# The Feasibility and Application of using a Zero-knowledge Protocol Authentication Systems

Becky Cutler
Rebecca.cutler@tufts.edu
Mentor: Professor Chris Gregg

## Abstract

Modern day authentication systems utilize cryptography and the concept of secret keys by allowing participators to communicate only if they have the proper identification: their own public and private keys. This system relies on both parties having previously agreed upon the secret key system, and requires each user to maintain secure possession of their own keys. Though this method may seem cryptographically secure, there are a variety of vulnerabilities. Attackers could obtain a users private key, or eavesdrop on the conversation and obtain the transmitted data. A zero-knowledge protocol is secure alternative to modern authentication systems because it allows for user authentication without transmitting vital data. With the current degree of online privacy at an alarmingly low rate, there is a need now more than ever for cryptographically sound protocols that allow for safe and secure transactions.


## Introduction

In a zero-knowledge authentication protocol, users prove an assertion or secret without unveiling any hidden information about it. Because the "Prover" can only show that the statement is in fact true, the protocol extinguishes any opportunities for attackers to gain access to secret information because there isn't any transmitted. Zero-knowledge protocols are free from common cryptographic vulnerabilities because they forbid users to falsely authenticate themselves. In the protocol, if the Prover does not know the secret information, then they are unable to accurately guarantee the assertion and therefore cannot fulfill the necessary steps to complete the proof. These characteristics, in addition

to others discussed later, show that zero-knowledge proofs are an effective and efficient method for user identification and authentication.

## To the Community: What is cryptography and why does it matter?

Whether you know it or not, cryptography is vastly used in your every day life. From email to online shopping, cryptography allows us to go about our online activities without worrying that someone might try to steal our personal data. The main purpose of cryptography is to hide private information so it can be securely transferred between parties. This transfer process usually involves encrypting the data so that it is rendered useless to an eavesdropper, but upon decryption the data is readable once again. The encrypted message is called cipher text while the original message is called plain text. This process of encrypting plain text to cipher text should be easy for the parties involved, however the reverse process of transforming cipher text to plain text should be virtually impossible (6). This way, not everyone who obtained the cipher text could translate the information into plain text. This concept is called a one-way function and it is often used to encrypt messages so that it is infeasible to reverse the encryption and obtain the original data. In symmetric crypto systems, this process is paired with a key that is only known to the sender and receiver of the message. Both parties agree on a secret key before communication begins, and this additional information allows the receiver to decrypt the cipher text and unveil the message.

Although this system appears secure, there are numerous vulnerabilities. Because both parties must know the key before beginning communication, they must be cautious

of the exchange so an eavesdropper does not acquire the key. Additionally, it is necessary to change the key frequently to ensure that if an attacker does gain access to the key, it will not continuously give them access to the hidden information (6). Because of the trust issues that come with symmetric cryptography, it is not used frequently in modern day systems. Instead, public key cryptosystems are more commonly used.

Unlike symmetric key systems, public key cryptosystems dictate that each user must have a key that is publically available, and their own personal private key. Any message encrypted with the owner's public key can only be decrypted using their private key (6). In this system, there is no need for a secure meeting between both parties because the sender can encrypt the message using the receiver's public key, and then the receiver can decrypt that message using their private key. Once again, a one-way function is used to encrypt the message to ensure that it is nearly impossible to backtrack to the original plain text. Because of its secure characteristics, public key cryptography systems are often used in the authentication process to verify a user's identity, such as logging into a website or checking your email (6).

Although the idea of public key cryptography appears secure, the entirety of this system is only as strong as its algorithm. If any of these methods or procedures are broken, the entire cryptographic system becomes compromised, leaving an open door for attackers to gain access to the keys or the plain text data. Additionally, failure to properly implement the cryptographic algorithms could cause the system to fail. In some cases, the plain text data may not be properly erased after it is encrypted and thus is easier to exploit

(6). What this means for you is that all of your data, from credit card information to browsing history, could potentially be out in the open for anyone to view.

