CSEC 604 Crypto and Authentication
Project Report

AUTHENTICATION OF IOT DEVICES USING MULTIGRAPH ZERO KNOWLEDGE PROOF

December 2, 2022

Vyshnavi Maddela
Pranjali Thakur
Dhruva Mhatre
Department of Computing Security
College of Computing and Information Sciences
Rochester Institute of Technology

1 Abstract

Many businesses try to gather as many details as they can about us in order to promote more advertisements, provide everyone with better recommendations, or hold us on their portals for long durations. This is because any of our personal data, such as our contact details, preferences, or behavioral patterns, is very valuable. It would appear to be difficult to maintain our information secret considering the number of businesses and services gathering it. We can predict that, as technology advances and the use of IOT devices rises, there will be a rise in the number of gadgets that rely on an internet connection. The security system will become more sophisticated as a result of the IOT devices being connected to it. Therefore we would require an authentication protocol for the rising IoT devices, which can carry our sensitive data but will never reveal the exact information. The Zero Knowledge Proof is one such approach where the even though data is transferred over an unsecured network called the internet our data will never be revealed to anyone. This research mainly focuses on how zero-knowledge proofs are used to verify the data and authenticate systems during this high-level transmission of data over the internet and to make an analysis of how the different protocols of ZKP show different results on IoT systems. Also to analyze Multigraph Zero-knowledge proof which reduces the overhead in IoT systems and hence we proposed, Multi graph with one round such that the computation power required is lowered in resource constrained IoT devices.

2 Motivation

Assume two persons, Alice and Bob, have a chat about knowing the password to an external account. Alice tries to persuade Bob that she knows the password and can access it, but Bob is skeptical and wants her to prove it. Alice wishes to demonstrate this to Bob, but she does not wish to disclose the password of the external account. The question now is whether Alice can demonstrate her secret knowledge without exposing any password information. Such proofs exist, they are referred to as Zero Knowledge Proofs. Alice can use zero knowledge proof to demonstrate her password knowledge to Bob in the following manner. 1. Alice requests that Bob close his eyes for a particular period so that she does not expose the password. 2. Alice enters the passcode and displays all the external account information.

3 Organization of Report

Section 4 of the report has the Introduction to the basics of the report, Section 5 deals with the past work done that is the Literature Review, Section 6 comprises of the Authentication done in IoT devices currently, Section 7 introduces Zero Knowledge Protocol (ZKP) in IoT, Section 8 has different ZKP Protocols proposed previously, Section 9 starts with the

Multi-Graph Approach to ZKP, Section 10 is the our approach to the M-ZKP, Section 11 consists of Future Work and Section 12 is the Conclusion of the entire report.

4 Introduction

IoT: The Internet of Things is essentially concerned with the notion that anybody may access any device, anywhere, at any time, on any network in any organization. The Internet of Things refers to a global network of networked devices and technology that enables interaction among devices, infrastructure, and technology, as well as among devices. In today's world, IoT devices transmit confidential and highly sensitive data via an unsecured network such as the internet, whether for diagnostic purposes, e-voting, or remote monitoring. In today's world, IoT devices have various advantages, including decreased operating costs and the ability to boost the efficiency of any firm by automating everything. IoT devices, for example, may manage, monitor, and warn the personnel of any business in the event of any changes in processes by assisting them in making wiser decisions. At the same time, these gadgets have the following drawbacks.

Security and Privacy: IoT devices convey sensitive information about a person, group, or organization. As a result, the emphasis remains on securing the data they contain, with security and privacy playing a key role. It is never easy to keep the data collected and transferred by these devices secure. Although data security is a top issue, IoT devices aren't typically included in the plan. Another issue is data privacy, especially as IoT devices are being employed in more sensitive sectors such as health care and banking. Globally, information privacy regulations are also taking effect, which means that not only does it make excellent commercial sense to secure data, but organizations are also legally compelled to do so. Connectivity and power consumption: Many devices rely on the internet and constant electricity to work effectively. When either fails, the gadget and everything attached to it fails. Because IoT devices are so entwined with today's organizations, when they go weaker than expected, everything comes to a standstill. IoT devices must process massive amounts of data, yet their computing power is limited.

Integration: It has grown challenging to integrate many security and privacy approaches. To safeguard the data, anybody must be aware of who is accessing the data. As a result, authenticating the persons who access the data is critical. The approaches used to authenticate the user either need the usage of expensive hardware or rely on a third party. As a result, integrating IoT devices with third-party or heavy hardware has become challenging. Resource-constrained devices are more likely to be embedded devices, indicating that resources are limited as compared to standard computing devices, mobile phones, laptops, and desktops. They are constrained by critical computational resources, storage, the networking environment, and energy usage. These laptops and PCs are far more powerful than IoT devices and are thus employed for several purposes. A significant proportion of IoT devices are embedded and resource constrained. Smart sensors, controllers, and activity trackers are a few examples. IoT works with a massive amount of information and creates an enormous quantity of information that must be saved but has small storage capacity. As a result, the data is stored on edge gateways, endpoint devices, and the cloud. As a result, IoT devices are frequently designed to be resource-constrained, with computing capabilities but limited storage space. As a result, because they lack proper safeguards, IoT devices are prone to security breaches. To prevent IoT devices from being subject to security threats, they must be verified in a timely way when executing any information exchanges.

5 Literature Review

Internet of things (IoT) devices are small non-standard computing devices that are connected to wireless connections over the network and have the ability to transfer data across devices on the internet. IoT devices include embedded devices such as wireless network sensors which communicate over a large network. In essence, the purpose of the internet network's security system is to preserve the transmitted data so that users can benefit from elements of confidentiality, integrity, and availability. Data that is transferred may be statistical, financial, private, etc. A key component of secure IoT systems is mutual authentication between IoT servers and devices. Since the server and the IoT system are connected over a wide network, it is a necessity for the IoT device to provide the proof of their identification to the server. Since these devices are connected remotely, they are more vulnerable to security threats.

