysis methodology

Quantitative Data We began quantitative analysis by removing the reports and log data of those who did not provide informed consent by the end of the semester. This provided us with a dataset that we could study for concrete usability habits and patterns. We utilized server logs to analyze the amount of failed attempts against our CA including the error codes corresponding to why they were rejected. Also, we determined the time it took to setup access by subtracting the timestamp of the first failed request from the timestamp of the first successful request.

We were also able to pull key length data from CSR attempts that were both rejected and accepted by the CA. This was achieved by collecting the raw certificate request data that was sent by the client when negotiating a connection with the pass-off server and utilizing the OpenSSL cryptography library to extract details about the certificate request. Giving insight into the security considerations that were made during the key generation process. Our logs also collected the raw signed certificates used throughout the semester by each student, allowing us to analyze their use over time.

In addition to log data, we scored SUS and ASQ questionnaires to gain information on the perceived usability and confidence level students reported in both assignments to compare their changes over time.

Qualitative Data In our qualitative analysis of the student reports, we performed thematic analysis to unearth common themes within the students' reflection responses. This process began with a comprehensive initial reading of each report, aimed at understanding the full spectrum of student experiences and perspectives. Following this, we engaged in a detailed coding process, identifying and labeling specific text segments that encapsulated significant insights related to our predefined themes. These themes, centered around the easiest and hardest steps identified by students, management of certificates and keys over time, and certificate synchronization, served as focal points for our qualitative investigation.

After initial reading and identifying our focus points, related codes were grouped to form overarching themes in order to structure the analysis of the data. This grouping allowed us to dive into the experiences of students, examining both the facilitators and challenges they encountered in navigating TLS client authentication and key management processes. By comparing the themes across the study's two assignments, we could evaluate the learning curve students faced, highlighting any shifts in the ease or difficulty of tasks as they gained more experience.

One of the pivotal aspects of our analysis shifted focus towards students' experiences in managing their keys and certificates throughout the semester. This examination provided insights into the practical aspects of key management, revealing how students navigated the complexities of maintaining the security and integrity of their cryptographic credentials over an extended period. The decision-making process regarding whether to synchronize certificates in the second assignment was particularly telling, shedding light on the considerations students made in response to managing security across multiple devices.

The thematic analysis culminated in a reflective synthesis of our findings, drawing attention to key challenges, successes, and pathways forward. Through this process, we gained a deeper comprehension of student interactions with managing keys and credentials, but also identified valuable knowledge gaps that led to pathways forward in order to improve usability of the TLS client authentication system.

Chapter 5

Key Management Part One

In this section, we report the quantitative and qualitative results from the first assignment. We start by reporting the perceived usability score for TLS client authentication. Next, we report survey scores on the ease, time invested in completion, and the level of support received throughout the process. Then, we report the amount of time taken to complete the task, as well as the amount of errors encountered along the way. Finally, we report on the key length security choices observed in certificate keys and credentials.

5.1 Quantitative Results

System Usability Scale (SUS) We asked the students to complete System Usability Scale questions in their reflections. We leverage the work of previous researchers who have contextualized a range of scores to more descriptive categories of usability. The first impression score of 55.0 is rated as "Marginal Low" usability and receives a D grade. This indicates that TLS client authentication has a large learning curve and was perceived as not intuitive to use. This acceptability range and grade corresponds to an adjective rating in the middle of the "Ok" range. The distribution of scores can be visualized in a violin plot in Figure 5.1 showing a wider distribution of ratings between 40-65 still generally lower than most systems. The

system did receive more scores below 40 than above 70 indicating that the overall usability was perceived to be lower.

After Scenario Questionnaire (ASQ) We also asked the students to complete a short survey designed to gather scores on the ease, time invested in completion, and the level of support received throughout the assignment. Each score for the after scenario questionnaire was calculated by averaging the 1 through 7 score. We then took the Q1, median, and Q3 from all the survey scores to visualize the score in a violin plot showing the distribution amongst reports in Figure 5.2. The median score for the first assignment was relatively low at 2.5 out of 7. Combined with the perceived system usability scores, this shows that students were not confident in the process, and believed there to be a lack of support when generating a CSR and accessing the pass-off site.

Time to setup In addition to investigating the perceived usability, we tracked and asked the students how much time they estimated that it took them to access the pass-off server successfully. In order to determine the recorded time for students to successfully access the pass-off server, we subtracted the time difference between the first failure and the first success logged. The initial setup times recorded by logs and reported in reflections were relatively long, highlighting the complexities and challenges that are inherent when comparing knowledge to the practical implementation of cryptographic protocols. We then visualized the distribution of setup times in a violin plot in Figure 5.3. The geometric average time to setup recorded by server logs was 2.29 hours. A geometric average was used in order to reduce the effect of outliers on the calculated average. Students self-reported times were averaged in the same way resulting in a geometric average of 2.15 hours, remaining consistent with the recorded log times.

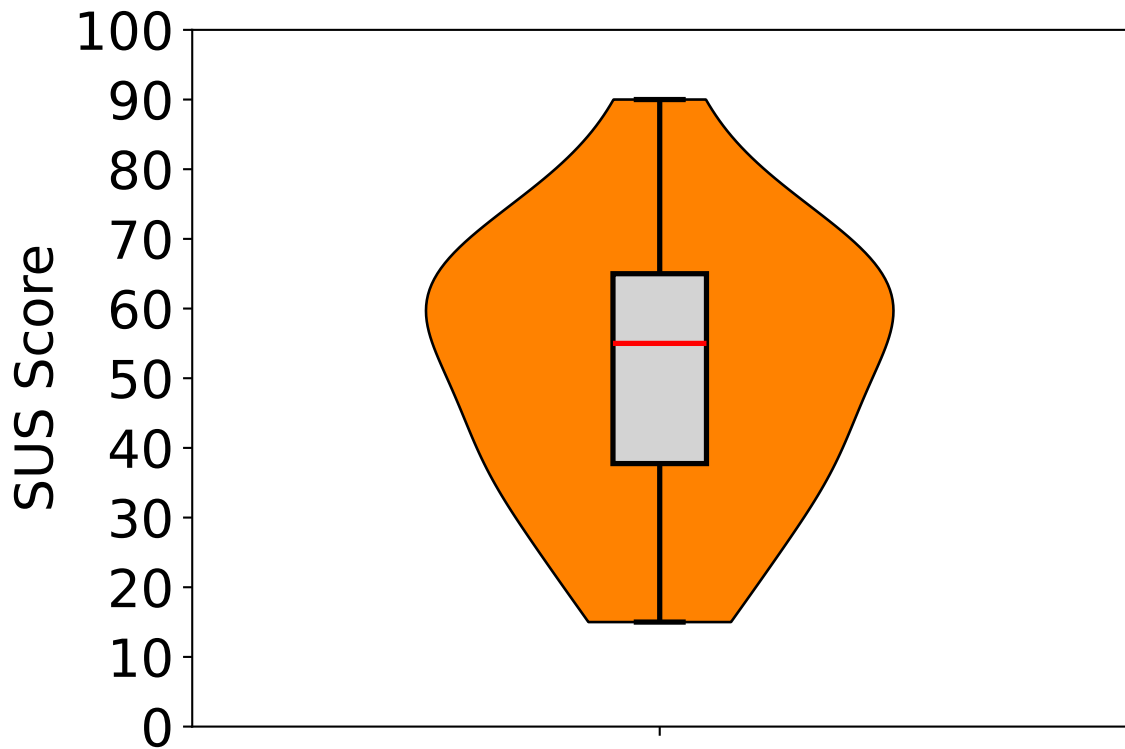


Figure 5.1: Key Management Part One System Usability Scale (SUS) violin plot shows the distribution of students' scores. This plot shows a wider distribution of ratings between 40-65 still generally lower than most systems. The system also received more scores below 40 than above 70.

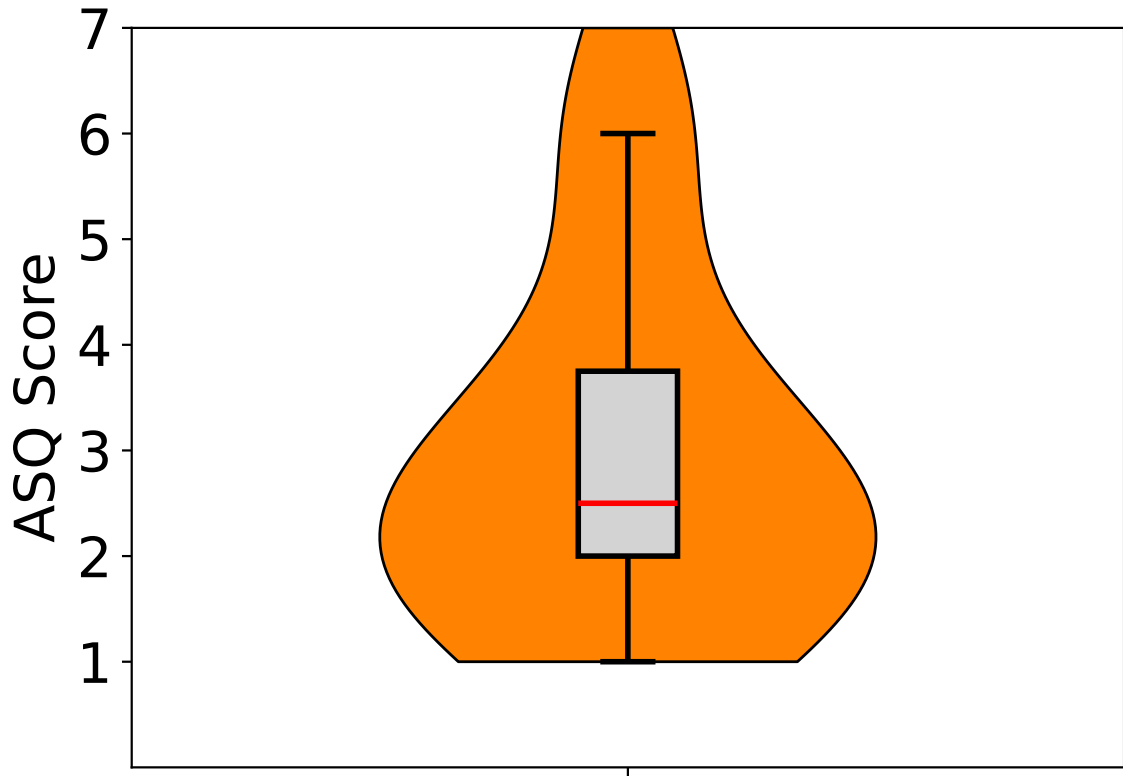


Figure 5.2: Key Management Part One After Senario Questionnaire (ASQ) score violin plot shows the distribution of students' ratings. The median score for the first assignment was relatively low at 2.5 out of 7.

