International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING

ISSN:2147-6799 www.ijisae.org Original Research Paper

Designing a New Hashing Algorithm for Enhancing IoT Devices Security and Energy Management

Teba Mohammed Ghazi Sami[1], Subhi R. M. Zeebaree[2], Sarkar Hasan Ahmed[3]

Submitted: 08/09/2023 **Revised:** 22/10/2023 **Accepted:** 06/11/2023

Abstract: The Internet of Things (IoT) is a rapidly growing network of interconnected devices that collect and exchange data. While IoT devices offer many benefits, they also pose unique security and energy management challenges. One promising approach to addressing these challenges is to develop new hashing algorithms that are both secure and energy-efficient. This paper proposes a new hashing algorithm for IoT devices based on multi-level hashing and the SHA family, called the Multi-Level Hashing Algorithm (MLHA). The MLHA consists of eight levels, starting from a lightweight level to a heavy level. Each level is based on a different modified SHA function, with the heavier levels using more complex SHA functions. The output of each level is then produced and represents the final hash value. The use of multiple levels provides several advantages. First, it makes the algorithm more resistant to brute-force attacks. Second, it allows the algorithm to be tailored to the specific security requirements of the IoT device. For example, a device with limited resources may use a lower of level, while a device with more resources may use upper levels for increased security. The use of SHA functions also provides several advantages. First, SHA functions are well-studied and well-respected. Second, SHA functions are relatively efficient to implement in hardware. The number of levels can be easily adjusted to meet the specific needs of the IoT device. The MLHA can also be used to create different types of hash functions. For example, a collision-resistant hash function can be created by using a large number of levels and complex SHA functions. The MLHA is a promising new approach to hashing for IoT devices. The algorithm is efficient, secure, and energy-saving. It is also flexible and scalable, making it suitable for a wide range of IoT devices, from devices with limited resources to devices with complex resources such as supercomputers. The MLHA was evaluated on a variety of IoT devices, including Raspberry Pi 4B hardware, and on two different operating systems, Windows and Ubuntu. The results showed that the MLHA is both secure and energy-efficient. The fact that the MLHA was tested on two different operating systems and on a variety of IoT devices, including a Raspberry Pi 4B, demonstrates the algorithm's portability and versatility. This is important because it means that the MLHA can be used in a wide range of IoT applications.

Keywords: *hashing, IoT security, energy management, multi-level hashing, SHA family, Raspberry Pi 4B.*

1. Introduction

The evolution of the Internet of Things (IoT) marks a significant turning point in our technological landscape, a transformation that has seamlessly interwoven our physical and digital realities in an unprecedented manner. The IoT, a dynamic network of interconnected devices, has orchestrated a paradigm shift across a multitude of sectors, fundamentally altering our interactions with and understanding of the world around us. This interconnected web of devices, often endowed with sensors and actuators, communicates effortlessly, creating a symbiotic relationship between the physical and virtual realms [1].

The pervasive influence of IoT extends across diverse domains, leaving an indelible mark on our daily lives. In smart homes, IoT orchestrates an orchestra of devices, optimizing energy usage, enhancing security, and personalizing the living experience. Industrial automation has witnessed a transformative wave with IoT, streamlining operations, improving efficiency, and fostering predictive maintenance [2]. In the realm of healthcare, IoT-enabled devices monitor patients in real-time, enabling remote healthcare, early detection of anomalies, and data-driven decision-making. Smart cities, an epitome of urban development, are leveraging IoT to enhance infrastructure, traffic management, waste management, and public services [3].

IoT is no longer a distant futuristic vision but an integral facet of contemporary living. Its potential to elevate convenience, efficiency, and productivity is vast and ever-expanding. The promise of a more connected, data-driven, and responsive world drives the continuous growth and

[1] *Computer Science Department, Faculty of Science, University of Zakho, Duhok, Iraq; teba.sami@uoz.edu.krd*
[2] *Energy Eng. Dept., Technical College of Engineering, Duhok Polytechnic University, Duhok, Iraq; subhi.rafeeq@dpu.edu.krd*
[3] *Network Department, Sulaimani Polytechnic University, Sulaimani, Iraq; sarkar.ahmed@spu.edu.iq*

innovation within the IoT sphere. As we traverse this digital evolution, it becomes increasingly imperative to address the inherent challenges posed by this expansive interconnected network [4].