Traditional authentication mechanisms are not suitable for Iot devices as they lack the required resources for performing the authentication through these mechanisms. Hence in our work, we present a study of various authentication mechanisms that can be used for IoT devices. According to our study, multigraph Zero knowledge protocol [1] is the best suited protocol for providing authentication in IoT devices. Multi-graph systems provide multiple personal graphs for one user and thus helps in preventing impersonation of the prover or any sorts of forgery attacks. Evaluation and experiment results performed by the authors of [1] show that the M-ZAS outperforms other existing authentication systems while providing the same level of security. The authors of [2] proposed a graph based method of providing privacy and data integrity and end-to-end entity authentication among various IoT devices among networks. The only difference between the graph and the multigraph system is that, graph uses only a single graph per user, whereas the multigraph system uses multiple graphs per user. This provides the reduced transmission overhead for ZKP iteration. There has been a study of various mechanisms for providing authentication for IoT devices, but the best suitable mechanism is ZKP [3].

6 Authentication in IoT

Authentication is the process of validating a user or device before granting access to a system or resources. It is a methodology for building trust among IoT equipment and devices when information is transmitted across an unprotected network. As a result, robust authentication is necessary to ensure that connected devices on the Internet of Things may be trusted to be what they claim to be. Administrators may also utilize this to maintain track of each device throughout its life cycle, interact securely with it, prohibit it from performing hazardous processes, and protect it from unauthorized users or devices. Because authentication is primarily required for IoT devices, it must be distinct and significantly lighter than the present user or personal authentication method to ensure robust security.

6.1 Different IoT Authentication methods

6.1.1 Single/One-factor authentication

One of the most basic kinds of authentication, single or one-factor authentication just requires the user to provide anything in order to validate their identity. Usernames and passwords, security questions, and pins are some instances of single-factor authentication. The effectiveness of this authentication mechanism is mostly determined on the strength of the password that the user chooses. Weak passwords are vulnerable to security assaults; thus, a strong password that includes capital letters, tiny letters, digits, and special characters is less vulnerable to security attacks from any intruder. It saves time when authenticating any IoT device or user since it certifies the user in a single step. Furthermore, because this method certifies your device in a single step, it is easy to use and has a high user usage rate when compared to other methods.[4] While single authentication has its advantages, it also has certain downsides, such as being particularly vulnerable to security threats and having a high data breaching rate when compared to alternative techniques. Because passwords and pins are easily stolen, real-time security is compromised. It restricts the user's access to specific devices. If the user assigns a password to any device, their access is restricted to that device. The biggest disadvantage of this method of authentication is the use of the same password multiple times. The use of the same password for various devices in the system can easily compromise the system, putting other systems in the environment in danger. Single-factor authentication is subject to phishing and brute force attacks, theft, and keyloggers.

6.1.2 Two-Factor Authentication

When compared to single-factor authentication, this approach adds an extra layer of protection by verifying the user based on two of the three elements, which are:
something they are aware of

Something they have
something that they are
The user's application sends a request to the authentication service, which validates the user's identity and issues a token, after which the user is able to use the application. The most common forms of two-factor authentication include the use of OTPs, biometrics, face recognition, and so on.

This approach is more secure than single/one factor authentication since it needs two steps to authenticate any person or device. At the same time, the increased number of procedures for confirming the user increases time consumption. Integration becomes tough since IoT devices are resource restricted, and this solution relies on third-party services or hardware. Many dependence difficulties can be detected as a result of its reliant behavior, as external services cannot be regulated. When IoT devices are coupled with biometrics or facial recognition systems, a maintenance issue is also encountered. Recently, it has been seen that biometric authentication is failing as new approaches to engage the user in breastfeeding the authentication process are not being implemented with IoT devices that are free of vulnerabilities that might be abused by adversaries to acquire unlawful access. In addition, new ways for safeguarding biometric reference templates must be implemented with IoT devices so that these templates do not seem to be compromised by hackers at the remote server level.

6.1.3 Multi-Factor Authentication

Authentication with many factors - MFA takes security to the next level by combining several authentication processes to authenticate any IoT device or user. It employs all three criteria, which are:
Something they know (passwords, pins, security questions)
Something they have (OTP, authenticator applications)
Something they are (Fingerprints and Facial recognition)
As this method uses multiple mechanisms, it requires more time to authenticate any device or user, hence, time consumption is increased. Because of multiple mechanisms, it adds friction in user experience which might lead to poor adoptions in IoT systems.

6.2 Possible attacks on IoT authentication

6.2.1 Sniffing

IoT devices carry sensitive information, and to authenticate these devices, we need to provide something we know, something we have, and something we are, there is a high chance that the private information of any user or device is sniffed and also that someone other than the actual user can impersonate a user.

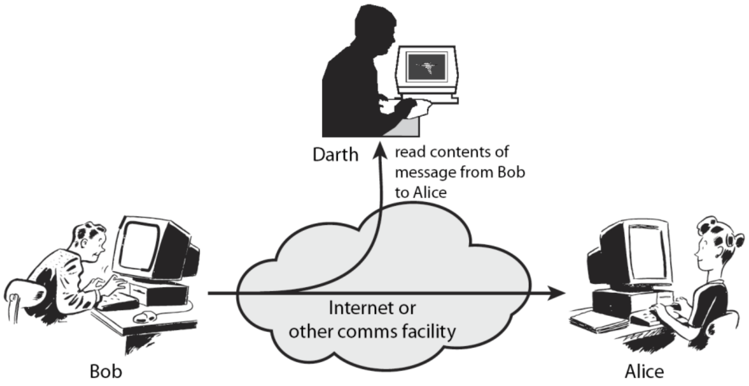
Darth
read contents of message from Bob to Alice
Internet or other comms facility
Bob
Alice

Figure 1: Sniffing

6.2.2 Cloning attack

An adversary can technically seize the equipment, collect critical information, replicate the devices, and strategically deploy them in desired areas to perform various insider assaults via a cloning technique.

