Authentication Security

Shobana Chandrasekaran
Computer Science
University of Southern California
Los Angeles, USA
shobanac@usc.edu

***Abstract*— Zero-knowledge authentication is an alternative to public key authentication. This paper discusses the frequently used authentication systems and talks about some of their weaknesses. ZKP and its applications in blockchain, cryptocurrency etc. are discussed. The working of two famous identity-based protocols is also discussed. Various studies of zero-knowledge based authentication are discussed in detail. The paper concludes with the security and performance aspects of the zero-knowledge proof of identity.**

I. Introduction

The process of being able to confirm one's identity has always been inevitable in applications ranging from simple access control mechanisms to e-commerce models. Authentication has become a significant prerequisite for secure transactions and access to vital information. Cybercriminals are finding new means to get access into a system and steal information. This asks for authentication schemes to be put in place to secure the systems. Moreover, if the scheme itself is not secure it has far reaching consequences of putting an entire organization at risk.

This paper discusses the traditional authentication schemes and their vulnerabilities in Section 2. Section 3 gives an overview of zero-knowledge proofs. In section 4, we dive into some real-world applications of zero-knowledge proofs. Authentication systems being one of its applications, we discuss zero-knowledge based authentication and its security and performance aspects in section 5 and section 6 respectively.

II. Existing authentication mechanisms

The user authentication methods fall into three broad categories:

- Something the user is (voice recognition, retinal scanners etc)
- Something the user has (IDs, smart cards)
- Something the user knows (passwords, PINs)

This section discusses the types of existing authentication mechanisms and the vulnerabilities in each type.

1. *Password-based authentication*

It is the most frequently used method of authentication. The plaintext password is sent directly or through an SSL connection. Sometimes hash of the password is sent which is verified using a stored hash. A process called salting is also used where random bits are appended to the password before it is hashed to prevent pre -computed dictionary attacks [1].

There are several attack vectors when a system uses passwords to authenticate users. One is brute-force hash cracking[2]. The adversary compromises the server containing the hashes of the passwords and later carries out a brute-force attack by computing the hashes of common passwords trying to find a match in the compromised server's hash database.

Another possible attack is wire sniffing[2]. A malicious third party listens in on the conversation between a client and a server and sniffs out the password from the traffic. To alleviate this, SSL can be used for authentication and confidentiality.

An attack that stems from brute-force hash cracking is identity theft. By stealing the authentication database, the attacker can pretend to be someone else he's not. Furthermore, when users use the same password for multiple services, once an attacker gains access to the password he gets a chance to commit fraud in all the services the user has signed up for.

2. *Multi-factor authentication*

It combines two or more authentication methods. Access to the system is granted only when the user is able to provide evidence for each authentication mechanism. Common examples include password along with a code sent to the phone, password with fingerprint, pin with card details etc. It adds multiple levels of security to the system.

Even though it provides additional layers of security, it simultaneously puts the burden on users. Because not only the user has to manage a password, he also has to take care of the additional steps incurred by the layers. Also, there are ways in which MFA can be bypassed. Some of them are:

- Phishing the MFA code
- Brute-forcing the MFA code (if the code generation is not random)
- Hijacking the user session
- SIM swap, where attackers port the user's phone number to theirs
- Exploiting logical errors in the application (Eg: Skippable authentication)
- Gaining access to Single Sign-On password (can log into service B by logging into service A, easily bypassing MFA in service B)

3. *Certificate-based authentication*

A certificate contains the user's public key along with the digital signature of the certification authority. During authentication, a client provides the verifier with his/her certificate. The verifier checks the credibility of the certificate and poses cryptographic challenges to the user to check whether the public key found in the certificate corresponds to the user's private key.

Certificate-based authentication only verifies whether the private key corresponds to the public key in the certificate; it doesn't guarantee that the keys belong to the actual owner. So it is an added responsibility to the owner to keep his private key a secret.