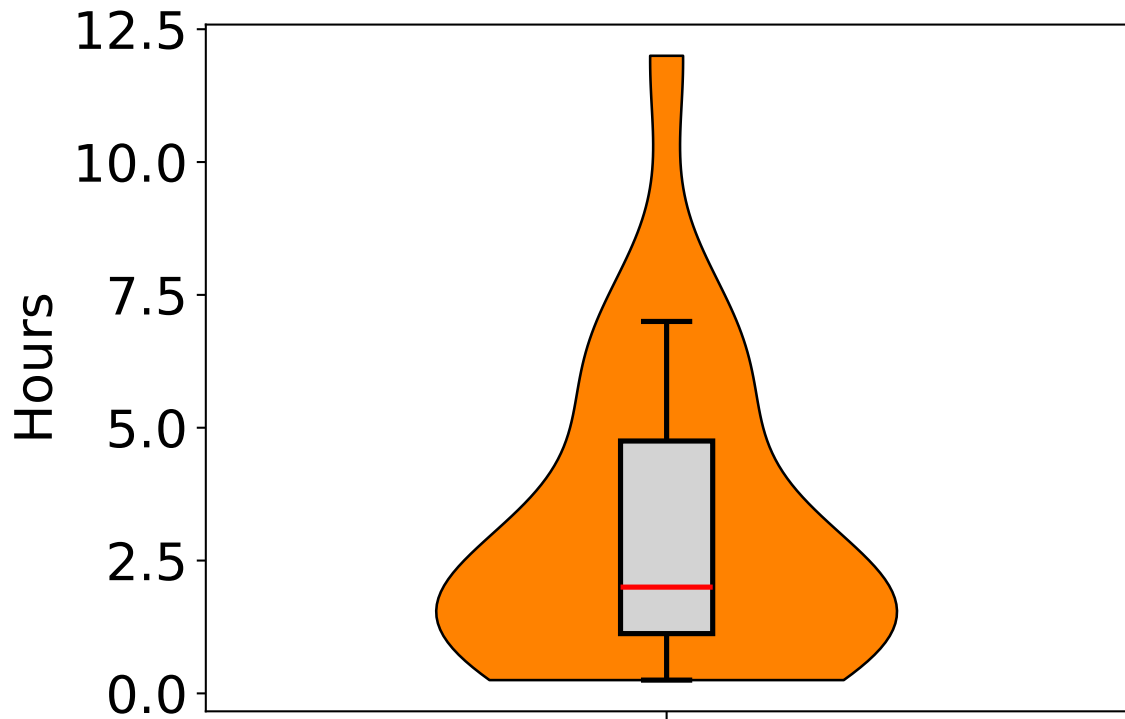


Figure 5.3: Key Management Part One time to setup violin plot shows the variation of times that it took students to complete the first assignment. The geometric average time to setup recorded by server logs was 2.29 hours. A geometric average was used in order to reduce the effect of outliers on the calculated average.

CSR errors We also tracked server logs towards our CA. This allowed us to gather more insight into the attempts being made towards the CA and what students received the most errors with. Logs were collected from the inception of our CA server until the end of the semester. For analysis, we focused on capturing the data between the beginning of the semester until the second assignment was released. Analysis of the CSR errors revealed a stark learning curve, underscored by the considerable disparity between 178 rejected certificate signing requests (CSRs) and 90 accepted ones. We visualized these errors by the count of occurrence in error logs per individual error in Figure 5.4. At first glance, there appears to be a very large number of errors for only 22 students. This is because there could be more than one error in any given submission. Reviewing the data manually showed that this was true; there was a plethora of initial hurdles faced by students, primarily due to missing CSR attributes and failure to meet the CA's minimum 128-bit security criteria. Because these errors were very trivial it shows that the students may have not expected that the certificate metadata attributes must be an exact copy of the requirements specified. One small error or mistype would lead to a rejected request. Furthermore, we determined the total number of failed CSR attempts per student by mapping each of their NetIDs to each certificate request. These results showed that each student had an average of 7.5 (≈ 8) attempts before successfully submitting a CSR and receiving a signed certificate, further supporting the conclusion that students did not find the process intuitive.

Final key length As previously mentioned, we noticed a significant amount of the CSR rejections were due to key sizes that did not meet the CA's 128-bit security requirement. Analysis of these initial key size preferences among students offered insights into their security considerations, with a majority of the submissions gravitating towards more secure 4096-bit keys that achieve the security requirement. Despite this inclination, there were significant learning opportunities, highlighted by 39% of submissions utilizing key parameters that did not fulfill the 128-bit equivalent security requirement, illustrating the complexity of aligning theoretical knowledge

with practical implementation]. We visualized the key length choices made in both accepted and rejected submissions as a percentage of the total submissions in Figure 5.5. We also added a dotted red line to show that lengths above the line would be rejected by the CA. We can compare and contrast key length choices from all CSR submissions with only the submissions that succeeded to get a better view of how students changed their choices after being presented an error message. From Figure 5.6, we can see that after being presented with the error either once or several times students landed on 4096 as their final submitted choice 63% of the time. Interestingly, a 6144-bit key size also saw a large percentage increase in preference up to 18% from 6% of submissions.

Unique certificates We also aimed to uncover whether students were able to successfully utilize their initial certificate without regenerating another one before the second assignment. This would highlight that there were significant usability issues with long term key management. In order to track this behavior, we referred to project log data collected from users submitting their intermediate values for labs on the pass-off server. Since the first lab requiring access to the pass-off server occurred after the submission of the study's first assignment and our study's second assignment was released after the completion of the final lab, this data was sufficient for our analysis. From this data, we gathered a list of all the certificate serial numbers and tied them to each student via their NetID. Each unique serial number was counted per NetID to provide a list of the total number of certificates used per student prior to the second assignment's release date. We found that 59% (13/22) of the students had generated more than one certificate prior to completing our study's second assignment. This highlights that the majority of students experienced issues maintaining their keys and/or certificate throughout the semester. This observation remains consistent with the conclusion that there are challenges encountered in practical key management scenarios where credentials are forgotten or the certificate is not properly stored/maintained. Furthermore, this provides even more challenges

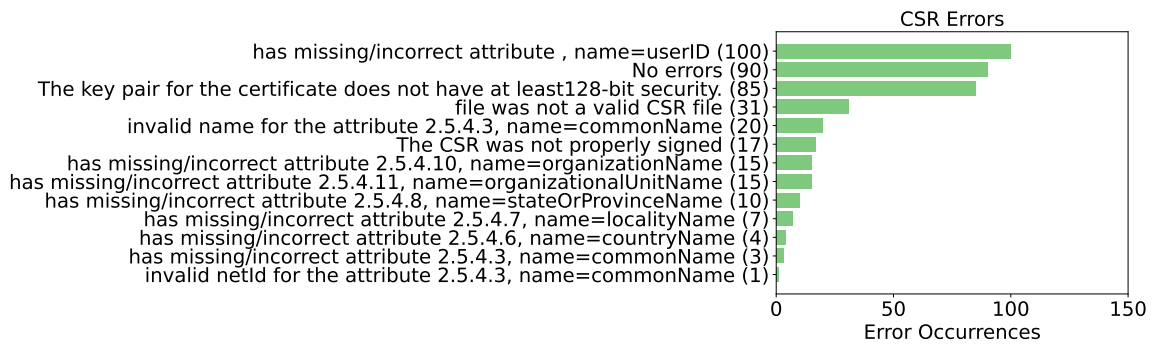


Figure 5.4: Bar plot showing the amount and types of errors that students encountered the most. there was a plethora of initial hurdles faced by students, primarily due to missing CSR attributes and failure to meet the CA’s minimum 128-bit security criteria.

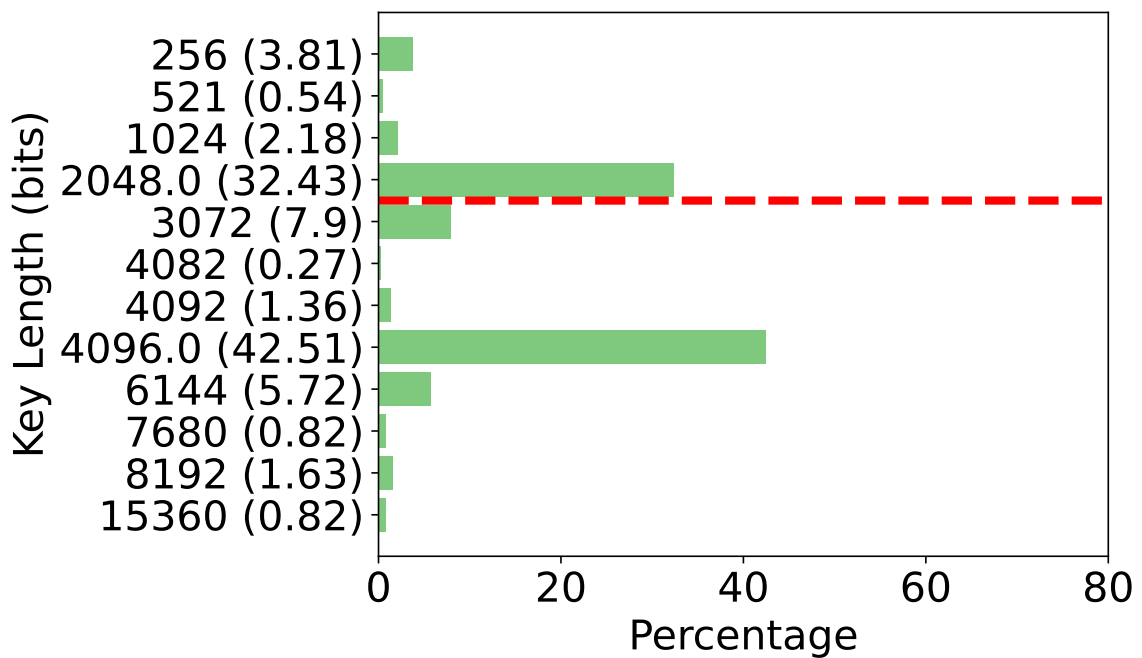


Figure 5.5: Percentage of CSR submissions for each key length before being accepted. Key lengths above the dotted red line would be rejected by the CA. 39% of submissions utilized key parameters that did not fulfill the 128-bit equivalent security requirement