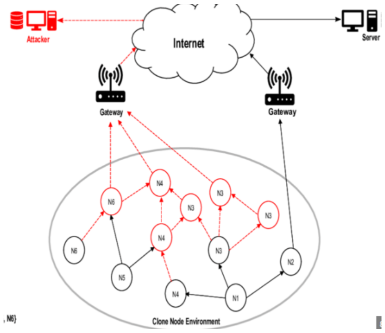

Figure 2: Cloning

6.2.3 Reusing of same passwords

Same password can compromise the whole system, If an attacker can exploit any one device of the IoT system and since passwords are reused, the attacker can easily steal the password for one device and try to compromise the entire environment. Due to the reusing of passwords, IoT systems are highly prone to replay attacks.

[5]

7 Why use ZKP in IoT authentication?

It is possible to demonstrate a fact without disclosing anything about the proof other than its veracity. Zero-knowledge proofs allow for such proofs. Zero Knowledge protocols are cryptographic protocols in which no secret information is revealed throughout the operation. The verification party cannot masquerade as the prover to any third party since the secrets are not given to the certifying party.

7.1 Background of ZKP

Goldwasser, Micali, and Rackoff developed the idea of zero knowledge in 1985. It is a cryptographic protocol that explains how to minimize the amount of information exposed to the verifier when proving the algorithm in order to secure the information.

7.2 What are Zero-knowledge proofs

It is a cryptographic protocol that mainly deals with two participants, namely Prover and Verifier. When the prover has some secret value X, then the main goal of this proof is that the prover has to prove that the prover knows the secret value without revealing any information about the secret to the verifier [6].

Zero Knowledge Proofs
Prover
Secret Data & Proofs
Verifier

Figure 3: Zero Knowledge Proof

7.3 Example of Alibaba Cave

Consider the example of Alibaba cave, where we have a Alibaba cave,which is only opened by a secret password. In this example, we have two persons named Pranjali and Vyshnavi. Pranjali has to prove Vyshnavi that she knows the secret password of the cave to the vyshnavi. The graphic below represents Alibaba's cave, which has two entrances (left L and right R) and a secret passageway between them. Hence both of them, tried to implement an exercise in order to verify if Pranjali knows the secret. Starting at point A Pranjali

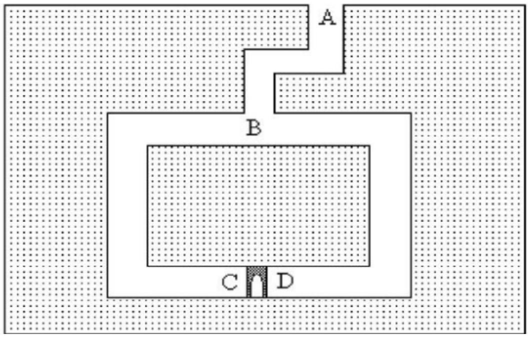

Figure 4: Zero Knowledge Proof

walks all the way to point C or point D. Vyshnavi walks till point B and requests pranjali to either come out either from passage C or from passage D. Pranjali can use the secret password if required to open the door and can come out of the passage to prove vyshnavi that she knows the secret. She will have to repeat the process until vyshnavi belives that pranjali knows the secret. Even if the process is totally completed and vyshnavi belives that pranjali knows the secret password, vyshnavi will never be able to know the secret password. This is the main reason to use zero-knowledge proofs, as the verifier will never be able to know the secret and the secret will always be anonymous.

7.4 ZKP Conditions

Completeness: If the assertion is true, the honest verifier, who is following the procedure correctly, will be convinced by the honest prover - In the example provided, Since pranjali knows the secret, she can successfully finish the proof.

Soundness: If the assertion is incorrect, no cheating prover can persuade the verifier that it is true - In the example provided, if Pranjali does not know the secret, she cannot prove anything to Vyshnavi.

Zero-knowledge - Even though the total proof is completed, Verifier can never know the secret. In the above example, pranjali was able to convince vyshnavi that she knows the secret but vyshnavi will never know the secret.

Repudiable - even if Vyshnavi records whole process she will not be able to prove that Pranjali knows the secret to any third party.

Non-transferable - Vyshnavi cannot use the proof to pretend to be the prover to the third party.

7.5 Types of ZKP

There are two types of Zero knowledge proofs which are mainly Interactive zero- knowledge proof and Non-interactive zero-knowledge proofs.
[7] Interactive Zero-knowledge proof: In this method, the prover and the verifies interacts with each other multiple times, inorder to prove the verifier that the prover knows the secret. Verifier sends the challenge to the prover and prover has to prove the verifier that he knows the secret until verifier is convinced with the prover. Computation power is more in interactive ZKP as there are multiple interactions.

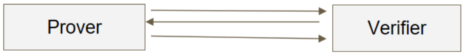

Figure 5: Interactive Zero Knowledge Proof

Non-Interactive Zero-Knowledge proofs: In this method, the prover and verifier interact only once and the verifier sends the challenge to the prover. The prover will get only one chance to prove that he knows the secret. In this computational power is less as the number of interactions is only one.

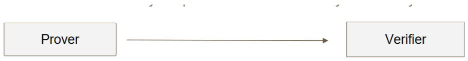

Figure 6: Non-Interactive Zero Knowledge Proof

7.6 ZKP of Discrete Logarithmic Problem Proofs

The proof for ZKP of Discrete Logarithmic Problem was mentioned by [2] Suppose, the equation is $b = a^x$ then x will be the discrete log to base a of c. Given that any p that is a large prime number,b any number in Z_p, and g which is a generator for multiplicative group Z_p, the Prover(P) will have to prove to the Verifier(V) that they know the value of x with revealing the value such that $g^x = b(modp)$.