There is an overhead in issuing, managing and maintaining the certificates. The credibility of the CA also plays an important factor in certificate-based authentication. Certificate-based authentication is considered to be more complicated than password-based authentication.

4. *Biometric authentication*

It is based on "something the user is". Common methods include facial recognition, voice recognition, fingerprints scan, iris and retinal scan etc. Special kinds of authenticating devices are required to authenticate the users.

Privacy issues are no new news in biometrics. Any malicious party can record one's voice, take one's photo and use it without consent in facial recognition, copy one's fingerprints etc. The equipment used to authenticate is also subject to flaws: accepting an unauthorized person or rejecting an authorized person.

5. *Token-based authentication*

These technologies require users to enter the credentials once. Upon authentication, users receive a token that can be used to access the system any number of times instead of entering the credentials once again.

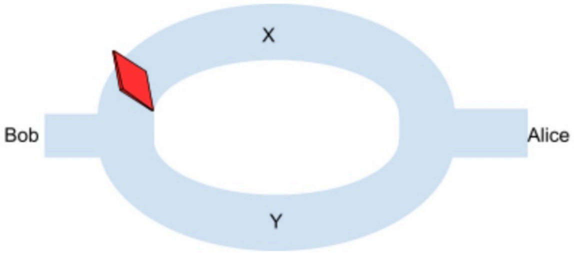
Session tokens when not generated in a secure manner pose the risk of guessing tokens by the adversary. There are also problems of adversaries taking advantage of long token-validation time and insecure token storage, all leading to the adversary stealing the token.

Some of the issues discussed above can be remediated using:

- Public-key crypto systems either for authentication itself or for encrypting the authentication database. It requires key generation, exchange and management.
- Secure communication protocols for authentication like SSL/TLS, HTTPS etc. provide a secure communication channel between the hosts.
- Zero-knowledge proofs. It does not require key exchange or storage and does not reveal the password. However, it is computationally intensive.

III. Zero-Knowledge Proofs

Zero-knowledge proof (ZKP) is one of the most powerful and popular cryptographic techniques first introduced by Goldwasser, Micali, and Rackoff [3]. ZKP is a challenge-response protocol where there are two parties: a prover and a verifier. The verifier puts forth mathematical challenges to the prover. The prover's aim is to prove the correctness of an item of information he is in possession of without exposing what the data is. To illustrate the logic behind the protocol, let us consider the famous "Ali Baba cave" example[4]. There are two characters: Alice and Bob. The cave is structured as shown below:



Bob is at the cave's entrance. Alice is on the other side of the cave. Bob has two paths to exit the cave: X, Y. In order to travel through path X, Bob has to know the magic words to open the red door. So Bob (prover) has to show Alice (verifier) that he knows the magic words to open the door.

The process goes like this: Alice names the path that Bob has to take. Since Bob claims he knows the magic he will always be able to exit the cave through the path that Alice asks him to assuming Bob is an honest prover. However, there is a 50% chance that Bob knows the magic. In order to reduce the chance of Bob cheating, the process is carried out multiple times. After a sufficient number of iterations, Bob has demonstrated to Alice the validity of his statement without revealing what it is.

A. *What makes it Zero-knowledge?*

Goldwasser, Micali, and Rackoff proposed three essential requirements that every ZKP must satisfy[2]. They are:

- *Completeness:*

If the statement is true, the honest verifier will be convinced by the honest prover's claim.

- *Soundness:*

If the statement is false, a malicious prover can convince the truth of the statement to an honest verifier with negligible probability.

- *Zero-knowledge:*

The verifier will not be able to know any information beyond the validity of the statement.

Now the question that arises is whether ZKP can be used in all problems. In [3], Goldwasser proved that any problem that is NP has a zero-knowledge proof under the assumption that one-way functions exist. But not all of them can be implemented in reality. Fiat-Shamir was the first realistic zero-knowledge protocol. A number of other protocols came into the picture after this like Schnorr, Guillou Quisquater, Ohta-Okamoto, Beth etc.