In an era where the Internet of Things (IoT) has transcended novelty and become an integral part of modern life, the security of these interconnected devices has come under heightened scrutiny. The widespread adoption of IoT devices has made them an attractive target for malicious actors seeking to exploit vulnerabilities. The potential consequences of such security breaches are vast and could range from compromised data integrity and breached privacy to disrupted operations. Thus, ensuring robust security measures to protect the data transmitted and stored within IoT networks is paramount. The evolving threat landscape necessitates a proactive and adaptive approach to fortify the security and privacy of IoT devices, allowing the IoT ecosystem to fulfill its transformative potential securely [5].

Concurrently, the rise of battery-powered and energy-constrained IoT devices has underscored the pressing need for efficient energy management strategies. Efficient energy usage is crucial not only to extend the operational lifespan of these devices but also to reduce operational costs and minimize the environmental impact of IoT proliferation. Striking a delicate balance between enhancing security and optimizing energy consumption in IoT devices is a complex challenge that calls for innovative and comprehensive solutions. By developing strategies that address both security and energy efficiency, we can advance the sustainability and effectiveness of IoT ecosystems [6] [7].

At the core of this intricate challenge lies the hashing algorithm, a fundamental cryptographic tool that transforms input data into a fixed-size string of characters, known as a hash value. Hashing algorithms play a critical role in various security applications, providing the foundation for digital signatures, data integrity verification, authentication, and more. When applied to IoT, the potential for designing a new and specialized hashing algorithm is vast and promising. Such an algorithm can significantly contribute to addressing the security and energy efficiency concerns inherent in IoT environments [8].

This paper embarks on a voyage of innovation. The goal is to conceive, develop, and implement a cutting-edge hashing algorithm meticulously crafted to suit the unique demands of IoT ecosystems. The overarching objective is to bolster the security posture of IoT devices, ensuring the sanctity of data through measures that encompass data integrity, confidentiality, and authenticity. Moreover, the innovative hashing algorithm is designed to optimize energy consumption, paving the way for sustainable and efficient IoT operations. Through this concerted effort, we strive to contribute groundbreaking advancements to the rapidly evolving field of IoT technology, propelling the industry towards a future that is secure, energy-conscious, and resilient. Ultimately, this innovation aims to unlock the untapped potential of IoT technology, making it safer, more efficient, and ultimately more impactful in our swiftly evolving digital era.

2. Background Theory

Because of the considerable push given by the Fourth Industrial Revolution toward the integration of real-world industrial settings with virtual computer systems, the path has been made for the successful deployment of the Internet of Things (IoT). The Internet of Things (IoT) is a notion that originated as a direct result of technological breakthroughs that were made practicable as a direct consequence of the industrial revolution. These technological developments made it possible for devices to be connected to the internet. Because of improvements in technology, it is now feasible for individuals to link the products they use on a daily basis to the internet. In this scenario, it is of the utmost importance to set up a large network of linked systems, and each of these systems has to have its own distinct identity in order to maintain effective communication and interaction with the other connected systems. This is a substantial change from the traditional organizational structure that is used in modern industrial settings for the production of commodities. This new structure will be far more effective. These networked systems have the capability to transfer data over the network on their own, without depending on interactions between humans and computers or between humans and other individuals on the network. The capacity to do so is referred to as "autonomous data transfer." [9].

The most recent advancement in internet technology, known as the Internet of Things (IoT), offers the ability to connect a diverse array of devices, each with varying levels of available resources. These devices encompass a spectrum ranging from those with limited resources to those with ample resources. Research findings emphasize that IoT has the transformative potential to digitize physical objects, converting them into their digital counterparts [10][11].

Due to the expansive proliferation and accelerated development of these applications, there has been a noticeable surge in cyberattacks specifically targeting Internet of Things (IoT) applications. Consequently, there has been a rise in the overall number of cyberattacks tailored for IoT. Both the manufacturers of these devices and the end-users face increasing challenges in ensuring the security of IoT devices [12]. This challenge stems from the widespread use of similar sensor nodes in IoT, directly linked to the scarcity of secure devices within the IoT. Within the context of IoT, heterogeneity denotes a diverse array of interfaces, frameworks, laws, and a variety of physical resources, such as differing levels of processor compute power and storage capacity. It is widely acknowledged that the susceptibility of additional apps to various vulnerabilities poses a significant threat to data security [13]. The primary cause for concern is the absence of conventional safety measures. Envisioning a future where IoT devices seamlessly blend into their environment, generating vast amounts of data, is plausible. Processing and securely storing this data are crucial to maintain its meaningfulness and value. Consequently, anyone involved in IoT application development must thoroughly research the unique security challenges posed by IoT, distinct from those encountered in traditional networks [14][15].

