Exploring Password Authenticated Key Exchange (PAKE) Protocols: A Usability Evaluation

Thomason Zhao
UW–Madison

Thomas Peng
UW–Madison

ABSTRACT

This proposal outlines the development and evaluation of a privacy-preserving authentication system leveraging zero-knowledge proofs to authenticate users without revealing any personally identifiable information.

1 INTRODUCTION

The widespread reliance on password-based authentication has long been a contentious topic in the cybersecurity landscape. Traditional password-based authentication systems are susceptible to various attacks, such as password guessing, dictionary attacks, and credential stuffing, which can compromise user accounts and expose sensitive information. To address these vulnerabilities, researchers have developed more robust authentication methods, with password-authenticated key exchange (PAKE) protocols emerging as a promising solution.

PAKE protocols are designed to establish a secure session key between a client and a server without revealing the user's password, even in the face of an adversary with extensive knowledge of the system. These protocols leverage cryptographic techniques to ensure that after a login attempt, whether valid or invalid, the client and server only learn whether the password matched the expected value, without leaking any additional information. This property is particularly valuable in threat models where adversaries may have access to password leaks, password distributions, and substantial computational power.

The primary goal of this project was to evaluate the usability of different PAKE protocols, assessing their performance, scalability, and user experience. By conducting a comprehensive analysis, we aimed to provide insights into the practical implications of adopting PAKE protocols in real-world authentication systems, addressing the tradeoffs between security, efficiency, and user-friendliness.

To achieve this objective, we implemented three PAKE protocols - Password Over TLS, Secure Remote Password (SRP), and OPAQUE - within a web-based application framework utilizing React.js for the front-end, Node.js for the back-end, and MongoDB for the database. We then performed a series of experiments to measure the protocols' performance and scalability, focusing on metrics such as CPU usage, memory consumption, and storage requirements. Additionally, we conducted a user experience study to gather feedback on the usability, trustworthiness, and perceived safety of the PAKE protocols from a diverse group of participants.

The findings from this project contribute to the understanding of the practical considerations surrounding the deployment of PAKE protocols in real-world settings. By evaluating the tradeoffs between the security benefits and the system-level impact of these protocols, we hope to inform decision-makers and system designers on the feasibility and trade-offs of incorporating PAKE-based authentication into their applications, ultimately enhancing the overall security posture while maintaining a positive user experience.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

2 BACKGROUND AND RELATED WORK

Passwords remain the most common form of user authentication on the internet today. However, traditional password-based authentication systems have long been a point of concern due to their inherent vulnerabilities. Passwords can be easily guessed, stolen, or compromised through a variety of attacks, exposing user accounts and sensitive information to unauthorized access. This has prompted the research and development of more robust authentication methods that can better protect user credentials while maintaining a positive user experience.

2.1 Password-Authenticated Key Exchange (PAKE) Protocols

Password-authenticated key exchange (PAKE) protocols are a class of cryptographic techniques designed to establish a secure session key between a client and a server without explicitly revealing the user's password. These protocols leverage the user's password as the primary authentication factor, while ensuring that the password itself is not exposed during the authentication process. PAKE protocols are particularly useful in scenarios where traditional password-based authentication mechanisms are vulnerable to offline attacks, such as password guessing, dictionary attacks, and credential stuffing.

The core idea behind PAKE protocols is to allow the client and server to mutually authenticate each other and derive a shared session key, without the server ever learning the client's password. This is achieved through a series of cryptographic operations that ensure the password is only used as a key to unlock the authentication process, rather than being directly transmitted or stored on the server. By preserving the confidentiality of the password, PAKE protocols offer a higher level of security compared to traditional password-based authentication mechanisms.

In a typical PAKE protocol, the client provides their username and password to the server, and the server responds with a challenge or some form of cryptographic information. The client and server then engage in a series of message exchanges, performing various computations and verifications to establish the shared session key. At the end of the protocol, both the client and server are confident that the other party has successfully authenticated, without revealing the password to either party.

2.2 Threat Model and Research Goal

The primary threat model considered in this project assumes that all adversaries have access to any exposed information, such as password leaks or password distributions, as well as some computational power, such as the ability to precompute password hashes. This threat model reflects the reality of modern cybersecurity challenges, where attackers often possess significant resources and knowledge about the target system.

Within this threat model, we identified two main adversarial roles:

- **Adversarial Client:** A malicious client that attempts to gain unauthorized access by exploiting any vulnerabilities in the authentication process.
- **Adversarial Server:** A malicious server that tries to learn a user's identity and credentials by manipulating the authentication process or storing sensitive information.

The research goal of this project was to evaluate the usability of different PAKE protocols, assessing their performance, scalability, and user experience. By conducting a comprehensive analysis, we aimed to provide insights into the practical implications of adopting PAKE protocols in real-world authentication systems, addressing the tradeoffs between security, efficiency, and user-friendliness.

2.3 Existing PAKE Protocols

To achieve our research goal, we selected three PAKE protocols to evaluate:

- Password Over TLS: This protocol leverages the Transport Layer Security (TLS) protocol to encrypt the communication between the client and server, ensuring that the password is transmitted securely. The server then checks the user's password by comparing it to the stored value in the database.
- Secure Remote Password (SRP): SRP is a PAKE protocol that allows the client and server to authenticate each other and establish a shared session key without explicitly revealing the password. The protocol involves a series of computations and message exchanges to verify the password without transmitting it in plain text.
- OPAQUE: OPAQUE is a more recent PAKE protocol that utilizes an oblivious pseudorandom function (OPRF) to perform the password-based key exchange. This protocol aims to provide stronger security guarantees and protect against a wider range of attacks, such as offline dictionary attacks and server compromise.

By evaluating these different PAKE protocols, we aimed to gain a comprehensive understanding of the tradeoffs between their usability, performance, and security characteristics, ultimately providing insights to guide the adoption of PAKE-based authentication in real-world applications.

3 EVALUATION PLAN

To rigorously assess the proposed system, we will employ a multi-dimensional evaluation plan encompassing qualitative and quantitative metrics.

3.1 Evaluation Metrics

- **Authentication Success Rate (ASR)**: The ratio of successful authentications to the total authentication attempts.
- **Authentication Time**: Duration from the initiation to completion of the authentication process.
- **System Efficiency**: Computational and memory resources utilized during authentication.
- **Security Analysis**: System robustness against common attack vectors.
- **User Satisfaction**: User experience assessed through surveys and interviews.
- **Scalability**: The system's ability to handle increased load effectively.

3.2 Data Collection

Data will be collected through system logs, user feedback, and security incident reports to provide a comprehensive evaluation of the system's performance.

3.3 Evaluation Methodology

A multi-stage evaluation approach, including laboratory testing, field testing, A/B testing, security auditing, and scalability testing, will be utilized to ensure a thorough assessment.

4 MILESTONE

The project will follow a series of milestones to ensure structured progress towards the system's development and evaluation.

4.1 Milestone 1: System Design and Initial Development

- Deadline: 2024-03-04
- Goals:
  - Complete a detailed literature review.
  - Finalize the system architecture.
  - Begin development of the authentication prototype.

4.2 Milestone 2: Prototype Testing and Iteration

- Deadline: 2024-04-05
- Goals:
  - Complete the prototype development.
  - Conduct initial internal testing and debugging.
  - Gather early feedback and iterate on the prototype.

4.3 Milestone 3: System Evaluation and Finalization

- Deadline: 2024-04-24
- Goals:
  - Perform comprehensive system testing, including security and performance evaluations.

- Finalize the user interface based on usability testing feedback.
- Prepare the project report and documentation.

REFERENCES