There are two types of ZKPs:

- *Interactive* - both the prover and verifier are online and verification requires interaction between the two.
- *Non-interactive* - the verification requires no interaction between the prover and verifier or in some cases deferred verification occurs. It should be noted that these require additional software or computer overhead.

B. *ZKP Applications*

ZKPs have the ability to enhance privacy and security in a wide range of applications ranging from fraud prevention systems (requires user's age) to the Internet of Things (feeding anonymous data). This section covers some major applications of ZKP.

Blockchain Technology

The entire functionality is based on the decentralized consensus protocol. This protocol maintains a common global record which is called the ledger of all the transactions thus allowing transparency to all users. The protocol is supported by individuals referred to as miners who add transactions to this global ledger by solving a cryptographic puzzle. This global ledger is commonly referred to as Blockchain. Therefore, the technology is trusted for accuracy, transparency and accessibility but not for privacy. Incorporating zero-knowledge proofs in blockchain technology provides a mix of privacy, security and immutability. It also ensures the validity of the transactions in the ledger.

Zk-SNARKs

Zero-Knowledge Succinct Non-Interactive Arguments of Knowledge (zk-SNARK), coined by Alessandro Chiesa[5] is a non-interactive ZKP that has the following characteristics[6][7]:

- Zero-Knowledge: As mentioned above, zero-knowledge proofs help a prover prove the truth of a statement to a verifier without revealing any other information.

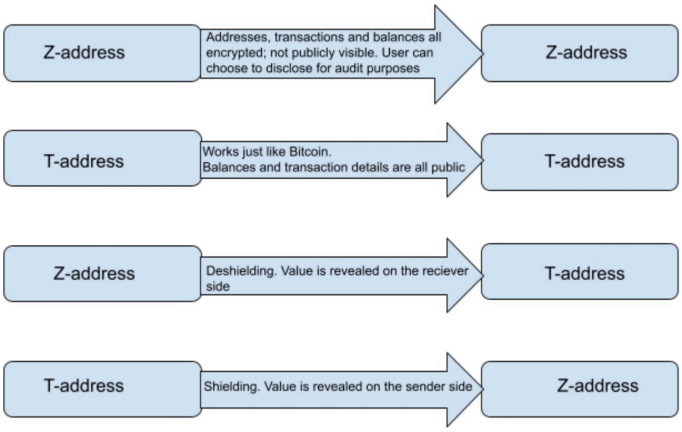
- Succinctness: This is the most important trait in zk-SNARK. It implies that the proofs can be verified in a matter of milliseconds and the length of the proofs are only a few hundred bytes. This makes zk-SNARK an attractive technology but has limitations of its own.
- Argument: It imposes computational constraints on the prover so that wrongful statements are not created in the proof by the prover. This provides protection for the verifier.
- Knowledge: It implies that any prover cannot create proof without the knowledge of a witness.

Zk-SNARKS depends on elliptic curve cryptography for security and quantum attacks loom over elliptic curves. So zk-SNARKS are not quantum resistant. In addition to being based on elliptic curves, zk-SNARKs require an initial setup phase to exchange the common reference string between the prover and verifier. The string is helpful in creating non-interactive short ZKPs. Therefore, it is important that the initial setup phase be performed correctly. Zk-SNARKs are the underlying technology in Zcash, a decentralized privacy-protecting cryptocurrency.

Zcash

Zcash, a cryptocurrency like Bitcoin, is built with Bitcoin as a base but provides a choice of privacy feature to the users to make the transactions confidential. Bitcoin transactions are transparent, so everyone can see them. Bitcoin is trusted for security but not trusted for privacy. Details of the sender, receiver and the amount transacted are not disclosed in ZCash. It adopts ZKPs, in specific zk-SNARKs, to encrypt all the information. To make it audit and regulation friendly, the decryption keys are disclosed by Zcash users to approved parties to view the transactions. A good analogy to compare Bitcoin and ZCash would be HTTP and HTTPS.