The efficient management of energy within IoT networks has been identified as a critical element to enhance power efficiency and extend the operational duration of IoT devices. This observation was derived from the outcomes mentioned earlier. In the realm of applications utilizing the Internet of Things (IoT), there has been a noticeable surge in energy consumption. Effective energy management is pivotal in this context, given its potential for substantial energy savings and a more streamlined response to energy crises [16]. This concern about energy management stems primarily from this reason. Although the user has provided a numerical reference, it lacks associated supporting material or context. The recent years have seen a significant acceleration in urbanization rates, necessitating the implementation of sustainable, efficient, and intelligent measures to address a range of challenges encompassing transportation, governance, environmental issues, and overall quality of life [17]. Consequently, the adoption of sustainable, efficient, and intelligent measures has become imperative to tackle these challenges [18]. The Internet of Things (IoT) plays a pivotal role in supporting a broad spectrum of cutting-edge and general applications crucial for the advancement of smart cities. The simultaneous growth in both the number and specifications of IoT devices has led to an upward trajectory in energy consumption within IoT applications in recent years. Consequently, it is vital for solutions within smart cities to effectively harness energy and adeptly handle the associated challenges, given the increasing significance of smart cities [19]. It is a widely accepted notion that energy management should serve as the cornerstone for the development of intricate energy systems within smart cities.

Addressing the energy challenges posed by Internet of Things (IoT) networks necessitates the adoption of diverse and proven strategies. These strategies encompass the development of algorithms, protocols, and hardware solutions geared towards reducing network energy consumption and extending their operational longevity [20]. The primary objective of energy-saving initiatives is to minimize energy usage within networks. The limited energy capacity of IoT devices, intended to function continuously for extended periods without frequent battery replacements, presents a significant hurdle for the IoT, constituting a notable drawback. Moreover, the proliferation of devices connected to IoT networks has surged due to the growing prominence of the IoT, consequently resulting in a substantial increase in the carbon footprint of these networks in recent times. Academics and industry professionals are increasingly drawn to the intriguing and pertinent field of study related to Green-IoT and IoT energy management [21]. Green-IoT essentially entails running the Internet of Things (IoT) using alternative or renewable energy sources. Despite the potential of these systems, their widespread adoption has been somewhat impeded due to the current unavailability of easily deployable, easily maintainable, and cost-effective technologies. The immense volume of data collected from various urban areas across a nation poses significant challenges in data storage, management, and analysis. Utilizing Internet of Things (IoT) technology and harnessing big data can prove to be an effective approach in overcoming these challenges [22]. The composite technologies constituting the IoT hold the potential to establish a pervasive computing platform, enabling the monitoring, detection, and control of energy consumption in household appliances on a large scale. This potentiality opens avenues for reducing overall energy consumption. Data gathering is facilitated through a diverse array of wireless sensors placed within residential units [23].

Over the course of the last several years, it has been abundantly evident that a number of hashing algorithms that were originally assumed to be dependable have severe faults that render them unsuitable for assuring the continuous safety of cryptographic systems. These issues include the inability to properly handle collisions, which is a common kind of attack against cryptographic systems. As a result of continuous advancements in computer power, a variety of cryptographic attacks that were formerly thought to exist only in the world of theory are now within the grasp of those who would intentionally inflict damage. These attacks were previously supposed to be impossible. Recent significant advancements in the field of quantum computing predict that in the not-too-distant future, there will be an enhanced capacity to launch assaults against hashing algorithms that are more successful [24]. This is based on the idea that quantum computers would be able to solve certain problems that were previously unsolvable. As a consequence of this, the creation of novel hashing algorithms that provide enough cryptographic resilience in the face of contemporary assaults made feasible by supercomputers and quantum computers is an imperative need. Utilizing effective cryptographic hash algorithms makes it feasible to protect the data even in the face of ongoing increases in computing power that may be made available to potential adversaries

[25]. This is because these adversaries are unlikely to be able to get their hands on such capacity.