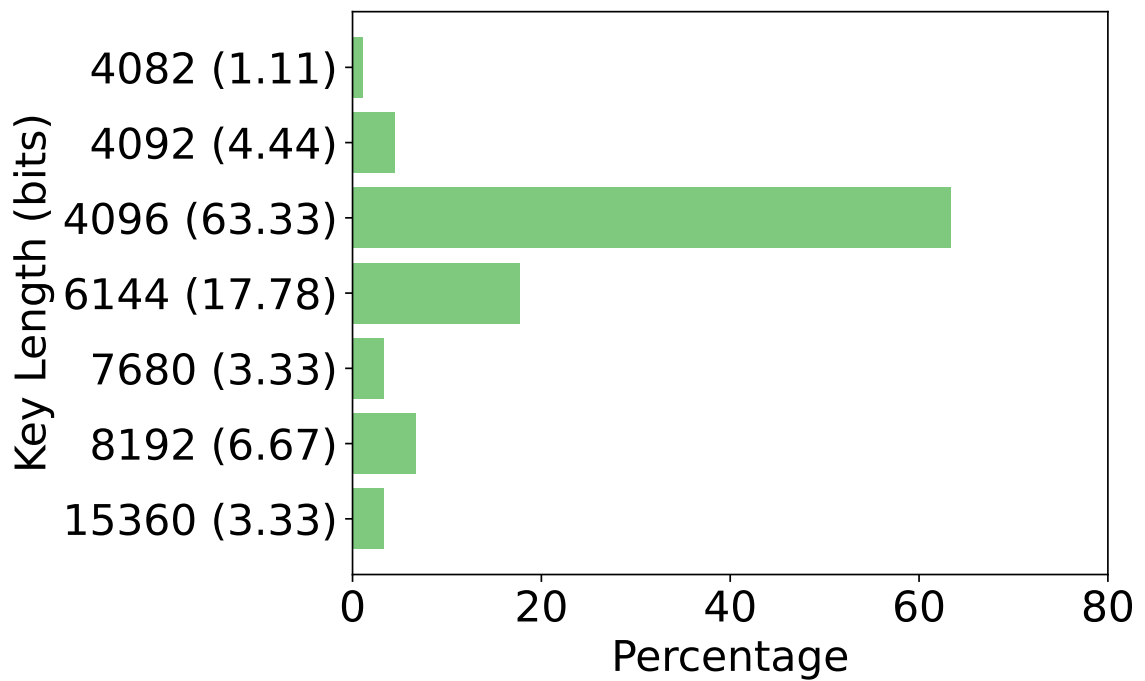


Figure 5.6: Percentage of CSR submissions for each key length after being accepted. After being presented with the error either once or several times students landed on 4096 as their final submitted choice 63% of the time. Interestingly, a 6144-bit key size also saw a large percentage increase in preference up to 18% from 6% of submissions.

for developers hoping to identify which certificates need to be revoked, given that each user has multiple identities. Failing to properly revoke certificates can lead to individuals picking up a stale certificate that the server still validates, granting them unauthorized access.

5.2 Qualitative Results

What went well? In general, students found generating the cryptographic key pair and submitting their CSR (Certificate Signing Request) relatively straightforward once they overcame initial hurdles with OpenSSL commands. Utilizing these tools, several students were able to successfully generate the key pairs and CSR needed for TLS client authentication. The documentation and online resources for OpenSSL played a critical role in guiding students through this process. For instance, one student described using OpenSSL to generate a key pair and CSR as the easiest part of the process due to clear, step-by-step instructions found online.

“Generating the CSR and private key was straightforward as OpenSSL provided a single command for creating the CSR and key pair.” (S1)

The standard attributes for OpenSSL were reported to be very easy to input since the tool will prompt the user for each field entry. However, as part of the study we asked students to add a custom attribute to their request which was more challenging for students.

Furthermore, uploading the CSR to receive the signed certificate was also mentioned as an easy step by many. This part of the process was often facilitated by user-friendly interfaces on our certificate authority website, where students could upload their CSRs and promptly receive the necessary certificate for client authentication with the pass-off server. This process is also facilitated by the browser user interface making it clear whether the certificate was sent to them from the CA.

Pain points Adding the User ID attribute to the CSR and configuring the certificate with web browsers were identified as significant challenges. The lack of straightforward documentation on how to include this specific field using OpenSSL led to confusion and significant delays. One report highlighted that understanding and manipulating OpenSSL's configuration file to add the User ID was not intuitive, requiring students to seek out additional resources and sometimes trial-and-error approaches. Student S1 highlighted these issues in their response.

“There was no information that I could find on a “User ID” attribute for a CSR subject which caused me to go down many wrongpaths like creating a subject extension or trying to manually edit the ANS.1.” (S2)

Most ended up utilizing a configuration file which requires a deeper knowledge of the library being used and is much more challenging to use than the OpenSSL user interface prompts.

Another major challenge reported by students was importing the signed certificate and private key into browsers to complete the access to the pass-off server. Students reported inconsistent experiences across different browsers, with many facing issues in Chrome and ultimately resorting to Firefox or Safari for successful authentication. The necessity to convert certificates to specific formats (i.e. PKCS12) before they were recognized by browsers added an extra layer of complexity and caused additional hurdles for students. Student S1 highlighted this issue in their response.

“The hardest step was to figure out that I need to create a .pfx which would consist of the certificate and the private key. I spent hours with various browsers and different ways to add the certificate to the browser and played around with different file types.” (S3)

This process was fraught with vague error messages and a lack of clear guidance on why a specific certificate was being used or missing, leading to frustration and additional research to troubleshoot these issues.

Resources/Tools used Students utilized a diverse array of resources and tools throughout the first assignment. Predominantly, OpenSSL emerged as a critical tool across the board from many student responses, facilitating the generation of cryptographic key pairs and Certificate Signing Requests (CSRs). Despite its widespread use, students encountered notable challenges, particularly with installation, configuration, and embedding specific fields like the UID. The variability in the utility and accessibility of OpenSSL's documentation and online tutorials underscores the necessity for clear, comprehensive guidance. Our certificate authority site provided a smoother experience for obtaining signed certificates, suggesting that its interfaces and procedural instructions were more user-friendly.

The attempt to import certificates and keys into various web browsers revealed a landscape rife with obstacles for students. Firefox generally offered the least issues compared to Chrome, especially for macOS users. The necessity of converting certificates to browser-compatible formats added an extra layer of complexity to the process. Many relied on online platforms and forums like Stack Overflow highlighting the critical need for current, relevant, and practical advice, though the quality of information varied greatly. A few students explored with more unconventional routes, with mixed results. This exploration; however, pointed to a broader issue: the lack of a one-size-fits-all solution and the consequent need for a deeper understanding of advanced options in current tools.

This varied reliance on tools and resources, coupled with the mixed success across different platforms and environments, brings up the essential need for better standard documentation materials that are not accessible and the need for better tools that can be a one-size-fits-all for key generation and management.

Chapter 6

Key Management Part Two

In this section, we report the quantitative and qualitative results from the second assignment. We start by reporting the perceived usability score for TLS client authentication. Next, we report survey scores on the ease, time invested in completion, and the level of support received throughout the process. Then, we report the amount of time taken to complete the task, as well as the amount of errors encountered along the way. Finally, we report on choices to synchronize or regenerate new certificate credentials when accessing from a different device.

6.1 Quantitative Results

System Usability Scale (SUS) Part two's score trended downward to 45.0 and is rated below the "Not Acceptable" threshold. This shift potentially underscores the evolving challenges and complexities faced by students as they navigated the long-term key management landscape. This is an even lower score compared to other systems listed in the figure indicating that synchronization and accessing from a different device was perceived as even more challenging. This acceptability range and grade corresponds to an adjective rating in the lower bound of "Ok" closer to the upper bound of a "Poor" system usability rating. The distribution of scores can

be visualized in a violin plot in Figure 6.1 showing a wider distribution of ratings between 40-60.

After Senario Questionnaire (ASQ) improvement and increased comfort with the process of obtaining a signed certificate over time. We also asked students in the second assignment to complete a ASQ to gather scores on the ease, time invested in completion, and the level of support received throughout the assignment. The second assignment's survey scores were again visualized in a violin plot showing the distribution amongst reports in Figure 6.2. The median score for the first assignment was high at 5 out of 7. This high score shows that students felt more confident in the process, however, some challenges remained. Some of the reasons for this higher score could be explained by the majority of the students (59%) choosing to repeat the process they did in the first assignment. This indicated improvement and increased comfort with the process of generating a CSR and obtaining a signed certificate.

Time to setup For the second assignment, we only analyzed self-reported times for setup instead of analyzing recorded times. This is because we gave students the option to synchronize or request a new certificate. Our log data would not have recorded times for when a student chose to synchronize their certificate because it logs request submissions. The second assignment's self-reported setup times were again visualized in a violin plot showing the distribution amongst times in Figure 6.3. The median self-reported setup time was 0.39 hours, showing that students were able to complete the second assignment quickly. Combined with the ASQ survey, these results suggest that the students became more comfortable with setting up their access and synchronization was not as time consuming.

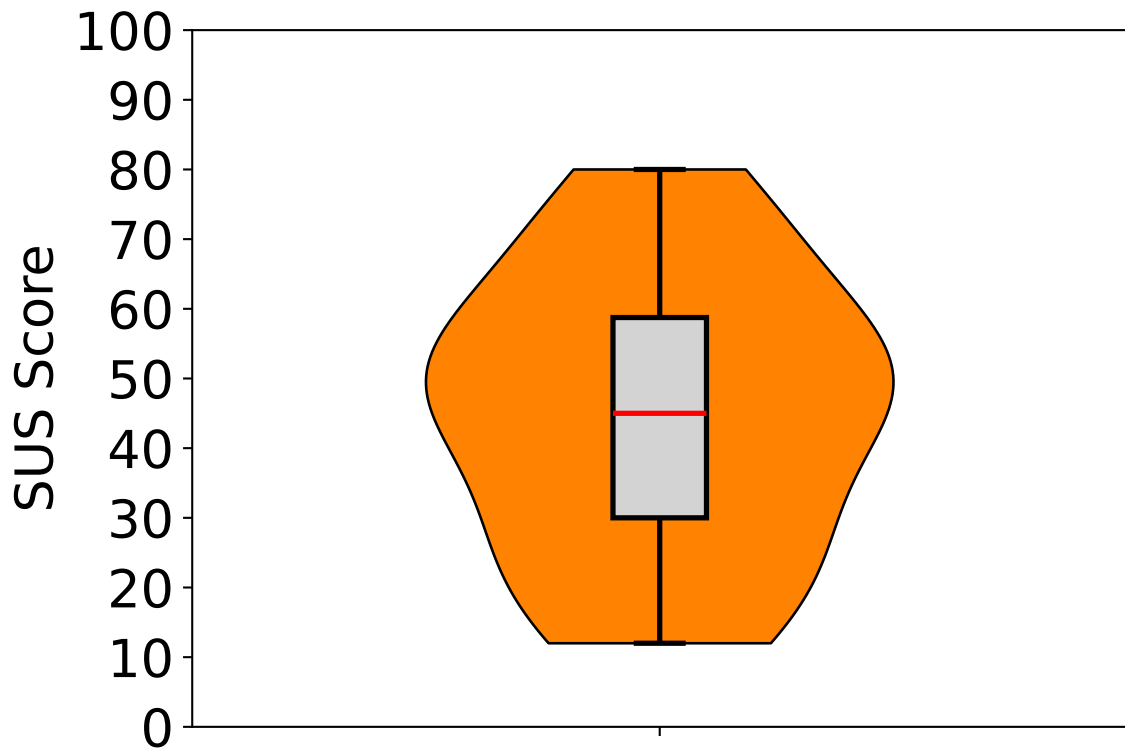


Figure 6.1: Key Management Part Two System Usability Scale (SUS) violin plot shows the distribution of students' scores. This shows a wider distribution of ratings between 40-60, trending downward from the scores received during initial setup.