There is also a great deal of trust that comes with asymmetric cryptography. For example, the system can still be vulnerable if the keys used in the transaction were not certified by a viable source such as the Certification Authority (7). In this case, someone could potentially impersonate the user by illegally logging into their account and accessing private information (2). Essentially, the security of public key cryptography relies on many wavering factors. But when it comes to accessing your personal data, don't you want the safest and most secure method of retrieval? This is where zero-knowledge proofs come in. These protocols allow for easy yet secure authentication without sending private data out into the open.

# Applications

## What are zero-knowledge proofs?

There are many vulnerabilities that lie within symmetric and asymmetric authentication schemes, bringing about the need for a unique system that is not susceptible to common attacks. Zero-knowledge proofs provide a method of authentication though a secure form of communication. The main concept behind zero-knowledge proofs is proving knowledge about some piece of secret data, without directly revealing any information about it. The protocols of a zero-knowledge proof allow the user to prove to the other party that they know the "secret", however the secret itself is never transferred between the parties. In this instance, a "Prover" shows that they know some secret to the "Verifier" without directly revealing the secret. The Verifier can ask questions to show that the Prover knows the secret, but it is infeasible for the Verifier to

unveil information about the secret whether or not they properly follow the protocol (5). The Prover must respond to challenges issued by the Verifier, such that if the protocol is repeated T times, all rounds must be correctly answered by the Prover to show that they know the secret and are in fact who they say they are (4). For the system to be properly implemented, the protocol must follow two criteria to withstand attackers. For example, if the Prover wants to falsely authenticate themself, and they do not know the secret, then it is infeasible for them to pretend they do. Many repeated rounds of the scheme must guarantee that the Prover cannot deceive the Verifier. Similarly, the Verifier must not be able to obtain additional information that might allow them to convince someone else that they know the secret (5). The following example of a zero-knowledge proof demonstrates these principles.

## Real world example

In this example, the Prover wants to assert that they can count the leaves of a large tree within seconds. They cannot reveal their exact method, or the current number of leaves on the tree. In one round of the protocol, the Prover closes her eyes while the Verifier can either take a leaf off, or do nothing. The Prover then examines the tree and describes what the Verifier did. If the Prover is incorrect, then the Verifier instantly knows that the Prover is lying and the assertion fails. However, if the Prover is correct, then the Verifier may think that it was a lucky guess, so the protocol is repeated. This process can be repeated thousands of times, but after approximately 100,000 rounds, the Verifier should feel confident in the Prover's knowledge. In this example, the Prover was able to demonstrate that she had the knowledge to quickly count the leaves, without

revealing two vital aspects: the methodology or the current number of leaves. Herein lies the importance of a zero-knowledge proof. Because the Prover did not discuss the methodology, the Verifier cannot impersonate the Prover because they do not know the necessary information. Additionally, the repetition of the protocol insures that the Verifier has control over when they believe that the assertion holds true (5). This example leads to a discussion that zero-knowledge proofs can be implemented in user authentication schemes.

## Zero-knowledge proofs in authentication systems

An authentication scheme is a type of protocol that allows a user P to prove her identity various times to a user V, without allowing V to falsely represent himself as P to another user (3). This proof of identity happens in real time, and depending on if the protocol was either accepted or rejected, the desired access is granted or denied. As discussed previously, one of the main concerns with current authentication schemes, such as an encrypted password, PINs or identification cards, is that P must expose secret information by means of a numeric value or printed card (3). In order for a zero-knowledge proof to be properly implemented for an authentication scheme, it must have both completeness and soundness. It is complete because if the Prover properly implements the protocol, authentication is successful. It is sound because no user can falsely identify themself (1). Finally, the proof is dubbed "zero-knowledge" because if the assertion is true, then the Verifier doesn't know any other secret information other than that the statement is true (1). One useful application of this kind of proof is in the use of websites, which is where approximately 80% of the reported software security

flaws are found (1). Although there are many ways to implement a zero-knowledge protocol for website authentication, the solution that follows properly implements the necessary characteristics while utilizing basic mathematics (1).