A mathematical operation is said to be a hash function if it generates an output with a size that is both stated and fixed, in addition to accepting data of any quantity as its input. An in-depth analysis of the standard hash function is the first step towards discovering cryptographic hash functions. Examples of hash algorithms that are used often in the modern world include the Message-Digest Algorithm 5 (MD5) and several iterations of the Secure Hash Algorithm (SHA). The integrity of the data must be preserved at all costs in order for the system to be considered highly secure. When users interact with a cryptographic hash function system, they are presented with the option to produce a hash value for the message that they are sending [26]. The fundamental objective of the system is to make it much easier to discover any unlawful modifications that have been done to files; this is one of the benefits that the system provides. It is of the highest need to take safeguards in order to maintain the security of vital computer systems and sensitive data. It is possible to verify the origin of data by combining hash functions with other forms of cryptography that are already well-established. Several distinct approaches are possible for achieving this goal. When hashing methods are employed in combination with encryption, the result is the creation of one-of-a-kind message hash values that may be used to determine the origin of the data [27][28].

The National Institute of Standards and Technology (NIST) introduced the Secure Hash Algorithm (SHA) in 1993 and subsequently made enhancements in 1995 to address identified issues from its initial implementation. Recognizing the limitations of smaller digital messages, a new hashing algorithm, SHA-2, was proposed in 2001. SHA-2 operates using larger digital messages, increasing their resistance to potential attacks and allowing for processing larger data chunks [29]. SHA functions, including SHA-224, SHA-256, SHA-384, and SHA-512, share core features but differ in their sizes, encompassing operators, initiators, and the final message expression [30].

Various applications employ hashing as a security enhancement method, validating the authenticity of data submitted by users or confirming received information integrity. Hash value comparisons are essential for ensuring data integrity. Established standards for generating hash values are MD and SHA, which comprises SHA-1 and SHA-2. Due to the ever-evolving technological landscape, continuous advancements or improvements in existing methods are imperative to meet contemporary requirements. Hash algorithms, a specific cryptographic procedure, compress the content of an input message into a numerical hash value, allowing for validation of the original message's authenticity. A key distinguishing characteristic among hash algorithms lies in their complexity in finding distinct messages generating the same hash result [31]. Figure 1 illustrates the typical operation of a hash algorithm

Fig 1: A Typical Process of Hash Algorithm.

3. Methodology of Proposed Algorithm

Next this, we will discuss the architecture of the system that has been shown, which entails converting the traditional SHA family algorithms into the brand new Multi-Level Hashing Algorithm (MLHA). This will be covered in the next section. The software engineering life cycle, in addition to a number of other approaches, is used here in order to facilitate the successful accomplishment of this objective. Creating a list of needs, which should contain both functional and non-functional requirements, is the first step in the process. This list should be comprehensive. This phase marks the beginning of the procedure itself. In addition to this, the process of developing a system include addressing the difficulties and finding solutions to the issues that were identified in the first phase of the process. This is done in order to complete the system. In conclusion, the purpose of this study was to provide a complete illustration of the many techniques that were involved in the creation of an innovative machine learning-based heuristic algorithm (MLHA). On the other hand, the approach that has been proposed turns the concept into action by writing code for it, which finally results in the development of an application that can be run.

1.1 General Idea of the Proposed Algorithm

The fundamental purpose of this investigation is to create an original hashing algorithm that is tailored to meet the requirements of Internet of Things (IoT) devices. As a result of the Internet of Things (IoT) including both low-power small devices and high-capacity larger devices, the goal is to build an algorithm that demonstrates optimum performance over the whole of this range of sizes and capacities of devices. By having an understanding of the many different classes of Internet of Things (IoT) devices, by doing an in-depth analysis of the algorithms that are already in use, and by developing an innovative technique that successfully balances efficiency and security. This research aims to improve the data protection and security mechanisms of Internet of Things (IoT) devices by using more sophisticated hashing algorithms. This will be accomplished as a result of this study. The algorithm that is now under discussion offers an innovative approach to the problem of improving data security in Internet of Things (IoT) devices. The methods under examination are derived from the Secure Hash Algorithm (SHA) family of safe hashing algorithms. These algorithms are mostly based on bitwise operations. On the other hand, these algorithms have been altered in order to improve their effectiveness and scalability in relation to devices that are part of the Internet of Things (IoT).