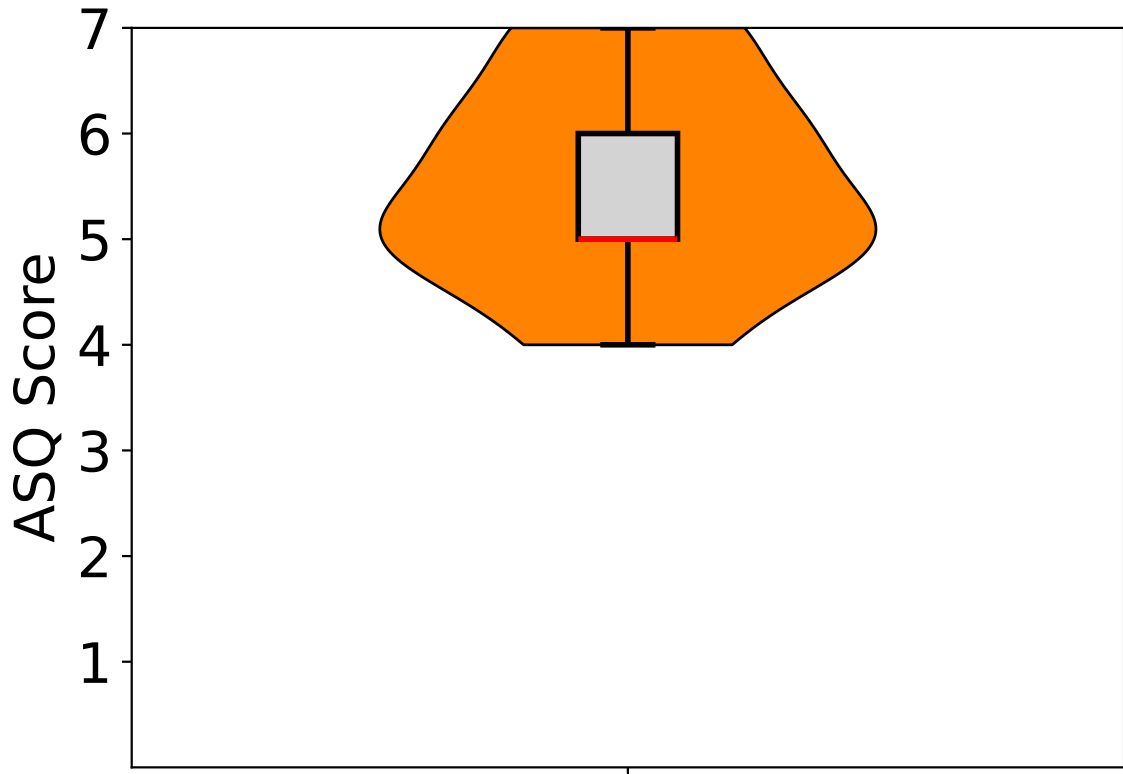


Figure 6.2: Key Management Part Two After Senario Questionnaire (ASQ) score violin plot shows the distribution of students' ratings. The median score for the first assignment was high at 5 out of 7.

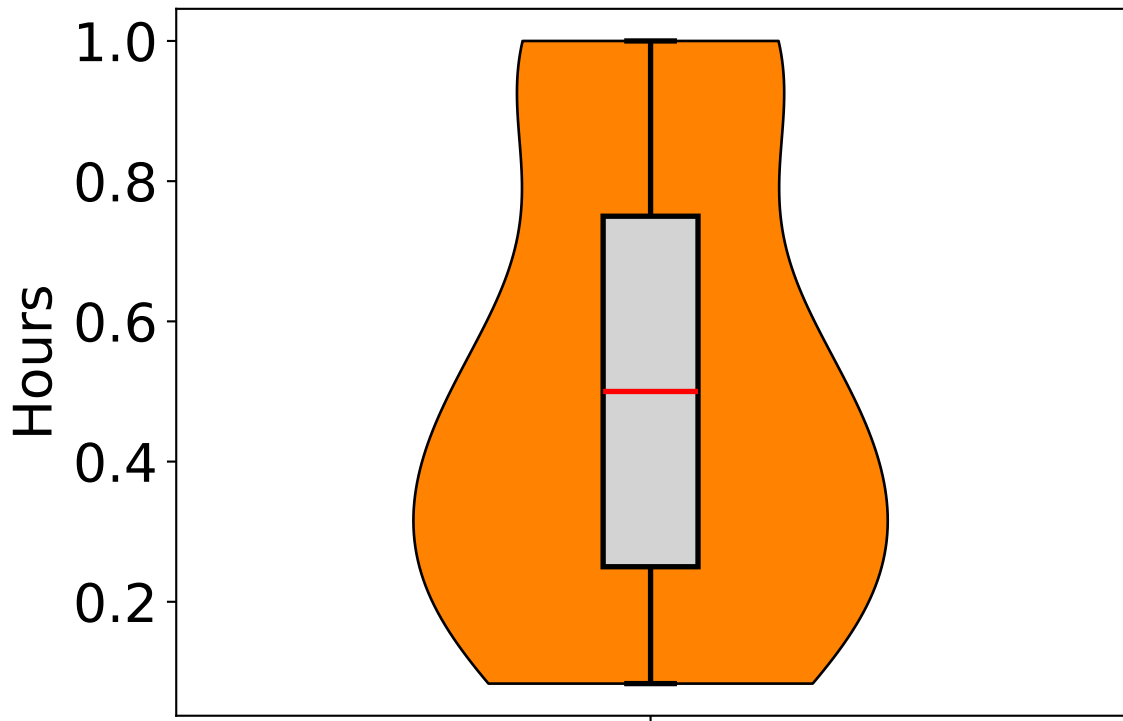


Figure 6.3: Key Management Part Two time to setup violin plot. The median self-reported setup time was 0.39 hours, quicker than the setup times from the first assignment.

6.2 Qualitative Results

What went well? Students reported finding certain aspects of TLS client authentication notably straightforward in the second assignment, reflecting an improvement in their familiarity and skill set with the processes involved. The generation and submission of Certificate Signing Requests (CSRs) to the Certificate Authority (CA) were highlighted as the easiest steps by multiple students. This ease was attributed to the repetitive nature of the task, which reinforced learning from the first assignment. From those who chose to synchronize their certificate, transferring the certificate and private key between devices using secure copy protocol (SCP) or similar methods was mentioned as particularly straightforward, indicating that students were becoming more comfortable with managing key materials and understanding the importance of secure transfer methods.

Pain points Notable challenges reported in the second assignment was the initial setup on new devices, particularly for those who chose to create a new certificate rather than synchronizing an existing one. This choice, while potentially more secure, introduced additional steps and complexities, such as generating a new key pair, submitting a new CSR, and configuring the browser on the new device, all of which required careful attention to detail and adherence to security best practices. One student referred to this process in their writing as easier for them because they have generated keys before, but acknowledged that others might struggle.

“I have generated keys before for other classes. It is hard for me to imagine someone who is not very tech literate coming close to figuring this out.”

(S4)

There were also several students who reported attempting initially to synchronize their certificate, but that they had forgotten their private key password preventing them from reusing their certificate. It is best practice when generating keys to set a secure password on them to prevent unauthorized use even if the key file has

been intercepted. Unfortunately, one of the benefits of password-less authentication method comes with ramifications. The private key is hardly ever touched by the user following its creation. This is because once the signed certificate becomes bundled with the user's private key, the user only has to import that file into their browser initially, allowing all further communication with the web server to seamlessly transpire.

Certificate synchronization In order to complete part two of the project, we instructed the students to access the pass-off server from a new device by either synchronizing their original certificate or by creating a new one by remaking a CSR to the CA. We then asked the students directly which method they chose to access the pass-off server from a new device. 61.9% of the students reported that they generated a new certificate on their second device rather than synchronizing their existing one. Most students explained that their reasoning for not synchronizing was due to being more familiar with generating a CSR and not understanding how they could synchronize a certificate between two different devices. Additionally, student responses to thought exercises showed that many believed synchronization could cause security concerns for the web server.

Experiences using keys Reflecting on the management of private keys and certificates students chose to manage and store them throughout the semester in different ways. One student candidly admitted to keeping them in a generic file folder, acknowledging in hindsight the potential security risks.

“I just kept them in the file where I had all my school stuff in, I am aware this is a bad idea, but at the time I did not think of the importance of always keeping the private keys secure.” (S5)

This acknowledgment not only speaks to the ongoing need for education and awareness around security but also highlights the everyday decisions that impact the security choices of those navigating complex digital environments.

Many students referred to storing their key and certificate in either MacOS keychain or a folder on their computer. When asked if they had to regenerate during the semester the majority had forgotten their password they used to initially create their keys which caused several issues when troubleshooting access problems. Another student described a transition from using a PKCS12 file that initially did not accept their password on a new machine, leading them to transfer the CRT file and private key via SCP to create a new PKCS12 file directly on the new machine. This narrative underscores not only the technical maneuvering of remembering passwords involved but also highlights the reliance on foundational tools like OpenSSL and secure file transfer protocols to maintain the integrity and functionality of their cryptographic assets across different environments.

Thought exercises The student responses to the thought exercise questions in the second assignment demonstrate a range of understandings regarding the scenarios presented. The insights drawn from these responses are crucial in evaluating the depth of students' comprehension of the effectiveness and best security practices of TLS client authentication.