7.6.1 Iterative ZKP of Discrete log problem

In 7, initially the Prover and the Verifier both know the values of g, b, p. Only the Prover knows the secret value of x. The Prover starts by sending $h = g^r(modp)$ where r is a

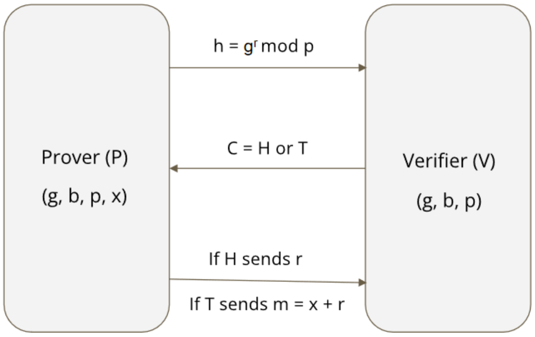

Figure 7: Iterative ZKP of Discrete log problem.

random seed value. After receiving this value of h the Verifier chooses between HEADS(H) or TAILS(T) and sends their decision to the Prover. According to the decision made by the Verifier the Prover will need to send the secret with a respond to the Verifier i.e. they send r if HEADS was selected and $m = x + r$ if TAILS was selected. Therefore, if the Prover is lying and does not have the value of x, they can only cheat for either one of the two outcomes. These steps keep on repeating until the Verifier is satisfied by the fact that the Prover has the secret value x. This has the probability of $1 - 2^{-k}$ where k is the number of iterations.

7.6.2 One-Round ZKP of Discrete log problem

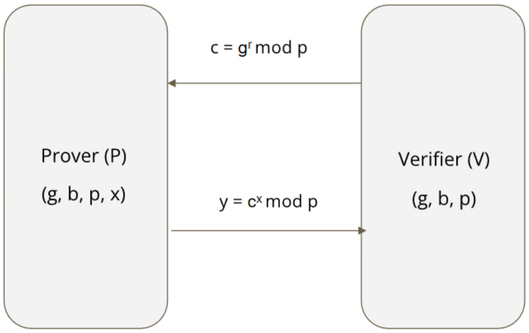

Figure 8: One-Round ZKP of Discrete log problem.

One-Round ZKP is the improved version of the Iterative ZKP where all the steps happen only once. Here in 8, instead of the Prover the Verifier starts by sending $c = g^r(modp)$ to the Prover. The prover after receiving the value of c calculates $y = c^x(modp)$ and sends the reply to the Verifier. The Verifier now checks is the received reply is $y = b^r(modp)$ and only accepts if it is True. As the number of rounds have reduced to one, the execution time also reduces and there is less communication cost as the number of exchanges over the network is less.

8 Different ZKP Protocols

There are various classical problems that involve ZKP. These are Discrete Logarithm Problem, Square root of an integer modulo n, Graph Isomorphism, Integer Factorization etc. These belong to a class of NP problems. Among these we compare and discuss the Discrete Logarithm Problem and the Graph Isomorphism Problem.

8.1 ZKP with Elliptic Curve Discrete Logarithm Problem (ECDLP-ZKP)

There are a wide variety of Zero Knowledge Proof Protocols based on Discrete Logarithm Problem (DLP). Zero Knowledge Proof can be done over arbitrary cyclic groups. One such example for this is ZKP with Elliptic Curve Discrete Logarithm Problem (ECDLP)[8]. The ECDLP problem is as follows, given an elliptic curve E over a field F of order n, generator point $G \epsilon E \mid F_n$ and a point $B \epsilon E \mid F_n$ it is computationally hard to find x such that $B = x.G$. Same level of security is achieved with ECC as with other public key cryptosystems except with a smaller key size. This makes it easier for use in low constrained IoT devices as it reduces the memory use and needs less computational time. Many ECDLP protocols have been proposed previously, the brief discussion of some are done below.

8.1.1 ECDLP ZKP with Coin Flip

The elliptic curve of one the original ZKP protocol presented in [8] is as follows: given an elliptic curve E over a field F of order n, generator point $G \epsilon E \mid F_n$ and a point $B \epsilon E \mid F_n$ the Prover wants to prove that they know the secret value of x without revealing it such that $B = x.G$ In (figNo), the Prover(P) initially calculates the value of A by multiplying the generator point(G) with a random seed(r) and sends it to the Verifier(V). After the Verifier receives this value, they in return choose either HEADS(H) or TAILS(T) and send it to the Prover. If the Prover receives H they send r else send $m = x + r(modn)$. In case of H the Verifies checks that $r.G = A$ else checks $m.G = (x + r).G = x.G + r.G = A + B$. These steps are similar to the Iterative Discrete Log Protocol in (ref) and need to be repeated till the Verifier is convinced that the Prover knows the value of x. This has the probability of $1 - 2^{-k}$ in which k is the number of iterations. A Prover that is dishonest and doesn't know the value of x can be prepared for only one of the two outcomes and hence the probability of them cheating is 1/2.

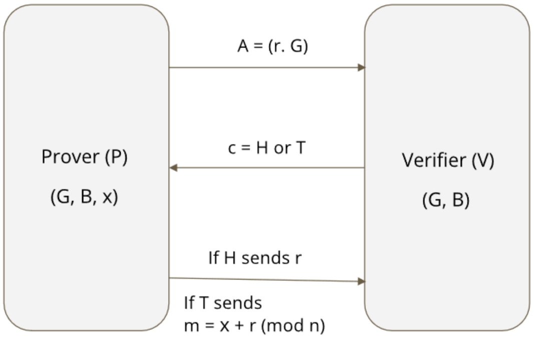

Figure 9: ECDLP ZKP with Coin Flip

8.1.2 ECDLP ZKP with Schnorrâs Protocol

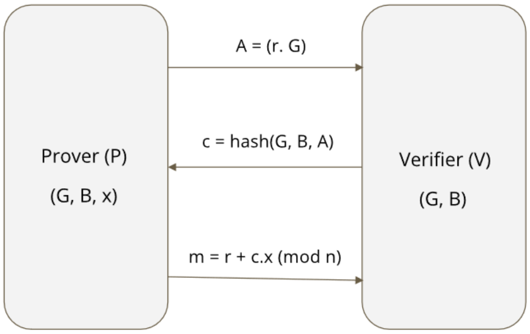

Figure 10: ECDLP ZKP with Schnorrâs Protocol

The ECDLP Schnorr′s Protocol is an enhanced version of the previous protocol in 9. The problem statement for this is same as the Coin-Flip Problem with the only difference being that it needs to be executed just for one round in contrast with the Coin-Flip Problem. The Verifiers coin flips are done using a hash function that is only known to the Verifier, that is, $c = HASH(G, B, A)$. This use of hash function ensures that the Prover who does not know x cannot cheat.