It was mentioned that Zcash users are given a choice to make their transactions public or private. This is achieved using Zcash's two types of addresses: private (z-address) and transparent (t-address). This implies there are 4 transaction types in Zcash.


Z-address
Addresses, transactions and balances all encrypted; not publicly visible. User can choose to disclose for audit purposes
Z-address
T-address
Works just like Bitcoin.
Balances and transaction details are all public
T-address
Z-address
Deshielding. Value is revealed on the reciever side
T-address
T-address
Shielding. Value is revealed on the sender side
Z-address

ZKP-based voting system

In [8], Mutaza et. al. proposed a voting system that is based on ZKP. The voters are assigned a Voter ID and a public key by the election commission. The IDs and the public keys are signed by the election committee (ID token) and the ID token is pushed into the blockchain. Once the voters prove their identity using biometrics, the election commission provides a secret PIN to the voter. The PIN along with the barcode in the voter's passport serves as a private key. On the election day, the polling machine asks for the voter's barcode which prompts the voter to cast his vote. Once the candidate is selected, the secret PIN is entered and the vote is cast. The polling machine generates a large random number S and a small random number R. These are used to identify the voterID and make it unusable. This process is called burning. The identification of the valid ID token in the blockchain is done by constructing a ZKP using S and R where the voter proves the validity of the statement that he knows R such that when combined with S gives the ID token. The vote is sent to the candidate after the ZKP along with the signature of the voter and verified by the blockchain. The system is decentralized and all transactions, signing and verification are done across the blockchain.

In Embedded systems

In [9], the author introduced a C library, ZPiE, that generates Zero-Knowledge Proofs in embedded systems. These systems are severely restricted in processing power and storage. Thanks to zk-SNARKs, the creation and verification of ZKPs have been made easier.

In Authentication Systems

A party demonstrates its identity to a verifier using some secret information but does not show secret data to the verifier. ZKP in authentication systems is an important incorporation and serves as an alternative for public-key authentication. The next section talks about this in detail.

IV. ZKP in Authentication

In section 2, we saw that one of the major problems with classical ways of authentication is interception where a third party can eavesdrop on the authentication data to later fake his identity. There is also a problem with password-guessing attacks. To solve the problem of authentication information exposure, ZKP can be adopted to confirm a user's identity. In this section, we discuss the ZKP application in the authentication.

a. *Protocol Basics*

There are several approaches to Zero-knowledge protocols each using different data structures and having unique properties [10]:

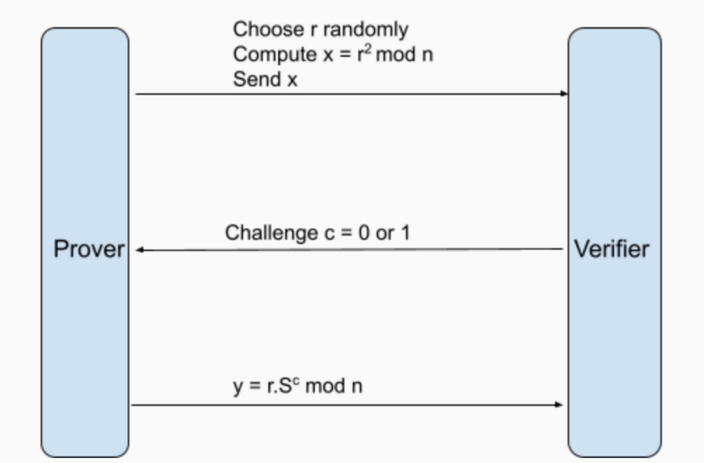
- Discrete logarithm problem: Given real numbers a and b, it is hard to find x such that $b^x \bmod n = a$.
- Elliptic Curve Cryptography: ECC offers the same level of security as public-key systems with smaller key sizes[11]. So in environments where there is a constraint on memory, time and energy, ECC outperforms conventional public-key crypto systems[11].
- Graph Isomorphism: Two graphs (G_1 and G_2) with the same set of vertices are said to be isomorphic if there exists a permutation on vertices. The public key in this case is the two isomorphic graphs G_1 and G_2; the private key is a permutation π_p such that $G_2 = \pi_p (G_1)$. A prover generates a random permutation $_r$ and sends a graph $G_r = \pi_r(G_1)$ to the verifier. Then, depending on the verifier's challenge, the prover sends back π_r or π_s such that $\pi_s = \pi_r \circ \pi_p^{-1}$. The verifier is able to check one of

the conditions: $G_r = \pi_r(G_1)$ or $G_r = \pi_s(G_2)$. The protocol does not reveal the prover's private key.