TLS vs. Password Authentication: Many students (20/22) correctly identified the primary advantage of TLS client authentication: it mitigates the risks associated with password theft and phishing by utilizing cryptographic certificates. However, a couple of responses did not respond (2/22), but those who did reflected a misunderstanding or oversimplification of TLS client authentication's benefits, focusing more on the convenience aspect (e.g., not having to enter passwords) rather than security enhancements. This indicates a partial understanding of TLS client authentication's core advantages over password-based systems. One student identified that a drawback of TLS client authentication compared to password-based authentication was the management of a large quantity of certificates.

“The drawback is that certificate management is horrible. It is ludicrously hard to manage large quantities of certificates and issue them and revoke them as needed (properly).” (S6)

Synchronizing vs. New Certificate: All of the students (22/22) recognized the convenience of synchronizing certificates across devices but had varied perceptions of the security implications. They correctly pointed out the security risk associated with synchronizing certificates, especially if one device becomes compromised. Others acknowledged the increased security of generating a new certificate for each device but mentioned the administrative overhead and potential for errors as drawbacks. These responses suggest a general understanding of the security trade-offs between synchronization and generating new certificates, though some lacked depth in considering the comprehensive security landscape.

Stolen/Loss of Certificate or Key: Responses to scenarios involving the loss or theft of certificates or private keys were mixed, with some students (12/22) providing solutions that aligned with security best practices, such as revoking the compromised certificate and generating a new key pair. However, there were also indications of confusion about the correct steps to take, particularly regarding the relationship between certificates and private keys. This variability highlights areas where it may not be immediately obvious to users what steps they should take when they realize a certificate and key pair has been compromised.

Detecting and Responding to Certificate Theft: While some students proposed or mentioned viable methods for detecting certificate theft (19/22), such as monitoring for unusual access patterns, others were unsure of how a Certificate Authority (CA) or server could identify a stolen certificate. The proposed steps to ensure a stolen certificate cannot be used, such as revocation and implementing additional authentication measures, show an understanding of basic concepts but also point to gaps in knowledge about the operational mechanisms behind certificate management and revocation processes. One student seemed unsure about whether certificates

could be revoked highlighting further confusion on what steps should be taken in the event at an anomolous IP logs into a site using a stolen certificate.

“If the user seems to be logging in from a different IP it may alert the server. Again, this certificate can be blacklisted?” (S7)

Overall, the students’ responses to the thought exercises reveal a foundational understanding of TLS client authentication and its security benefits over password-based authentication. However, there is a noticeable variance in the depth of understanding, particularly concerning the management and security implications of certificate synchronization, and the steps to take in response to certificate or key loss/theft. These insights suggest areas where there is a lack of understanding that could lead to security concerns, and what areas of TLS client authentication security may prove to be a challenge when widely implemented.

Chapter 7

Discussion

7.1 Comparing Key Management Part One and Key Management Part Two

The exploration of student experiences with TLS client authentication across two sequential assignments reveals a nuanced landscape of learning, usability challenges, and adaptation. The transition between assignments underscored a pivotal shift in students' engagement with key management, from initial confrontation to a more nuanced interaction with the complexities of cryptographic protocols.

System Usability Scale (SUS) The System Usability Scale (SUS) scores serve as a critical indicator of this journey, reflecting a noticeable disparity in perceived usability. The first assignment, while challenging, introduced students to the fundamentals of TLS client authentication, setting a baseline for usability that was perceptibly diminished in the second assignment. In the first assignment, the SUS score was rated as "Marginal Low" acceptability, which indicated a challenging experience for students engaging with TLS client authentication for the first time. This initial exposure to complex security protocols and key management tasks presented a steep learning curve. The second assignment's semester long

reflection witnessed a further decline in the SUS score, slipping below the "Not Acceptable" threshold. This decrease suggests that while students might have gained familiarity with the process, the additional complexity of managing keys across devices exacerbated usability challenges. To better visualize these scores, we leverage past research from [Bangor et al. \(2008\)](#). This work analyzed 2,324 SUS surveys to gather a range of acceptability ratings that help describe whether a given score is acceptable in terms of usability. The work by Bangor et al. also determined specific SUS scores that correlate to adjectives, helping describe a system's perceived usability. This provides acceptability ratings, adjective ratings, and the SUS score on the same graphic in [Figure 7.1](#). From this figure, we can see that both in part one and part two, the SUS score ratings are far from acceptable and part one barely falls into the "Ok" adjective rating. This highlights there is a lot of work to be done to improve the usability of the TLS client authentication system particularly when having to manage certificate and keys over a long period of time. This comparison implies that improvements to the usability of key management systems over a long period of time would be the most effective in improving the user experience with TLS client authentication and thus the system's security.

After-Scenario Questionnaire (ASQ) Interestingly, the After-Scenario Questionnaire (ASQ) results painted a slightly more optimistic picture of students' experiences in the second part, suggesting a potential reconciliation between perceived difficulty and actual task execution. The ASQ scores in the first assignment highlighted students' struggles with ease, time investment, and perceived support in setting up TLS client authentication, culminating in a median score that reflected dissatisfaction (2.5/7). However, the second assignment's semester long reflection showcased an improvement in ASQ scores (5/7), indicating a higher level of comfort and confidence among students, likely due to accumulated experience and familiarity with the process. This implies that users did find synchronizing and repeating the key generation process much simpler than the initial setup. This also shows how more

automation and better integration can significantly improve a users precieved ease of use for the TLS client authentication system, by removing the burden that caused students the most significant barriers during intially setting up their access.

Time to Setup Time metrics further illuminate this narrative, with a marked reduction in setup times from the first to the second assignment. In Part One, the plot revealed a broad distribution of setup times, indicating a varied range of experiences and challenges faced by students. The extended tails of the distribution suggested that some students encountered significant difficulties, taking much longer to complete the setup. On the other hand, Part Two’s violin plot showed a narrower distribution and a significantly reduced median time, illustrating an overall increase in efficiency and a more uniform experience among participants. This efficiency gain, however, did not translate to a perceived increase in usability, as evidenced by the SUS scores. This again suggests that having more integration and automation allows a user to more quickly and efficiently gain access through TLS client authentication.

7.2 Developers Perspective

After creating the study and CA, we realized that our own perspectives were worth disussing. From our perspective, we also identified several issues with setting up TLS client authentication on the pass-off server that are worth mentioning. For instance, debugging becomes difficult with error messages that lack verbosity. The most common errors when testing the pass-off server were HTTP 500 internal or 401 unauthorized errors. A 401 error is given when there is a problem verifying the client certificate, but there is no description to inform the client what went wrong. This makes trying to deduce whether the certificate has a flaw or if the browser can not find the certificate challenging. Another issue we encountered during the development process was that an obscure browser special flag needs to be set in order to test the web servers with client authentication on a localhost environment. Several times we

received errors when testing our pass-off server's new feature believing something to be wrong with our reverse proxy, just to find out that this browser flag was not set. It even requires manually searching the setting from a browser's search bar, lacking a selectable option in the user interface of a browser's settings. Lastly, the process of client authentication differs depending on the browser chosen. For instance, Firefox requires the client to add the CA to an authorized certificate authority list in their browser settings to connect to any web server using client authentication, but Chrome and Safari does this automatically utilizing the systems trusted CAs. This means that if multiple sites are used, the CA for that signed each sites certificate would need to be added to the trusted CA list which could become tedious if widely adopted.

7.3 Pathways Forward

The study's findings illuminate several pathways forward, each aimed at enhancing the usability of TLS client authentication and key management. The transition towards more user-friendly security systems involves the meticulous design of authentication interfaces that are not only clear and informative but also intuitive for users across varying levels of technical expertise. Simplifying the complex configurations and providing guided processes can significantly mitigate user errors and foster a more secure digital environment.

Automation Automation stands as a potentially pivotal enhancement, particularly for key generation or management tasks. By delegating the generation, renewal, and revocation of keys and certificates to intelligent systems, we can reduce the manual burden on users, thereby minimizing errors and ensuring the consistent application of security measures. This strategy underscores the importance of security systems working silently in the background, requiring user intervention only when absolutely necessary. This could be achieved by a tool that creates a user interface for OpenSSL, a tool many of the students used, to streamline the process of key generation and

certificate request generation. This would reduce the burden of research required to understand which manual commands must be run to generate secure keys.

Integration Furthermore, integrating TLS client authentication mechanisms within platforms and services already familiar to users can streamline the security process. Such integration ensures that secure practices are not an additional task but a seamless part of the user's digital interactions. This familiarity can lead to increased adoption and a more secure user base. One way this could be achieved is at the browser level. Many browsers do not make the process very obvious and do not contain much documentation on how to integrate the use of certificates. Adding a page to the browser settings specifically dedicated to the user of keys and integrating these setting with the machines certificate store could improve the ease of adding certificate credentials to browsers.

Certificate manager In order to increase the adoption and use of certificates and keys to be used in authentication, there exists the need for a usable tool that can effectively inform and assist the user. Many of the students in our study had to rely on system integrated certificate stores to keep track of their credentials which can become tedious for non-technical users. Our study identified that there are key issues that could be addressed by introducing a new tool that would help inform users about the secure use of their certificates and keys. For instance, many of the students in our study mentioned not understanding that the private key and certificate file they received from our CA needed to be bundled into a PKCS 12 file in order to be used within a browser. A certificate manager could be created to assist with creating this bundle automatically by prompting the user to choose which private key they wish to use with their certificate. In addition, the majority of students in our study chose not to synchronize their certificate and key. A certificate manager could allow users to easily import their certificates to the right place across multiple devices. This certificate manager could also solve the issues many students faced of forgetting

their private key password, by implementing some features of a password manager or suggesting the use of a password manager for key passwords.

7.4 Future Work

Future research endeavors should pivot towards the iterative development and empirical evaluation of tools that could be designed to simplify the key management process. Comparative studies examining TLS client authentication and these tools could yield insights into optimal strategies for creating a more user-friendly interface for generating and managing keys. Furthermore, broadening the participant base to include a diverse array of academic disciplines may uncover universal challenges and opportunities within password less authentication methods, providing a more holistic view of the challenges that arise with these forms of authentication. Another avenue for future work is the development of an automated and integrated tool that can perform background tasks to remove the burden away the user of a certificate based authentication methods like TLS client authentication. Some things this tool could include would be automated strong key generation that automatically generates secure keys for users to pair with their signed certificates, and browser integration to automatically bundle a private key with a signed certificate. This would reduce the need for users to understand how to generate strong private keys and significantly improve their security posture. Future studies could also be developed and executed to provide more information on the effect that educating users that alternatives to password-based authentication exist can have on a user's preference to ditch using their passwords. One significant barrier to widespread adoption of certificate-based authentication has been that passwords are king, it would be interesting to see if the everyday person is even aware that there are alternatives.

