Public Authenticated Key Establishment

Project Proposal and Plan

Matthew Robinson - Samuel Wang

September 16, 2021

1 Research Problem

Password Authenticated Key Establishment (PAKE) is a secure, useful technology that doesn't see enough adoption. We'd like to create a simple text based chat application where PAKE is used to allow 2 parties, who know the same password, to establish a secure channel over an insecure network. Following the creation of this simple application, we'd like to perform various types of cryptanalysis to see if the communication can be decrypted, or if any information can be gleaned from the key.

2 Background and Related Work

2.1 PAKE

Wikipedia defines Password-Authenticated Key Exchange as follows:

"Password authenticated key exchange (PAKE) is where two or more parties, based only on their knowledge of a shared password,[1] establish a cryptographic key using an exchange of messages, such that an unauthorized party (one who controls the communication channel but does not possess the password) cannot participate in the method and is constrained as much as possible from brute-force guessing the password. (The optimal case yields exactly one guess per run exchange.) Two forms of PAKE are Balanced and Augmented methods."

What this means, is that key exchange protocol like Diffie-Hellman (DH) can be used with an added layer of security. A password can be used for authentication and key exchange, whereas previously, a protocol like DH would be vulnerable to man-in-the-middle attacks due to it's trusting nature.

PAKE benefits can include the fact that an encrypted channel can be established first without giving an entry password over a non-encrypted channel. This provides significant protection as it can prove a severe hindrance to man-in-the-middle attacks. While it may be possible to steal a plaintext password, once the encrypted channel goes up, it would be very difficult to steal a password without having direct access to one or more of the machines or keys.

2.2 PAKE Cryptanalysis

Information on PAKE crypanalysis is seemingly scarce. However, we've found a paper from 2004 (linked below), that provides a cryptanalysis technique on two different PAKE protocols, basing their attack on "the small factors of the order of a large group $Z*_P$". Using this technique, the author's are able to demonstrate the use of dictionary attacks to discover the secret password. Careful examination of this paper's methodology and techniques will be done at a later time, as we're currently blocked from viewing the paper on our home networks (due to paywalls).

More research will be done to find other means of PAKE cryptanalysis (see our approach section).

3 Approach

We will demonstrate password authenticated key exchange by creating a chat program with encryption that requires a password in order to establish a pair of cryptographic keys. A wrong password would result in the program terminating. In a real world scenario, much nastier things would probably be happening to the IP, machine, and guy on the other end but in this case just demonstrating that a consequence can happen is good enough.

We'd like to implement the following multi-step approach:

3.1 Research

Before we can implement our simple text chat application, underlying PAKE protocol, or perform cryptanalysis on it, we need to obtain some foundational knowledge. Currently we have a number of sources including tech blogs and research papers obtained via google scholar (filtered for peer reviewed works). There are also resources available in the NMT Skeen Library online database.

3.2 Implementation

After performing our research, we should have the ability to implement our simple text based chat application using PAKE for a secure channel. It should be emphasized here that our chat app is just a means to an end, and the majority of our time and effort spent in implementation should be focused on the PAKE implementation and refinement.

Our initial implementation approach is to use Python to create our simple text based chat application. A very bare-bones infrastructure will be used, where we create a Python server that will perform the following operations:

- Listen for client connections
- Create threads for each connected client

And a client that performs the following operations:

- Connects to the server
- Listens for messages coming from the server
- Accepts user input to be sent to the server

With this as a starting point, we can begin to apply the PAKE protocol to our chat application. Again, this is where we want to spend most of our energy and time. Ideally, the chat application portion shouldn't take long to implement. Implementing PAKE from scratch, and possibly adding in different PAKE modes, is what we want to focus on.

3.3 Experimentation Analysis

Once our system is implemented, we can begin to perform cryptanalysis on it based on any additional methods that we obtain through research or from the class later in the semester.

With implementation and experimentation completed, we should have enough information and results to start comparing and analyzing our data. Hopefully we can answer questions like: Is PAKE secure?, How secure is PAKE?, Is it practical?, What cryptanalysis techniques is PAKE succeptible to?, etc.

4 References

We currently have a number of references that we can use in order to define the problem, and other resources to help understand PAKE implementation and cryptanalysis. These items are hyperlinks to the reference source.

- Wikipedia's Key Exchange
- Wikipedia's password-authenticated key agreement
- Computerphile Secret Key Exchange (Diffie-Hellman)
- PAKE - Password Authenticated Key Exchange
- Let's talk about PAKE
- Building encrypted chat app with the Web Cryptography API
- PAKE Crypanalysis Research Paper
- A Comparison of the PasswordAuthenticated Key Exchange Protocols, SRP-6a and PAKE2+