**Identification Protocol:** (let $g_0$ be the user's public key)

1. User enters their username and password
2. The hash of the password is calculated: x = H(password)
3. User calculates $Y = g_0^x$
4. User creates a random number *r* and computes $T_1 = g_0^r$
5. User computes the hash of Y, $T_1$, and $a^*$: $c = H(Y, T_1, a)$

*Note: *a* was received from step 1 of the authentication protocol

6. User computes $z = r - cx$
7. User transmits username, Y, c and z to server, where it is stored

**Authentication Protocol**

1. Server creates a one-time token *a*, stores it, and transmits it to the user
2. Server receives username, Y, c and z from user
3. Server calculates $T_1 = Y^c g_0^{zx}$
4. Server verifies that $c = H(Y, T_1, a)$
5. If the verification is successful, the user has been authenticated

*A step-by-step explanation of how a zero-knowledge proof authentication process can be integrated into a web application can be found in the supporting material.

Integrating zero-knowledge proofs into website authentication systems allows for a safe and secure login without sending vital data, such as a password hash, over the network. Additionally, because the system never obtains the password, it cannot store it in a database. This creates another level of security because if an attacker gains access to the database, they cannot retrieve the passwords (1). Possible eavesdroppers cannot learn any useful data that could be used for false authentication. An added layer of security can be used through a one-time token that is transmitted when the user begins the authentication process. This token is only valid for one session and therefore an attacker cannot reuse the intercepted information (1). Although zero-knowledge protocols may be more work for the user or server than typical asymmetric cryptography, this solution is

not extremely challenging to implement, and could possibly save the system work later on by withstanding dangerous adversaries.

## Conclusion

In the 21st century, if you have access to the Internet then you most likely have personal data, whether it be passwords or credit card numbers, that you hope is securely in your possession and no one else's. However, various adversaries have targeted web applications due to their increasing popularity. With common cryptographic schemes such as symmetric and asymmetric keys, there are numerous vulnerabilities that allow attackers to falsely authenticate themselves and gain access to your personal data. By implementing a zero-knowledge protocol, no direct data is transmitted between parties, thus alleviating the possibility of an eavesdropper obtaining private information. Additionally, because there are no passwords or numeric data stored in the server's database, if an attacker gained access to the database, they would not be able to acquire any useful information. Although cryptography remains a vital part of web-based security, attackers are constantly evolving and outsmarting security protocols. Because of this, cryptography must evolve alongside these attackers to ensure completely sound protocols. Zero-knowledge proofs are the next step in cryptography's ever-changing history, creating a more safe and secure transaction.

## Supporting Material

**YouTube Link**: https://www.youtube.com/watch?v=Uulmz86eP1M

## Works Cited


1. Lum Jia Jun, Brandon. "Implementing zero-knowledge authentication with zero knolwedge." *The Python Papers Monograph* 2, no. 9 (2010): 1-19.
2. Lowe, Gavin. "An attack on the Needham-Schroeder public-key authentication protocol." *Information Process Letters* 56 (1995): 131-33.
3. Feige, Uriel, Amos Fiat, and Adi Shamir. "Zero knowledge proofs of identity." *Association for Computing Machinery*, 1987, 210-17.
4. Gronkowski, Slawomir, Wojciech Zaremba, Maciej Zaremba, and Bill McDaniel. "Extending web applications with a lightweight zero knowledge proof authentication." *Association for Computing Machinery*, 2008, 1-6.
5. Giani, Annarita. "Identification with zero knowledge protocols." *SANS Institute*, 2001, 1-7.
6. Mohapatra, Pradosh Kumar. "Public key Cryptography." *Crossroads: The ACM Student Magazine*, 1-19.
7. Schneier, Bruce. "Cryptographic design vulnerabilities." *The ACM Digital Library* 31, no. 9 (September 1998): 29-33.