Among the many zero-knowledge based protocols, let us discuss the most used and most talked about protocols that are identity-based: Fiat-Shamir and Guillou-Quisquater.

Fiat-Shamir Identification Scheme

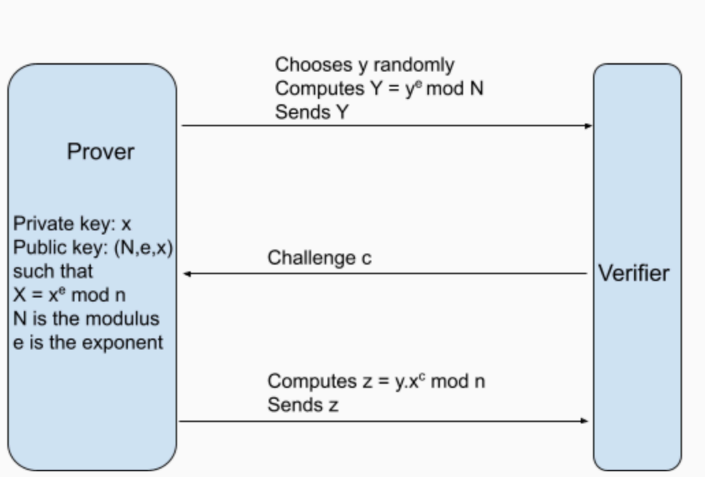
Initial Setup: There are two parties: a prover and a verifier. Two large prime numbers, p and q, are chosen by a trusted third party such that p*q = n. n is made public to the prover and verifier. The prover chooses a secret S (private key) and computes her public key $v = S^2 \bmod n$ and makes it available to the prover. The rest of the exchanges in the protocol is shown in the figure below:



The verifier checks if $y^2 \cong r^2 . v^c \bmod n$. If it is not congruent then the verifier rejects the proof.

Guillou-Quisquater Scheme

The GQ scheme is very much similar to Fiat-Shamir identification scheme. The challenge c takes any value between 1 and e.



The verifier checks if $z^e \cong Y.X^c \bmod N$. If not the proof is rejected by the verifier.

Based on the experiments done by Ahmed Patel et.al. in [15], Guillou-Quisquater performs better than Fiat-Shamir in terms of memory and communication complexity.

b. ZKP-Authentication

Web-based applications are seeing a huge increase in usage and popularity. They have replaced traditional desktop applications because of their scalability, cost-effectiveness, easy-to-update and easy maintenance. Authentication has always been an integral part of the internet and has been evolving with the internet. Practical implementations have been done to bring ZKP authentication into web applications.

The paper, *Lightweight Zero-Knowledge Proof Authentication* [10], presents a classic ZKP algorithm based on graph isomorphism and implements in web browsers like Firefox, Opera, Internet Explorer with different system configurations (varying RAM, CPUs and RAM sizes). The entire process depends wholly on the web browser. The user enters the username and password in the web browser which does not reach the webserver. With the password received, the web browser is responsible for generating the public and private key pairs and challenge graphs. It also has a role to respond to the challenges put forth by the webserver. From the graph isomorphism algorithm, we know that the private key is a permutation. The paper discusses a series of steps to convert a user's password into a permutation using hash algorithm SHA1. The webserver has only two roles: creating a random challenge and the verification step. The paper has successfully demonstrated the incorporation of ZKP in web authentication. However, it is one-way (i.e.) it presents only client-side authentication and not both client and server authentication. The performance tests for waiting time and the amount of data exchange are taken into consideration with only the client-side in mind. Applications requiring both server and client-side authentication cannot adopt this approach. Also, the performance statistics will significantly increase when two-way authentication is taken into account.