Although, this protocol is an improvement to the previous one it has a major issue of being computationally intensive. This may cause a problem with IoT implementation as it will be infeasible to be used in many low constrained IoT devices.

8.2 Goldreich-Micali-Wigderson(GMW-ZKP)

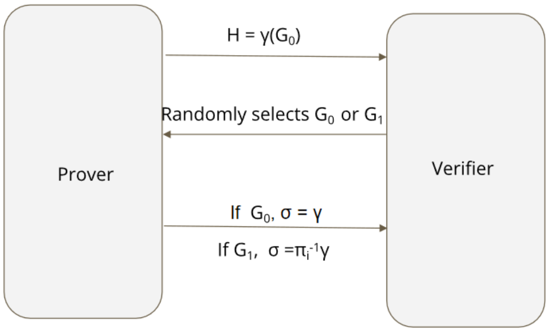
$H = \gamma(G_0)$
Randomly selects G_0 or G_1
Prover
Verifier
If G_0, $\sigma = \gamma$
If G_1, $\sigma = \pi_i^{-1}\gamma$

Figure 11: Goldreich-Micali-Wigderson (GMW) ZKP

Goldreich-Micali-Wigderson Protocol[9] has it's basis in Graph Isomorphism. Graph Isomorphism is an NP Problem. The protocol statement is as follows: Given two graphs $H = (V_1, E_1)$ and $G_0 = (V_2, E_2)$ having the same number of vertices will be isomorphic if there is a permutation π_0 on vertices of H, such that they can be mapped on G_0. In this protocol the secret is the graph permutation π. At the beginning of the round the Prover generates a random permutation and transforms one of the two graphs. They then send this newly created graph, which is known as the Initiation Graph, to the Verifier. After which the Verifier randomly picks either G_0 or G_1 and send the selection to the Prover. Depending on the randomly picked graph by the verifier the Prover may or may not need to send the send in the next message to convice the Verifier that they have the secret permutation.[10]

GMW overcomes the limitations of the ECDLP Protocols mentioned in (ref) that is, it is less computationally intense and uses less energy. But on the other hand has more transmission overheads which can still cause problems in some IoT devices.

9 Multi-Graph Zero Knowledge Proof Authentication Systems [M-ZAS]

IoT devices are heterogeneous devices with limited resources and dynamic network topology that are used for various services that carry personal and sensitive information about a user. Thus security and privacy are essential concerns for these IoT devices. But because of limited capabilities it becomes difficult or in most cases impossible to apply traditional security mechanisms to these IoT devices and thus makes them prone to various threats and attacks such as impersonation and forgery. Hence Multigraph Zero-Knowledge-based [M-ZAS] Authentication system was introduced in [1], that provides security and ensures user privacy for the IoT devices.

Multigraph authentication systemâs high adaptiveness and light weightness in nature makes it suitable for IoT devices and it provides better security as well as higher performance as compared to other security mechanisms. M-ZAS works on the concept of zero Knowledge proof for graph Isomorphism. The overall ideology of this architecture is that the prover needs to prove to the verifier the possession of some secret without actually revealing to the verifier any other information about the secret. The verifier issues a challenge for the prover and the prover must respond to that challenge with the correct response. The verifier can authenticate the prover based on its response and can decide whether to trust or not the prover. The challenge-response procedure is considered as one round and the verifier can perform multiple such rounds until it is satisfied with the proverâs identity. The ZKP procedure should ensure that no information is revealed about the prover in this whole procedure.

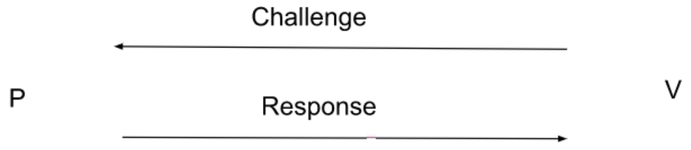

Figure 12: ZKP Challenge-Response

Along with ensuring that the verifier does not gain any information about the prover, ZKP also assures that the verifier does not cheat the prover as it does not contain any information about the proverâs secret and is thus, unable to calculate itâs secret. Since the challenge is repeated until the verifier is convinced of the proverâs identity, the prover is unable to cheat the verifier as well.

9.1 How is M-ZAS different from GMW-ZKP?

In the GMW-ZKP protocol, the prover is responsible for publishing two public graphs called G_0 and G_1 where, G_0 being the generator graph which is used for generating G_1, and G_1 representing the user's personal graph. The relation between G_1 and G_0 is $G_1 = \pi(G_0)$ 11. Here π is the one secret permutation that a prover possesses in advance. The prover's secret π in the GMW protocol is a graph permutation, which is the isomorphism between two publicly available graphs, G_0 and G_1 [3]. In ZKP, the prover needs to prove to the verifier, possession of some secret without revealing the secret. Thus, in GMW-ZKP, the prover would like to prove to the verifier about the possession of the secret permutation of Ï, while maintaining complete anonymity of any information about π.

It can be observed that for every attestation round, there is 50% chance that π is not involved in σ. If the challenge graph is G_0, then an impersonator pretending to be a prover who does not have any idea about original proverâs secret permutation π, has a chance to successfully pretend to be a prover as the original proverâs secret permutation π is not involved in the verification process. Thus, if n attestation rounds are performed, then GMW-ZKP achieves a security level of $1 - (1/2)^{-n}$. Because of this, the transmission overheads of GMW-ZKP are substantially higher than those compared to other ZKP systems.