The algorithm that is being considered consists of eight separate layers, each of which is designed to support a certain category of Internet of Things (IoT) devices. The first tier is defined by its simplicity and is especially made for devices that possess restricted resources, such as little RAM, CPU capacity, and battery endurance. The second tier is characterized by its complexity and is specifically tailored for devices that possess unlimited resources. This level makes use of simple mathematical equations and functions and has a restricted vocabulary. Its output is succinct, and it demonstrates good writing skills. The second level of the system requires a larger vocabulary and results in replies that are longer and more complex as a result of the use of more elaborate equations and functions. This stage is distinguished from the earlier one due to the complex structure of the system, which enables it to meet the needs of Internet of Things (IoT) devices that are becoming more smart and hold a significant amount of resources. The iterative development process is ongoing, and at each level, adjustments are being made so that they can suit the capabilities of different Internet of Things (IoT) devices. The algorithm eventually works its way up to its most advanced levels, which are characterized by more difficult equations, functions, and output lengths, in addition to a significantly increased number of words, arithmetic and logical operations, and iterations. These levels may be identified from one another by their complicated nature. In comparison to the IoT devices of the fundamental level, those of the advanced levels are designed to fulfill the needs of IoT devices that are larger in scale and more complex in their makeup. This is the primary objective of these higher levels.

The process that is presently being considered for deployment is one that is known as Multi-Level Hashing, or MLHA for short. This is the approach that is used to hash data across many levels. As can be seen from the data shown in Table 1, the system gives the people who utilize it a total of eight different challenges from which to choose and select. Due to the low level of computing that is required of them, the MLHA1, MLHA2, and MLHA3 algorithms are all excellent candidates for implementation on devices that are on the more compact scale. Both the MLHA4 and the MLHA5 are great options for use with electrical equipment that has power sources of a moderate size and requires a small to moderate quantity of power. These requirements describe a wide range of common household appliances. However, the MLHA6, MLHA7, and MLHA8 algorithms were developed with the intention of being utilized in electronic devices that have central processing units (CPUs) that are capable of high performance and that have a reliable source of electrical power. This was the design purpose behind the development of these algorithms. This was carried out with the anticipation that the aforementioned algorithms would be included into various electronic gadgets. Certain applications, such as the Blockchain Technology, are reliant on the use of a hashing algorithm that requires a significant amount of processing time in order to function in an efficient manner. This is necessary for the application to function properly. The current environment presents miners with a greater amount of obstacles to overcome on their way to effectively cracking hashes. Miners must overcome these obstacles in order to be successful.

Table 1: Comparing Levels of proposed MLHA Algorithm.

MLHA Level	Processing Power Required	Algo. Robustness
Level One	Minimal processing power needed	Vulnerable
Level Two	Basic to moderate processing power required	Less Secure
Level Three	Moderate processing power sufficient	Partially Secure
Level Four	Moderate to high processing power necessary	Moderate Secure
Level Five	Significant processing power needed	Robust
Level Six	Considerable processing power required	Secure
Level Seven	Ample processing power is a must	More Secure
Level Eight	Extensive processing power is essential	Highly Secure

The sequential steps that need to be finished in order for the contribution to be planned and carried out are shown in Figure 2. A block diagram serves to illustrate these different steps in the process. These stages have been written down in the sequence that must be followed in order to accomplish the operation effectively. This study is linked to the process of building a hashing algorithm that can adapt to the unique demands of Internet of Things (IoT) devices in order to satisfy their requirements. This technique is necessary in order to meet the requirements of these devices. The Multi-Level Hashing Algorithm (MLHA) is the name of the method that was developed with the intention of responding to the unique needs and limitations of IoT devices.

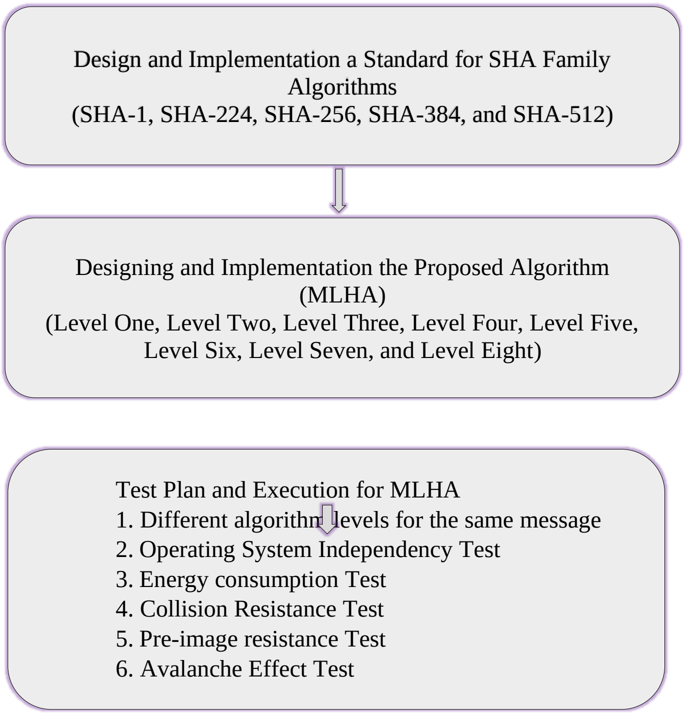
Design and Implementation a Standard for SHA Family Algorithms
(SHA-1, SHA-224, SHA-256, SHA-384, and SHA-512)
Designing and Implementation the Proposed Algorithm (MLHA)
(Level One, Level Two, Level Three, Level Four, Level Five, Level Six, Level Seven, and Level Eight)
Test Plan and Execution for MLHA
1. Different algorithm levels for the same message
2. Operating System Independency Test
3. Energy consumption Test
4. Collision Resistance Test
5. Pre-image resistance Test
6. Avalanche Effect Test