ZKP based authentication scheme is implemented using the two famous protocols: Feige-Fiat-Shamir and Guillou-Quisquater in [12]. The implementation involves an FTP client (prover) trying to access an FTP web server (verifier) after proving its identity. Each of the two identification schemes is adopted and the time taken by the prover and verifier were evaluated. It was found that Guillou-Quisquater takes more time for computation than Fiat-Shamir. However, Fiat-Shamir suffers from an impersonation attack and the challenge value it assumes is either 0 or 1. Because of these reasons, it was concluded that the Guillou-Quisquater scheme is better than the Fiat-Shamir scheme. Just as in [10], there is an underlying assumption that the webserver is an honest verifier. The authentication is one-way. A better authentication scheme should be both efficient and secure. Because a scheme is secure, doesn't mean it is efficient. The paper lacks sufficient evidence to show that Guillou-Quisquater is better than Fiat-Shamir. So, additionally, communication costs and memory costs should also be evaluated to compare the efficiency of the two identification schemes.

To reduce the response time, theories have been proposed to combine ZKP with a public-key cryptosystem. One such is ZKP with RSA (NARWHAL) by [13]. Because encryption-decryption is involved, a database is required for storing the user details. But it is made sure that even if an attacker gets hold of this database, he will not be able to deduce the plaintext password from it. So the scheme proposed here is not purely interactive. And because keys are involved, there are issues of key exchange and management. Javascript has been used to implement the scheme which requires the Javascript to be enabled. It was cited in the paper that around 2% of users have disabled Javascript (in 2010). So this requires work from the user and also trust to be placed on the website.

In applications that require minimum response time and computational complexity, especially IoT and WSNs (Wireless sensor networks), ZKPs are combined with application-specific protocols for efficiency and security. In [14] puts forth a base station authentication mechanism whereby a base station authenticates itself to the nodes in its network. A modified version of the Guillou-Quisquater has been proposed. A group of nodes in the network confirms the identity of the base station. The modified GQ protocol is combined with the µTESLA protocol for broadcast authentication. The process goes like this: initially, a random sensor node puts forth a challenge e. The challenge is sent as a broadcast message to ensure that any fraudulent base station does not replace the challenge. In response to the challenge, the base station sends the authentication message to all the nodes using the µTESLA protocol. In successive rounds, the sensor nodes are chosen in a round-robin fashion.

A software library for zero-knowledge authentication has been designed and implemented for smartcard applications with Java Card as the target platform [15]. Fiat-Shamir and Guillou-Quisquater were both studied under different performance criteria (communication cost, memory, etc.) to find a suitable protocol. Guillou-Quisquater was chosen for implementation. Authentication has two scenarios here: initialization and user authentication. In the initialization phase, the parameters of the GQ protocol (exponent, product of two primes), the public-private key pairs and PIN is generated for a user. In the user-authentication scenario, there are two interfaces: user interface and reader interface. The user enters his PIN to the user interface which talks to the reader interface. The reader interface serves as a bridge between the Java applet (prover), verifier and the UI. An authentication request containing the PIN is sent to the applet to start the GQ protocol between the applet and the verifier. The implementation is shown to be possible but requires more improvements in areas of memory utilization. The library can not be administered in devices with less than 1KB RAM and 16 GB ROM.

In IoT devices such as smart vehicles that move at a high speed, running interactive ZKPs is infeasible due to possibilities of connection disruptions. Time plays a crucial role in situations like this. Non-interactive ZKPs can be put into action in such situations. All the challenges are enclosed in a single message making it time-efficient. *Authentication based on NIZKPs for the Internet of Things* [16] proposes a non-interactive ZKP based authentication using the graph isomorphism approach. The base node verifies the identity of the IoT device in the network unlike [14] which focuses on base-station authentication.