M-ZKP overcomes the issue of higher transmission overheads of GMW-ZKP while maintaining the same security levels by decreasing the number of attestation rounds. Reducing the number of attestation rounds allows M-ZAS to mitigate the large transmission overheads and thus makes them highly adaptive and suitable candidates for light-weight IoT devices. To successfully decrease the total number of attestation rounds while maintaining its security levels, it is required to reduce the likelihood that fake provers without knowing the secret permutation of the original prover can successfully imitate the actual one in a single attestation round. To achieve this, M-ZAS protocol, increases the number of personal graphs for every user in a single attestation round, that is, it lets every user have possession of multiple personal graphs instead of a single one as done in G-ZKP. This decreases the probability of a fake prover to impersonate the original prover in every attestation round. Authors in [1] are able to prove that M-ZAS not only reduces the consumption of computational resources but is also able to provide 3 times faster results than GMW-ZKP and is 7 times faster than traditional authentication mechanisms in IoT devices. Because of decreased attestation rounds, M-ZKP is able to deduce the network traffic by 3 times as compared to the GMW-ZKP method. The M-ZAS is typically considered as a more practical mechanism for IoT environments than other centralised techniques because it doesn't require a need for a centralised controller.

Let us first understand the system Architecture of M-ZAS. Multigraph zero-knowledge-based Authentication is based on Graph-ZKP, where the only difference between the two is the number of personal graphs of the users. The G-ZKP proposes only one single graph G_1 as the userâs personal graph, whereas M-ZAS proposes a multigraph system for the user's personal graph. Overall, M-ZAS works by two main procedures :-

1. Multi-graph Zero-knowledge Proof (M-ZKP): This is responsible for offering lightweight authentication services that are quicker than other ZKP procedures and a lot quicker than the conventional Elliptical-Curve-Cryptography (ECC)-based method.

2. Adaptive Security Configuration (ASC) procedure: This technique is responsible for generating appropriate settings for M-ZKP procedure by taking into consideration pertinent circumstances, such as user requirements or device capability, as well as analysing certain security concerns.

Multi-Graph Zero-Knowledge-based Authentication M-ZAS comprises 4 major components as depicted in fig 1. These components are M-ZKP Prover, M-ZKP verifier, System Setup and Adaptive Security Configuration (ASC). The system setup component is responsible for setting up the default configurations of the public graphs and the secret permutation of the users of M-ZAS and storing these configurations into a security information database. The information about the public graphs can be conveyed to all the connected IoT devices of the system. The task of carrying out the right ASC procedures to assign the M-ZKP verifier with the correct settings falls under the preview of the Adaptive Security Configuration component. Depending on the user's functionality, the M-ZKP prover and the M-ZKP verifier component are responsible for carrying out the M-ZKP procedure.

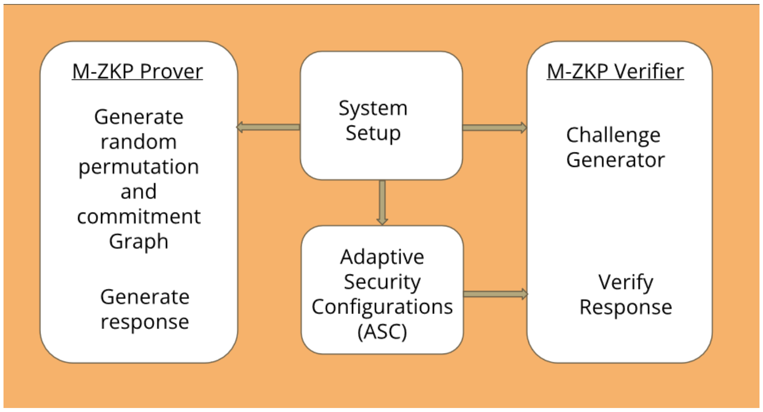

Figure 13: System Architecture of M-ZAS

As discussed earlier, M-ZAS works by providing multiple personal graphs for the user rather than one single graph as it is done by GMW-ZKP. The prover selects a random number Np and generates Np number of secret permutations π_i (where i ranges from 1 to N_p). As done in GMW-ZKP the prover generates a generator Graph G_0 but rather than generating one single personal Graph G_1, it computes G_i number of personal graphs where i holds the value from 1 to N_p. Thus overall, Np personal graphs plus one generator graph are generated by the prover. The relation between the personal graph, the generator graph and the permutation secret is as follows: $G_i = \pi_i(G_0)$

9.2 System Flow of M-ZAS

Initially, to begin with M-ZAS procedure, the system setup component starts setting up the public graphs G_0 and G_i, and the secret permutations π_i in advance (fig 2). Whenever a verifier wants to verify a proverâs authenticity, the verifier will begin applying the ASC procedures that would begin setting up the parameters for that particular session. The prover and verifier proceed with the M-ZKP method in accordance with the acquired ASC criteria for carrying out the identification and authentication process. As mentioned in the previous section, the prover selects a random number N_p and generates Np number of personal graphs and secret permutations. However, because of limited storage capacity, the verifier is unable to store N_p number of graphs. Thus, the verifier will only store $N_p(u)$ number of the personal graphs out of N_p total number of graphs[1].

Prover P,

- Generates π_i where i = 1 to N_p (i.e $\pi_i ... \pi_{N_p}$)
- Generate Public Graph G_i such that $G_i = \pi_i(G_0)$
- Overall graphs = $(N_p + 1)$
- At the Verifier's end only a $N_p(u)$ number of personal graphs are stored

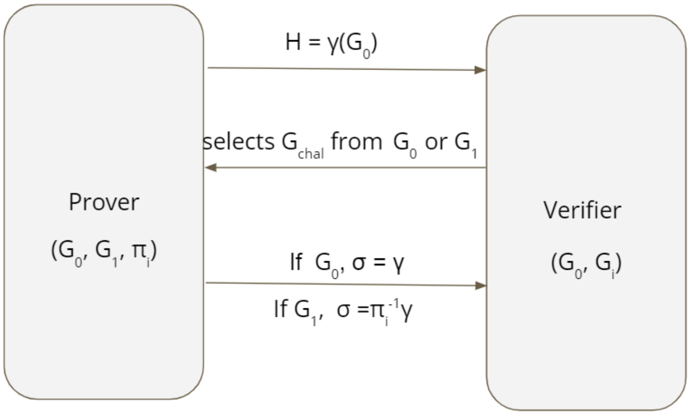

Figure 14: System Architecture of M-ZAS

The System flow of M-ZAS takes place in the following steps:

1. Prover selects random number N_p to generate π_i and G_i (where i ranges from 1 to N_p).
2. The prover generates a generator graph G0 which is used for generating Np personal graphs such as $G_i = \pi_i(G_0)$.
3. The prover generates a random permutation γ, and by using this permutation and the generator graph the prover generates a commitment graph H such that $H = \gamma(G_0)$.
4. Later, the generated commitment graph H is sent to the verifier.