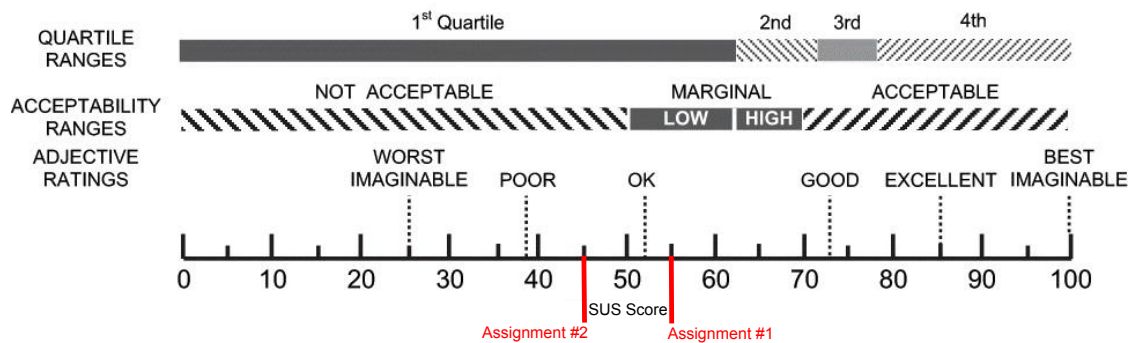


Figure 7.1: Bangor et al. (2008) System Usability Scale (SUS) with quartile ranges, acceptability ranges, and adjective ratings. Assignment #1’s average score was in the first quartile with ”Marginal Low” acceptability and slightly above and ”Ok” adjective rating. Assignment #2’s average score was in the first quartile with ”Not Acceptable” acceptability and in between ”Poor” and ”Ok” adjective ratings.

Chapter 8

Conclusion

This thesis explore the usability challenges of TLS client authentication, shedding light on the experience of future innovators in technology and security as well as the implications of implementing usable security and key management in digital systems. Through a meticulous examination of TLS client authentication practices, this research unveils the complexities involved in securing communications across the internet. It underscored the critical balance between enhancing security protocols and ensuring their usability, which is paramount for widespread adoption of effective cybersecurity measures.

This investigation into TLS client authentication served as a profound first step inquiry into the practical aspects of deploying these certificate-based authentication systems. It highlighted the significant hurdles users face in managing cryptographic keys and certificates, pointing to a pressing need for systems that are not only secure but also accessible to users without extensive technical expertise. Our study highlighted the need for usable security solutions, advocating for ideas that streamline the process of key generation, management, and certificate issuance.

Moreover, our research brought to light the nuances of implementing TLS client authentication across various platforms and devices. It identified key factors that influence the usability of security protocols, including the clarity of documentation,

the simplicity of user interfaces, and the integration of security practices into users' existing workflows. These insights pave the way for future advancements in the field, suggesting a holistic approach to designing security mechanisms that cater to the needs of a diverse user base.

Looking forward, this thesis underscores the importance of further research and development focused on improving the usability of TLS client authentication and similar systems involving long-term key management. It calls for innovative strategies that automate and simplify key management tasks, reducing the cognitive load on users and enhancing the overall security posture of digital systems. By advocating for user-centric design principles in the development of security tools, this research contributes valuable perspectives to the ongoing discourse on creating more secure, usable, and resilient digital infrastructures.

Bibliography

- Atighetchi, M., Soule, N., Pal, P., Loyall, J., Sinclair, A., and Grant, R. (2013). Safe configuration of tls connections. In *2013 IEEE Conference on Communications and Network Security (CNS)*, pages 415–422. IEEE. [2](#)
- Badra, M., Luchuk, A., and Schoenwaelder, J. (2015). Using the netconf protocol over transport layer security (tls) with mutual x.509 authentication. Technical report. [8](#)
- Banday, M. T. and Sheikh, S. A. (2014). S/mime with multiple e-mail address certificates: A usability study. In *2014 International Conference on Contemporary Computing and Informatics (IC3I)*, pages 707–712. [12](#)
- Bangor, A., Kortum, P. T., and Miller, J. T. (2008). An empirical evaluation of the system usability scale. *Intl. Journal of Human-Computer Interaction*, 24(6):574–594. [48](#), [53](#)
- Bendel, M. (2010). Hackers describe ps3 security as epic fail, gain unrestricted access. *Exophase. com*, pages 29–12. [5](#)
- Braeken, A. (2022). Public key versus symmetric key cryptography in client-server authentication protocols. *International Journal of Information Security*, 21(1):103–114. [2](#)
- Campbell, B., Bradley, J., Sakimura, N., and Lodderstedt, T. (2020). Rfc 8705: Oauth 2.0 mutual-tls client authentication and certificate-bound access tokens. [8](#)

- Chen, C.-L., Chiang, M.-L., Hsieh, H.-C., Liu, C.-C., and Deng, Y.-Y. (2020). A lightweight mutual authentication with wearable device in location-based mobile edge computing. *Wireless Personal Communications*, 113:575–598. [2](#)
- Chen, L., Qian, S., Lim, M., and Wang, S. (2018). An enhanced direct anonymous attestation scheme with mutual authentication for network-connected uav communication systems. *China communications*, 15(5):61–76. [8](#)
- Dewanta, F. and Mambo, M. (2019). A mutual authentication scheme for secure fog computing service handover in vehicular network environment. *IEEE Access*, 7:103095–103114. [2](#)
- Dierks, T. (1999). Transport layer security. *RFC 2246*. [8](#)
- Ellison, C. M. (1999). The nature of a useable pki. *Computer Networks*, 31(8):823–830. [10](#)
- Eskandari, S., Clark, J., Barrera, D., and Stobert, E. (2018). A first look at the usability of bitcoin key management. *arXiv preprint arXiv:1802.04351*. [14](#)
- G. Lopes, A. P. and Gondim, P. R. (2020). Mutual authentication protocol for d2d communications in a cloud-based e-health system. *Sensors*, 20(7):2072. [8](#)
- Gallagher, P., Kerry, C., and Romine, C. (2013). Fips pub 186-4: Digital signature standard, dss. *National Institute of Standards and Technology (NIST)*. [5](#)
- John, B. (1996). Sus: a” quick and dirty” usability scale. *Usability evaluation in industry*, pages 189–194. [3](#), [17](#), [18](#)
- Koblitz, N. (1987). Elliptic curve cryptosystems. *Mathematics of computation*, 48(177):203–209. [5](#)
- Lewis, J. R. (1991). Psychometric evaluation of an after-scenario questionnaire for computer usability studies. *ACM SIGCHI Bulletin*, 23(1):78–81. [17](#), [18](#)

- Melki, R., Noura, H. N., and Chehab, A. (2020). Lightweight multi-factor mutual authentication protocol for iot devices. *International Journal of Information Security*, 19:679–694. [8](#)
- Mueller, T. and Michalek, A. (2021). Let’s create! automated certificate management for end-users. In *2021 International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, pages 1–6s. IEEE. [11](#)
- Nguyen, B. M., Dao, T.-C., and Do, B.-L. (2020). Towards a blockchain-based certificate authentication system in vietnam. *PeerJ Computer Science*, 6:e266. [13](#)
- Oppliger, R., Hauser, R., and Basin, D. (2008). Ssl/tls session-aware user authentication. *Computer*, 41(3):59–65. [2](#)
- Pal, O., Alam, B., Thakur, V., and Singh, S. (2021). Key management for blockchain technology. *ICT express*, 7(1):76–80. [13](#)
- Rivest, R. L., Shamir, A., and Adleman, L. (1978). A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126. [5](#)
- Ruoti, S., Andersen, J., Dickinson, L., Heidbrink, S., Monson, T., O’neill, M., Reese, K., Spendlove, B., Vaziripour, E., Wu, J., et al. (2019a). A usability study of four secure email tools using paired participants. *ACM Transactions on Privacy and Security (TOPS)*, 22(2):1–33. [11](#)
- Ruoti, S., Andersen, J., Heidbrink, S., O’Neill, M., Vaziripour, E., Wu, J., Zappala, D., and Seamons, K. (2016). ” we’re on the same page” a usability study of secure email using pairs of novice users. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 4298–4308. [11](#)

- Ruoti, S., Andersen, J., Monson, T., Zappala, D., and Seamons, K. (2018). A comparative usability study of key management in secure email. In *Proceedings of the 14th Symposium on Usable Privacy and Security*. USENIX. [1](#), [11](#)
- Ruoti, S., Kaiser, B., Yerukhimovich, A., Clark, J., and Cunningham, R. (2019b). Sok: Blockchain technology and its potential use cases. *arXiv preprint arXiv:1909.12454*. [13](#)
- Ruoti, S. and Seamons, K. (2019). Johnny’s journey toward usable secure email. *IEEE Security & Privacy*, 17(6):72–76. [1](#), [11](#)

Appendix A

Study Materials

This appendix contains all the materials used to conduct our IRB approved study.

A.1 Informed Consent

Informed Consent

Consent for Research Participation

As part of COSC 466/566—Applied Cryptography, you have been using TLS client authentication to access the passoff server. You will have also written two reports about your experiences. We think that the class's experiences regarding client authentication would be of interest to the research community. As such, we would like to analyze the class's reports and write a research paper exploring what went well and what was challenging about the experience of using TLS client authentication.

To do so, we need your permission to use your reports as part of our research. The following consent form provides you with all the necessary information to provide informed consent.

Research Study Title

An Investigation on the Usability of TLS Client Authentication

Researchers

- Clay Shubert, University of Tennessee, Knoxville
- Scott Ruoti, University of Tennessee, Knoxville

Why am I being asked to be in this research study?

You are being asked to participate in this study as you are a student enrolled in COSC483/583—Applied Cryptography.

What is this research study about?

The purpose of the research study is to better understand the student's experiences utilizing TLS client authentication.

Who is conducting this research study?

This study is being conducted by Dr. Ruoti and Clay Shubert, a TA in your class.

How long will participation in this research study take?

None. We will be using information you have already created for this study.

What will happen if I say “Yes, I want to be in this research study”?

If you agree to be in this study, we will include the reports you wrote for the two Usable Key Management projects in the data we analyze and report on in a research publication. We will anonymize the data before we analyze

it and we will not identify whose data was used in our research paper.

What happens if I say “No, I do not want to be in this research study”?

Being in this study is up to you. You can say no now or choose to exclude your data later. Either way, your decision won't affect your grades, your relationship with your instructors, or your standing with the University of Tennessee, Knoxville.

What happens if I say “Yes” but change my mind later?