Kerberos, a widely used authentication mechanism, is prone to password guessing and replay attacks. Public key cryptography was adapted to overcome this vulnerability; however, it increases computational and communication complexity. Yuesheng et.al [17] proposed an enhanced Kerberos with NIZKP. It offers both client-side and server-side authentication without revealing any information associated with the client or the server. The proposed method is suitable in memory-constrained environments like mobile devices. The enhanced version adopts the ZKP for getting the ticket t_{gs} and the session key $K_{c,tgs}$ used for communicating with TGS. The first step is to modify the Feige–Fiat–Shamir's scheme to a non-interactive version as shown in the picture below:

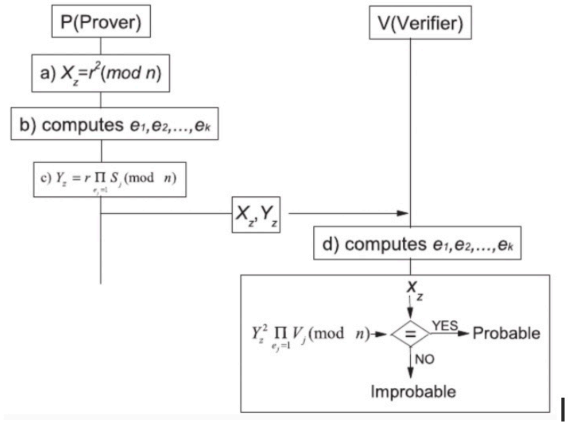

P(Prover)
V(Verifier)
a) $X_z=r^2 (mod\ n)$
b) computes $e_1, e_2, \dots, e_k$
c) $Y_z = r \prod_{e_j=1} S_j \pmod{n}$
X_z, Y_z
d) computes $e_1, e_2, \dots, e_k$
X_z
$Y_z^2 \prod_{e_j=1} V_j \pmod{n}$
=
YES
Probable
NO
Improbable

Image taken from [17]

The client then sends a request to the KDC such that the request message contains the User name, KDC name, timestamp, nonce, $K_{c,as}$ and ZKP parameters encrypted using the public key of AS. On receipt of the request, the KDC decrypts the message, checks the timestamp, computes e_1, e_2,…, e_k and verifies whether $X_z = Y_z$. If valid, KDC sends the response message containing TGT t_{gs}, session key $K_{c,tgs}$, timestamp, nonce, TGS name all encrypted using $K_{c,as}$. The remaining exchanges follow Kerberos protocol. The performance achieved by the proposed ZKP-Kerberos in terms of computation and communication costs is better than previously proposed schemes.

V. Security and Performance

Traditional authentication mechanisms are majorly prone to replay attacks, password-guessing attacks and authentication database leakage. However, ZKP-based authentication is resistant to these attacks because the data shared during the communication between the prover and verifier is unusable. Also, no sensitive information is stored or sent over the network in zero-knowledge authentication in plaintext form.

Zero-knowledge proofs are not without challenges. If not implemented properly, zero-knowledge proofs of identity are susceptible to man-in-the-middle attack: a prover (P) proves his identity to a verifier (V*) who in turn can prove to another verifier(V) that he is the prover (P). This can be avoided by imposing time constraints.

A main drawback of ZKP is performance. It is significant to keep the computational complexity and response time at a reasonable level. From [18], across five trails in desktop and mobile, the minimum and maximum response times for a single iteration are 57ms - 1230ms and 356ms - 1860ms respectively. Many variants have been proposed to enhance the performance of ZKP. Succinct proofs have been found to have better communication complexity so far. However, there is still a concern about the initial trust setup phase.

VI. Conclusion

This paper talked about the various authentication methods and their weaknesses. The concept of zero-knowledge was introduced and how it can be incorporated for authentication for added security. Zero-knowledge based authentication has started taking its role in different areas from web to IoT. In conclusion, ZKP is found to be significant technique for secure and authenticated transaction between two parties. From the various studies, it is found that non-interactive ZKPs are more efficient. Succinct proofs are found to be suitable for systems with low processing and memory capacity. However, there is a problem of initial setup phase. But ZKP and its variants have a promising role in authentication and requires improvements to be made to reduce its overhead.