5. Once H is received by the verifier, it chooses a challenge graph among G_0 or $G_{N_p(u)}$ as $G_{chal}(i)$ and informs the prover.

6. Based on the challenge Graph $G_{chal}(i)$, a response permutation Ï is sent by the prover to the verifier such that $H = \sigma(G_{chal}(i))$.

7. If $G_{chal}(i) = G_0$, then $\sigma = \gamma$

8. If $G_{chal}(i) = G_i$, then $\sigma = \sigma = \pi_i^{-1}\gamma$

After that the verifier can then validate the prover by determining whether $H = \sigma[G_{chal}(i)]$ holds true or not.

Since only $N_p(u)$ number of graphs are stored as the challenge graphs, the probability that an impersonator can successfully pretend to be the original prover reduces to $1/[N_p(u) + 1]$ for a single attestation round. It can be observed that for every attestation round, there is $1/[N_p(u) + 1]$ chance that π is not involved in σ. If the challenge graph is G_0, then an impersonator pretending to be a prover who does not have any idea about original proverâs secret permutation π, has a chance to successfully pretend to be a prover as the original proverâs secret permutation π is not involved in the verification process. Therefore, the number of verification rounds M-ZAS should perform reduces to $m/log_2[N_p(u) + 1]$ compared with m number of rounds for GMW-ZKP.

9.3 Security Analysis of M-ZKP

In M-ZKP, an impersonator has $1/[N_p(u)+1]]$ chance of pretending to be the real prover and passing the verification successfully without knowing the actual proverâs secret permutation. So let us analyse the ways an impersonator can try to break into the M-ZKP system. There are two ways any fake prover can successfully pass the verification. First, by acquiring the real proverâs secret permutation Ï through brute force attack. If a given graph has v vertices, then the permutation secret can be stated as a sequence of numbers from 1 to v due to the fact that the secret permutation is a mapping from the generator graph to the personal graph [i.e. $Gi = \pi_i(G_0)$]. Therefore, an impersonator must predict $v!$ different types of series to match the security level offered by a $log_2(v!)$-bit symmetric key. For example, around a 128 bit symmetric key can be generated for a generator graph composed of 34 vertices.

The second way to pass the verification is by figuring out the challenge sequence, which is made up of the indices of each challenge graph that the verifier has chosen. This is basically predicting the challenge graph $G_{chal(i)}$ of the verifier before producing the commitment graph (H). If the challenge graph is known beforehand, then based on that information the impersonator can build the commitment graph such that $H = \gamma(G_i)$ instead of generating it with the generator graph $H = \gamma(G_0)$. Therefore, the response permutation γ of the prover will not include Ï and thus will always be true. Hence, the imposter will always be able to successfully get through the verification process. So, for the first method of obtaining the secret permutation of the real prover through brute force attack, for one secret permutation pie the impersonator has $v!$ possibilities. Hence, for π_i permutations the possibilities for guessing the correct sequence of the secret permutation is increased by a very high number. For the second method of predicting the challenge graph, the number of personal graphs generated for the prover will be N_p and the number of attestation rounds for the M-ZKP procedure will be m. Thus, the soundness probability of a fake prover to predict the challenge sequence is $\delta = (N_p + 1) - m$. Therefore, this increases the challenge for the impersonator and thus reduces the likelihood of M-ZAS systems to be compromised.

10 Our Approach

The current M-ZKP approach uses the method of iterative protocol for ZKP which we have seen for discrete log problems [5]. In this, for every attestation round, the prover will generate a random value r and will use this value to calculate $h = g^r modp$ 7. Then the calculated value h is sent to the verifier. After receiving h from the prover, the verifier will flip a coin and send the result to the prover (i.e $C = HorT$). If the result of the coin flip is H, then the prover will send the generated random value r to the verifier and the verifier will then check if $g^r = h$. If the result of the coin flip is T, then the prover will send $m = x + r$ to the verifier and the verifier will check if $g^m = b * h$. If the value holds true then the prover will be authenticated by the verifier. These steps are repeated until the verifier is convinced that the prover acquires the knowledge about x, and in any of these steps no information about x is revealed to the prover. But here, for every attestation round any fake prover has 1/2 a chance to pretend to be the real prover without knowing its secret. This issue was resolved by one round ZKP as explained in the earlier section 8.

M-ZKP also follows the similar mechanism which we have explained in the previous section. Although, M-ZAS system reduces the number of attestation rounds as compared to other protocols (GMW-ZKP), it still uses multiple rounds which creates a repetitive process for verifying the prover. Therefore we propose a single round ZKP approach for M-ZKP authentication systems. In this proposed mechanism, appropriate ASC parameters will be calculated for the verifier's end rather than being set up for the prover.

10.1 System flow of one round M-ZKP

The initial setting up process for one-round ZKP will be similar to that of M-ZAS system which we have discussed earlier, where the system setup will be responsible for setting up the Public Graphs (G_0 and G_i) and the secret permutation π_i in advance (As shown in fig 2). But, the M-ZKP procedure for the proposed single round method will be different from the previous method. In this, when the verifier wants to verify the proverâs authenticity, it will begin applying the ASC procedure that would begin setting up the parameters for that particular round. The verifier will generate a random permutation value, and will derive a commitment graph from this random permutation value. Then the verifier will send this generated commitment graph to the prover, where the prover will use this commitment graph and its own permutation secret (which is known only by the prover) to compute a value X. Then the prover will send this computed X to the verify, where the verifier will check if the calculated value holds true or not and authenticate the proverâs identity based on the verification.