Even if you decide to be in the study now, you can change your mind and have your data removed from the study while we are still collecting consent from students. To do so, please contact [Dr. Ruoti](#) and let him know that you do not wish your information to be included. However, after the analysis of the anonymized data begins, it will not be possible to remove your data, as it is not feasible to identify which data is yours.

Are there any possible risks to me?

It is possible that someone could find out you were in this class, but we believe this risk is small because of the procedures we use to protect your information. These procedures are described later in this form.

Are there any benefits to being in this research study?

There is no direct benefit from being in this study. Your participation may help us to learn more about how to improve the usability of TLS client authentication. We hope the knowledge gained from this study will benefit others in the future.

Who can see or use the information collected for this research study?

We will protect the confidentiality of your information by not associating your name with your responses. If information from this study is published or presented at scientific meetings, your name and other personal information will not be used.

We will make every effort to prevent anyone who is not on the research team from knowing that you gave us information or what information came from you. Although it is unlikely, there are times when others may need to see the information we collect about you. These include:

- People at the University of Tennessee, Knoxville who oversee research to make sure it is conducted properly.
- Government agencies (such as the Office for Human Research Protections in the U.S. Department of Health and Human Services), and others responsible for

watching over the safety, effectiveness, and conduct of the research.

- If a law or court requires us to share the information, we would have to follow that law or final court ruling.

What will happen to my information after this study is over?

We will not keep your personal information to use for future research. Your name and other information that can directly identify you will be deleted from your research data collected as part of the study. We may share your research data with other researchers without asking for your consent again, but it will not contain information that could directly identify you.

Will I be paid for being in this research study?

You will not be paid for participation in this study.

Who can answer my questions about this research study?

If you have questions or concerns about this study, or have experienced a research-related problem or injury, contact Dr. Ruoti (ruoti@utk.edu).

For questions or concerns about your rights or to speak with someone other than the research team about the study, please contact:

Institutional Review Board
The University of Tennessee, Knoxville
Phone: 865-974-7697
Email: utkirb@utk.edu

By selecting the radio button below and clicking submit, you acknowledge:

- Your participation in the study is voluntary.
- You are 18 years of age.
- You are aware that you may choose to terminate your participation at any time for any reason.

I consent, begin the study

Powered by Qualtrics

A.2 Usable Key Management (Part One)

Project: Usable Key Management (Part 1)



PROJECT: USABLE KEY MANAGEMENT (PART 1)

In this project, you will be exposed to TLS client authentication, also referred to as mutual TLS or mTLS. This will give you first-hand experience with a modern, real-world cryptosystem.

Requirements

To use the pass-off server, you will need to authenticate using TLS client authentication. TLS client authentication works by having the client (your browser) send a certificate to the server along with a digital signature generated using the private key associated with the certificate. When receiving this data, the server checks that the certificate is signed by a source it trusts and that the digital signature is valid.

To complete the first part of the project and gain access to the pass-off server, you will need to take the following steps:

1. Create your cryptographic key pair.
2. Obtain a signed certificate for your key pair ([using this website](https://appliedcrypto.userlab.utk.edu/) [\(https://appliedcrypto.userlab.utk.edu/\)](https://appliedcrypto.userlab.utk.edu/)).
3. Register the signed certificate and your private key with your browser.

After successfully completing these steps, you can log into the [pass-off website](https://appliedcrypto.userlab.utk.edu/) [\(https://appliedcrypto.userlab.utk.edu/\)](https://appliedcrypto.userlab.utk.edu/). You will know everything is working when you can visit this website without receiving an error. After you have completed the above steps and gained access to the pass-off server, write a report describing your experience. This report should include the following details:

- How long did it take you to complete this project?
- What steps did you take to complete the project?
 - Also, include details about anything you attempted that ultimately did not work.
 - *Include screenshots if you think that would be helpful.*
- Describe what information sources you used and how helpful (or unhelpful) they were.
- Identify the tools you used.
 - Also, include details about any tools you ultimately abandoned.
 - Describe what went well with these tools and what was challenging.
- Answer the following questions:

- What were the easiest one or two steps in setting up TLS client authentication? Why were they the easiest?
 - What were the hardest one or two steps in setting up TLS client authentication? Why were they the hardest?
 - What would you change about the setup process, if anything?
 - Answer the after-scenario questionnaire (ASQ) by indicating how much you agree with the following statements on a scale of 1 (strongly disagree) to 7 (strongly agree).
 - Overall, I am satisfied with the ease of setting up TLS client authentication.
 - Overall, I am satisfied with the amount of time it took to set up TLS client authentication.
 - Overall, I am satisfied with the support information (online help, messages, documentation) I found when setting up TLS client authentication.
 - Answer the system usability scale (SUS) questions by indicating how much you agree with the following statements on a scale of 1 (strongly agree) to 5 (strongly disagree). **Note that this scale is different than the previous scale.**
 - I think that I would have no problem setting up TLS client authentication frequently.
 - I found TLS client authentication unnecessarily complex to set up.
 - I thought that setting up TLS client authentication was easy.
 - I think that I would need the support of a technical support staff to set up TLS client authentication in the future.
 - I found the various functions for setting up TLS client authentication to be well-integrated.
 - I thought there was too much inconsistency in setting up TLS client authentication.
 - I would imagine that most people would learn to set up TLS client authentication very quickly.
 - I found setting up TLS client authentication to be very cumbersome.
 - I felt very confident setting up TLS client authentication.
 - I needed to learn a lot of things before I could get going with setting up TLS client authentication.
 - Provide any other feedback you have about the project or setting up TLS client authentication
-

Getting Started and Getting Help

As the purpose of this project is to give you experience with a real-world cryptosystem, neither the instructor nor the TAs will tell you how to complete these tasks. **However, you are free to use any online information source you want.** You are also free to use whatever tools you wish to complete this project.

For the learning purposes of this project, avoid asking other students how to complete the project. I recognize that at times the lack of guidance may be frustrating. That is by design. The purpose of this project is to give you experience with using a cryptosystem as they are deployed in the real world.

If after ninety minutes (1.5 hours) you feel stuck on this project, reach out to the TAs for some tips. If after three hours you haven't completed this project, let the TAs know and they will help you complete it so that you can have access to the grading server. In either of the two proceeding cases, mention what help you got from the TAs when writing your report and you can still receive full credit.

Grading Rubric

- 25 points for a well-written description of how you set up client authentication. This must include the steps you took as well as the tools and information sources you used (whether successfully or unsuccessfully).
- 15 points for answering the three questions about what was easiest and hardest about setting up client authentication and what you would change.
- 5 points for answering the SUS questions.
- 5 points for answering the ASQ questions.

Submission

Submit your written report as a PDF file to Canvas.

Click **Next** to continue.

Points 50

Submitting a file upload

File Types pdf

Due	For	Available from	Until
Sep 28, 2023	Everyone	-	-

A.3 Usable Key Management (Part Two)

Project: Usable Key Management (Part 2)




PROJECT: USABLE KEY MANAGEMENT (PART 2)

Throughout the semester, you have been using TLS client authentication to log into the passoff server. In this project, you will be reflecting on that experience. You will also get a chance to experience what is necessary to sync client certificates between multiple machines.

Requirements

In this project, you will access the pass-off server from a new device you haven't used to access the server previously. More specifically:

1. If you have been using a personal computer (i.e., not a Tesla or Hydra machine) to complete the projects, you will need to access the pass-off server using a Tesla or Hydra machine; **OR**
2. If you have been using a Tesla or Hydra machine to complete the project, you will need to use a personal computer (i.e., not a Tesla or Hydra machine) to access the pass-off server.

Once you have access to the pass-off server on the new machine, you will use this [pass-off link](https://appliedcrypto.userlab.utk.edu/cert_sync/)  (https://appliedcrypto.userlab.utk.edu/cert_sync/) to verify that you are accessing the pass-off server from a new machine. Next, you will write a report describing your experiences. You will also reflect on your experience using TLS client authentication throughout the semester. Finally, you will complete several thought problems regarding the security of TLS client authentication. The report should have the following contents:

- **Section 1—Multi-device TLS client authentication**
 - How long did it take you to complete this project?
 - What steps did you take to complete the project?
 - Also, include details about anything you attempted that ultimately did not work.
 - *Include screenshots if you think that would be helpful.*
 - Describe what information sources you used and how helpful (or unhelpful) they were.
 - To complete this project, you could either synchronize your existing certificate to the new machine or get a new certificate issued for that machine. Why did you choose the approach that you ultimately used?
 - Answer the after-scenario questionnaire (ASQ) by indicating how much you agree with the following statements on a scale of 1 (strongly disagree) to 7 (strongly agree).

- Overall, I am satisfied with the ease of setting up TLS client authentication on a second device.
 - Overall, I am satisfied with the amount of time it took to set up TLS client authentication on a second device.
 - Overall, I am satisfied with the support information (online help, messages, documentation) I found when setting up TLS client authentication on a second device.
 - Answer the following questions:
 - What were the easiest one or two steps in setting up TLS client authentication on a second device? Why were they the easiest?
 - What were the hardest one or two steps in setting up TLS client authentication on a second device? Why were they the hardest?
 - What would you change about the process for setting up TLS client authentication on a second device?
 - Provide any other feedback you have about the project or setting up client authentication on a second device.
- **Section 2—Semester-long reflection**
 - *For this section, please reflect on your experience using TLS client authentication to access the pass-off server during the semester.*
 - What browser did you use when accessing the pass-off server? What version is it currently running?
 - What steps did you take to authenticate using TLS client authentication when accessing the pass-off server?
 - How did you manage your signed certificate file and private key throughout this semester?
 - If you had to regenerate one or both of these items, please describe your experience doing so.
 - Answer the system usability scale (SUS) questions by indicating how much you agree with the following statements on a scale of 1 (strongly agree) to 5 (strongly disagree). **Note that this scale is different than the previous scale.**
 - I think that I would have no problem using TLS client authentication frequently.
 - I found TLS client authentication unnecessarily complex.
 - I thought that TLS client authentication was easy.
 - I think that I would need the support of a technical support staff to use TLS client authentication in the future.
 - I found the various functions for TLS client authentication to be well-integrated.
 - I thought there was too much inconsistency in TLS client authentication.
 - I would imagine that most people would learn to use TLS client authentication very quickly.
 - I found TLS client authentication to be very cumbersome.
 - I felt very confident using TLS client authentication.
 - I needed to learn a lot of things before I could get going with TLS client authentication.
 - Answer the following questions:

- What was the easiest one to three things about using TLS client authentication? Why were they the easiest?
 - What were the hardest one to three things about using TLS client authentication? Why were they the hardest?
 - What would you change about using TLS client authentication on a second device?
 - How did the usability of setting up TLS client authentication compare to the usability of using TLS client authentication throughout the semester?
 - Provide any other feedback you have about the project or TLS client authentication on a second device.
- **Section 3—Thought exercises**
- *In this section, you complete several thought exercises. **You will receive full credit for whatever you put down.** I'm not looking for you to get the "right" answer, but rather for you to share your thoughts based on what you've learned this semester and your experiences using TLS client authentication.*
 - Are there any security benefits or drawbacks to using TLS client authentication as compared to password-based authentication? If so, what are they?
 - To use TLS client authentication from two devices, you could either synchronize your existing certificate to the new machine or get a new certificate issued for that machine. Each of these approaches has different potential security implications. What do you think the security benefits or drawbacks of each approach are?
 - If you were to **lose** your TLS client authentication signed certificate, but not your private key, what would you need to do to regain access to the pass-off server? Are there any security concerns with your proposed workflow? If so, how could they be addressed (*you can answer that you don't know*)?
 - If you were to **lose** your TLS client authentication private key, what would you need to do to regain access to the pass-off server? Are there any security concerns with your proposed workflow? If so, how could they be addressed (*you can answer that you don't know*)?
 - If you were to have your TLS client authentication private key **stolen**, what would you need to do to regain access to the pass-off server? Are there any security concerns with your proposed workflow? If so, how could they be addressed (*you can answer that you don't know*)?
 - Suppose that the certificate you received this semester gave you access to a secure database containing your assignments and labs. Suppose this database has a lateral movement vulnerability that an adversary, Mallory, has identified allowing her to edit grades as the admin user. Mallory's goal is to intercept a student's certificate which would allow her to impersonate that user and gain access to the database where she can exploit this vulnerability. How could the CA and/or pass-off server identify that a certificate has been stolen? In addition, what steps should the CA take to ensure that this certificate cannot be used?
-

Getting Started and Getting Help

As the purpose of this project is to give you experience with a real-world cryptosystem, neither the instructor nor the TAs will tell you how to complete this task. **However, you are free to use any online information source you want.** You are also free to use whatever tools you wish to complete this project.

Tesla and Hydra lab schedule

To complete this project, you will need to access the browser on a Tesla or Hydra machine. While this can be done over SSH with X forwarding, for most students, it will be easier to complete this project physically at a Tesla or Hydra machine. Below, you can find the schedule for the Tesla and Hydra labs. Use this to find a time when the lab is free to complete the first part of the project.

Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
September 24	25	26	27	28	29	30
	<p>8:00 am - 9:35 am COSC 302 001 MMP BG 42636 202340 <i>M 0800-0935 LAB 08/28</i> </p> <p>12:00 pm - 1:35 pm COSC 302 002 MMP BG 42637 202340 <i>M 1200-1335 LAB 08/28</i> </p> <p>12:00 pm - 1:35 pm COSC 307 001 MMP BG 48133 202340 <i>M 1200-1335 LAB 08/28</i> </p> <p>2:00 pm - 3:35 pm COSC 302 003 MMP BG 47781 202340 <i>M 1400-1535 LAB 08/28</i> </p> <p>2:00 pm - 3:35 pm COSC 307 002 MMP BG 48522 202340 <i>M 1400-1535 LAB 08/28</i> </p>		<p>10:00 am - 10:50 am COSC 202 005 MMP BG 63630 202340 <i>W 1000-1050 LAB 08/23</i> </p> <p>11:00 am - 11:50 am COSC 202 006 MMP BG 63952 202340 <i>W 1100-1150 LAB 08/23</i> </p> <p>12:00 pm - 12:50 pm COSC 202 007 MMP BG 65741 202340 <i>W 1200-1250 LAB 08/23</i> </p> <p>1:00 pm - 1:50 pm COSC 202 008 MMP BG 66560 202340 <i>W 1300-1350 LAB 08/23</i> </p>		<p>8:00 am - 8:50 am COSC 102 001 MMP BG 42631 202340 <i>F 0800-0850 LAB 08/25</i> </p> <p>9:00 am - 9:50 am COSC 102 002 MMP BG 42632 202340 <i>F 0900-0950 LAB 08/25</i> </p> <p>11:00 am - 11:50 am COSC 102 003 MMP BG 42633 202340 <i>F 1100-1150 LAB 08/25</i> </p> <p>12:00 pm - 12:50 pm COSC 102 004 MMP BG 42634 202340 <i>F 1200-1250 LAB 08/25</i> </p> <p>1:00 pm - 1:50 pm COSC 102 005 MMP BG 42635 202340 <i>F 1300-1350 LAB 08/25</i> </p>	
Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
September 24	25	26	27	28	29	30
	<p>8:00 am - 8:50 am COSC 202 001 MMP BG 53047 202340 <i>M 0800-0850 LAB 08/28</i> </p> <p>9:00 am - 9:50 am COSC 202 002 MMP BG 53048 202340 <i>M 0900-0950 LAB 08/28</i> </p> <p>10:00 am - 10:50 am COSC 202 003 MMP BG 53049 202340 <i>M 1000-1050 LAB 08/28</i> </p> <p>11:00 am - 11:50 am COSC 202 004 MMP BG 53050 202340 <i>M 1100-1150 LAB 08/28</i> </p>		<p>8:00 am - 8:50 am COSC 360 001 MMP BG 42639 202340 <i>W 0800-0850 LAB 08/23</i> </p> <p>8:00 am - 8:50 am COSC 367 001 MMP BG 56124 202340 <i>W 0800-0850 LAB 08/23</i> </p> <p>9:00 am - 9:50 am COSC 360 002 MMP BG 45265 202340 <i>W 0900-0950 LAB 08/23</i> </p> <p>9:00 am - 9:50 am COSC 367 003 MMP BG 56909 202340 <i>W 0900-0950 LAB 08/23</i> </p> <p>10:00 am - 10:50 am COSC 360 003 MMP BG 51225 202340 <i>W 1000-1050 LAB 08/23</i> </p> <p>11:00 am - 11:50 am COSC 360 004 MMP BG 52614 202340 <i>W 1100-1150 LAB 08/23</i> </p> <p>11:00 am - 11:50 am COSC 367 002 MMP BG 56125 202340 <i>W 1100-1150 LAB 08/23</i> </p> <p>12:00 pm - 12:50 pm COSC 360 005 MMP BG 52678 202340 <i>W 1200-1250 LAB 08/23</i> </p>			

Grading Rubric

- 10 points for completing the first part of the report.
- 20 points for completing the second part of the report.
- 20 points for completing the third part of the report.

Throughout, the report should be well-written, answering the questions listed in the requirements section. Also, if you do not setup TLS client authentication on a second machine, at most you can receive half points for the report.

Submission

Submit your written report as a PDF file to Canvas.

Click **Next** to continue.

Points 50

Submitting a text entry box or a file upload

File Types pdf

Due	For	Available from	Until
Dec 6, 2023	Everyone	-	-

Appendix B

User Interfaces

This appendix contains all the user interfaces that students interacted with during our IRB approved study.

B.1 Certificate Authority

T Applied Cryptography Certificate Authority

This site will help you get a signed certificate for logging into <https://appliedcrypto.userlab.utk.edu/>. To do this, you will need to upload a certificate signing request (CSR). You are welcome to using any tools you wish to create the CSR.

For your CSR to be approved, it must meet the following requirements:

- The cryptographic key must have at least 128-bit equivalent security.
- The certificate must have the following values as described below:
 - User ID: Your netId
 - Common Name: Your name
 - Organization: University of Tennessee
 - Organizational Unit: Department of Electrical Engineering and Computer Science
 - Locality: Knoxville
 - State: Tennessee
 - Country: US

Keep in mind that the Certificate Authority (CA) will not be forgiving of errors. You need to ensure that you use the correct capitalization and abbreviations or else the request will be rejected. Special characters should not be present.

Upload your Certificate Signing Request (CSR)

After uploading a valid CSR, you will be prompted to download a signed certificate. You will need this certificate **and** your private key for client authentication.

CSR file	Choose file	Browse
<input type="button" value="Sign CSR"/>		

B.2 Pass-off Server

T Applied Cryptography Pass-Off Server

Project: MAC Attack

Project: Diffie-Hellman

Project: RSA

Project: SRP

Project: Blockchain

Project: Usable Key Management (Part 2)

B.3 Pass-off Server Lab Page Example

Project: MAC Attack

Instructions

To complete this project, you will need to generate a legitimate MAC for a malicious message that extends the following message:

```
No one has completed Project #3 so give them all a 0.
```

The MAC of this message is: `d32bcec2a2a064b6c38ce3c78b2523a9b794a999`

The HMAC method is implemented as `HMAC(secret, m)=sha1(secret || m)`. Encode the message using ASCII.

Submission

Modified message (hex)

Modified MAC (hex)

Submit

Vita

Clay Shubert was born in Philadelphia, Pennsylvania, and shortly moved to Nashville, Tennessee, where he still currently resides. He is a graduate student at the University of Tennessee Knoxville Tickle College of Engineering, pursuing a Master of Science in Computer Engineering with focuses on Cybersecurity and Datacenter Management and Technology. He is expected to graduate in May 2024. Before embarking on his graduate studies, he completed his Bachelor of Science in Computer Engineering at the same institution, minoring in Cybersecurity and Datacenter Management and Technology between 2019 and 2023. Clay's has a few certifications as a LogRhythm Security Analyst and as an ISC2 Candidate, alongside a proficiency in Spanish at a limited working level.

Professionally, Clay has recently accepted a role as a Security Engineer at Mercury Systems in Huntsville, Alabama, where he is responsible for enhancing cybersecurity measures and solutions for the U.S. Aerospace and Defense industry. His journey in the realm of cybersecurity began with a significant tenure as a Intern Cyber Security Analyst at Avertium in Knoxville, Tennessee, from May 2021 to December 2023. In this role, Clay honed his skills in investigating alarms across various Security Information and Event Management (SIEM) systems, escalating malicious activity, and performing immediate mitigation for numerous customers.

Beyond his professional experiences, Clay has been actively involved in the academic community as a Graduate Teaching Assistant at the Tickle College of Engineering, University of Tennessee, since August 2023. His commitment to

cybersecurity extends into extracurricular activities, where he is a member of HackUTK at the University of Tennessee. This involvement allowed him to compete at the U.S. Department of Energy, Cyberforce Competition at Oak Ridge National Lab in 2019.

Clay's notable skills and certifications include proficiency in programming languages such as Python, C/C++, VHDL, and Java, and experience with SIEMs including Microsoft Sentinel One, Alienvault USM Central/Appliance, Fortinet, and LogRhythm. Driven by a passion for applying his skills and knowledge in the field of cybersecurity, Clay is dedicated to making a positive impact through his work and research.