References

[1] Zhou, Minqi, Rong Zhang, Dadan Zeng, and Weining Qian. "Services in the cloud computing era: A survey." In *2010 4th International Universal Communication Symposium*, pp. 40-46. IEEE, 2010.

[2] Jun, Brandon Lum Jia. "Implementing Zero-Knowledge Authentication with Zero Knowledge (ZKA wzk)." *Python Papers Monograph* 2 (2010).

[3] Goldwasser, Shafi, Silvio Micali, and Charles Rackoff. "The knowledge complexity of interactive proof systems." *SIAM Journal on computing* 18, no. 1 (1989): 186-208.

[4] Wikipedia link: https://en.wikipedia.org/wiki/Zero-knowledge_proof

[5] Eli Ben-Sasson, Alessandro Chiesa, Christina Garman, Matthew Green, Ian Miers, Eran Tromer, Madars Virza, “Zerocash: Decentralized Anonymous Payments from Bitcoin,” 2014 IEEE Symposium on Security and Privacy.

[6] ZK-SNARKs and Zcash Link: https://z.cash/technology/.

[7] ZK-SNARKS https://101blockchains.com/zksnarks-introduction/

[8] M. H. Murtaza, Z. A. Alizai and Z. Iqbal, "Blockchain Based Anonymous Voting System Using zkSNARKs," 2019 International Conference on Applied and Engineering Mathematics (ICAEM), 2019, pp. 209-214, doi: 10.1109/ICAEM.2019.8853836.

[9] Salleras, Xavier, and Vanesa Daza. "ZPiE: Zero-Knowledge Proofs in Embedded Systems." *Mathematics* 9, no. 20 (2021): 2569.

[10] Grzonkowski, Sławomir, Wojciech Zaremba, Maciej Zaremba, and Bill McDaniel. "Extending web applications with a lightweight zero knowledge proof authentication." In *Proceedings of the 5th international conference on Soft computing as transdisciplinary science and technology*, pp. 65-70. 2008.

[11] Chatzigiannakis, Ioannis, Apostolos Pyrgelis, Paul G. Spirakis, and Yannis C. Stamatiou. "Elliptic curve based zero knowledge proofs and their applicability on resource constrained devices." In *2011 IEEE Eighth International Conference on Mobile Ad-Hoc and Sensor Systems*, pp. 715-720. IEEE, 2011.

[12] Kayathri Devi, D., and S. S. Akilan. "Comparison of ZKP based Authentication Mechanisms for securing the web server."

[13] Cheu, R. et.al “An Implementation of Zero Knowledge Authentication”, Massachusetts Institute of Technology Narwhal, 2014.

[14] Anshul, Dev, and Suman Roy. "A ZKP-based identification scheme for base nodes in wireless sensor networks." In *Proceedings of the 2005 ACM symposium on Applied computing*, pp. 319-323. 2005.

[15] Patel, Ahmed, Kenan Kalajdzic, Laleh Golafshan, and Mona Taghavi. "Design and implementation of a zero-knowledge authentication framework for Java Card." International Journal of Information Security and Privacy (IJISP) 5, no. 3 (2011): 1-18.

[16] Martín-Fernández, Francisco, Pino Caballero-Gil, and Cándido Caballero-Gil. "Authentication based on non-interactive zero-knowledge proofs for the internet of things." *Sensors* 16, no. 1 (2016): 75.

[17] Zhu, Yuesheng, Limin Ma, and Jinjiang Zhang. "An enhanced Kerberos protocol with non-interactive zero-knowledge proof." *Security and Communication Networks* 8, no. 6 (2015): 1108-1117.

[18] Schultz, Johnathon. "Prove Yourself: Zero Knowledge Password Authentication." (2016).