The System flow of M-ZAS takes place in the following steps:

1. Prover selects random number N_p to generate $\ddot{I}_i$ (where i ranges from 1 to N_p).

2. The prover generates a generator graph G_0 which is used for generating Np personal graphs such as $G_i = \pi_i(G_0)$.

3. Then for the M-ZKP procedure, the verifier will generate a random permutation γ and will use this permutation to generate a commitment graph H such that $H = \gamma(G_i)$.

4. Later, the generated commitment graph H is sent to the prover.

5. Once H is received by the Prover, it will compute a value X using the commitment graph H and its own secret permutation π, such that $X = H\pi_i^{-1}$.

6. The computed value of X is sent to the verifier where it will check if $X' = \gamma(G_0)$, and compare the value of X and X'.

7. If $X =$ XÕX', then verifier will be convinced that the prover possess knowledge of its secret permutation and thus will authenticate the prover.

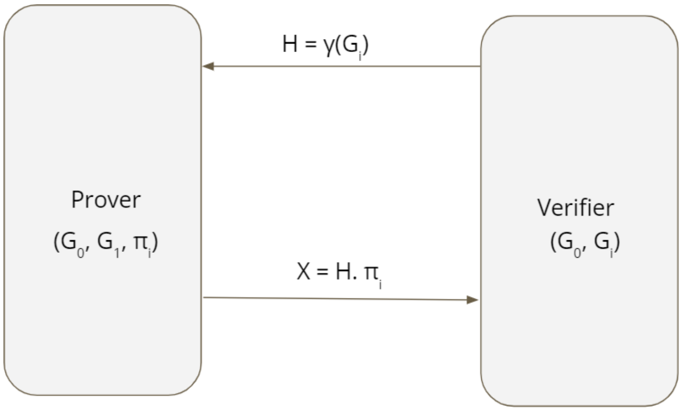

Figure 15: System Architecture of M-ZAS

10.2 Why One-Round M-ZKP?

M-ZAS was able to overcome the restriction of a higher number of transmission overhead by reducing the number of attestation rounds by lowering the probability that a fake impersonator would pose as the true prover without being aware of the real prover's secret permutation. Although M-ZAS lowers transmission overhead, it does not entirely eliminate the danger of an impersonator passing the verification without ever being aware of the permutation secret of the genuine prover. So, in our viewpoint, this problem can be handled by utilising just a single round of M-ZKP. Moreover, the single round protocol will further reduce the number of attestation rounds to a much greater extent and will be able to produce a better execution-time complexity. It will nullify the need for local computation (at the proverâs end) and thus will be beneficial in reducing the overall communication cost of the system. This will lower the system's overall network traffic because fewer messages will need to be sent over the network.Additionally, the prover and the verifier will exchange significantly less data in terms of bits and latency, which will result in speedier outcomes as compared to other ZKP mechanisms. Overall, we think that the one-round M-ZKP technique can offer a greater accuracy rate for user authentication.

11 Future Work

Although our study considers the single round ZKP to be highly suited for low constrained IoT devices, more research on this protocol is necessary to fully substantiate this claim and prove its legitimacy and effectiveness. In the future, we'll aim to provide more comprehensive information on our suggested system and attempt to evaluate its performance by contrasting it with the performance of existing ZKP protocols in terms of computation resources, communication, and storage. Additionally, we will strive to examine the security analysis of this mechanism and present a threat model for the suggested system. Also, further work may be done to incorporate the one-round M-ZKP systems into the real world IoT applications.

12 Conclusion

Having a light weight solution for authentication in low constraint IoT devices is a major challenge. Although this can be overcome by using may different protocols that are previously proposed. Therefore, by comparing all the discussed protocols, we conclude that the M-ZKP protocol is best suited for low constraint IoT devices as it not only reduces the consumption of computational resources but also provides less transmission overhead and is able to produce results at much faster rates.

References

[1] I.-H. Chuang, B.-J. Guo, J.-S. Tsai, and Y.-H. Kuo, "Multi-graph zero-knowledge-based authentication system in internet of things," in *2017 IEEE International Conference on Communications (ICC)*, 2017, pp. 1–6.

[2] A. Sardar, S. R. Y.V., and R. N., "Zero knowledge proof in secret sharing scheme using elliptic curve cryptography," in *Global Trends in Computing and Communication Systems*, P. V. Krishna, M. R. Babu, and E. Ariwa, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 220–226.

[3] A. Rasheed, R. R. Hashemi, A. Bagabas, J. Young, C. Badri, and K. Patel, "Configurable anonymous authentication schemes for the internet of things (iot)," in *2019 IEEE International Conference on RFID (RFID)*, 2019, pp. 1–8.

[4] H. A. Aronsson, "Zero knowledge protocols and small systems," 1995.

[5] B. Soewito and Y. Marcellinus, "Iot security system with modified zero knowledge proof algorithm for authentication," *Egyptian Informatics Journal*, vol. 22, no. 3, pp. 269–276, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1110866520301560

[6] A. Jaafar and A. Samsudin, "Visual zero-knowledge proof of identity scheme: A new approach," *Computer Research and Development, International Conference on*, vol. 0, pp. 205–212, 05 2010.

[7] K. T. Nguyen, M. Laurent, and N. Oualha, "Survey on secure communication protocols for the internet of things," *Ad Hoc Networks*, vol. 32, pp. 17–31, 2015, internet of Things security and privacy: design methods and optimization. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1570870515000141

[8] I. Chatzigiannakis, A. Pyrgelis, P. Spirakis, and Y. Stamatiou, "Elliptic curve based zero knowledge proofs and their applicability on resource constrained devices," 07 2011.

[9] P. Flood and M. Schukat, "Peer to peer authentication for small embedded systems: A zero-knowledge-based approach to security for the internet of things," in *The 10th International Conference on Digital Technologies 2014*, 2014, pp. 68–72.

[10] O. Goldreich, "Zero-knowledge twenty years after its invention," *Electron. Colloquium Comput. Complex.*, vol. TR02, 2002.