Fig 2: General diagram of the proposed Systems.

Figure 3 shows the diagram of Hash value generation.

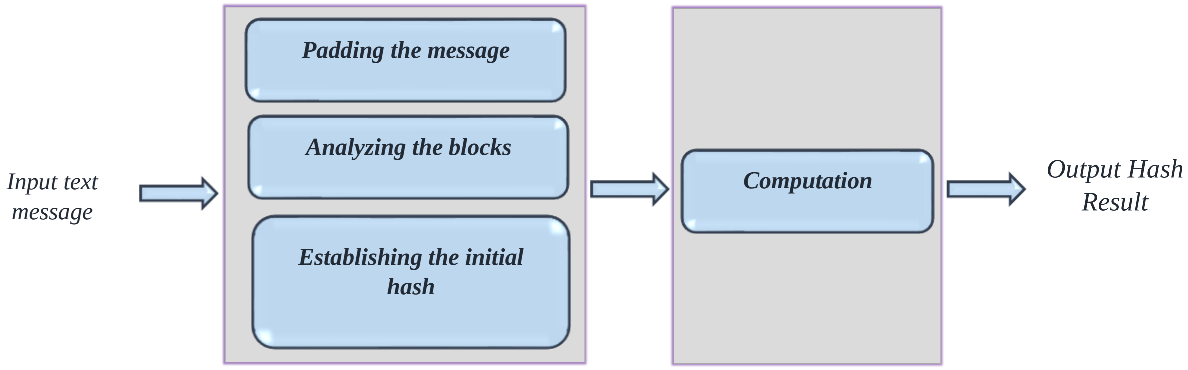

Fig 3: Diagram of Hash Value generation.

1.2 Requirements of the Proposed Algorithm

This section will offer a breakdown of all the needed criteria that need to be completed before the development of the proposed system can begin. These requirements must be met before the development of the proposed system can even begin. In this specific environment, a thorough examination of a broad variety of issues will be carried out, but there will be no effort made to take into consideration any potential solutions to these issues. This means that the issues will not be addressed in any meaningful way. There are two primary categories of requirements: software requirements and hardware requirements. SW requirements and HW requirements are both acceptable abbreviations for these two types of requirements.

1.2.1 Software Requirements

For designing and implementing the MLHA Algorithms for the eight levels the following SW requirements are needed for developing all existing and proposed MLHA Algorithm:

- A windows application should be developed using the C++ language to implement the design. Visual Studio 2022 will be used for the development.
- Net. Framework 4.8 will be used that includes all the required C++ Libraries.
- Each algorithm has its class. Then, unlimited objects can be created based on each of these classes.

- A database should be created to perform the tests and holds the results. MySql Database will be used for this purpose.
- XAMPP Version 3.3.4 will be used to create a webserver and manage the database using phpMyAdmin version 5.0.4.
- Windows 11 Professional (64Bit) is used as a platform for development.
- EdrawMax Version 10.5.2 is used to draw the diagrams.
- GitHub is web-based code version control used to back up the SW projects online to avoid losing the project in any unpredicted situation.

1.2.2 Hardware Requirements

Two different Personal Computers (PCs) with different specifications are used in this paper. And a Raspberry Pi with the properties: Raspberry Pi 4 Model B, and Memory of 64 GB.

1.3 System Design

During the requirements phase of the system, any issues that were discovered have to be documented and addressed before moving on to the design phase of the system. In order to finish the task, you will need to create a variety of distinct system parts, such as modules, architecture, components, interfaces, and data, based on a set of predetermined standards.

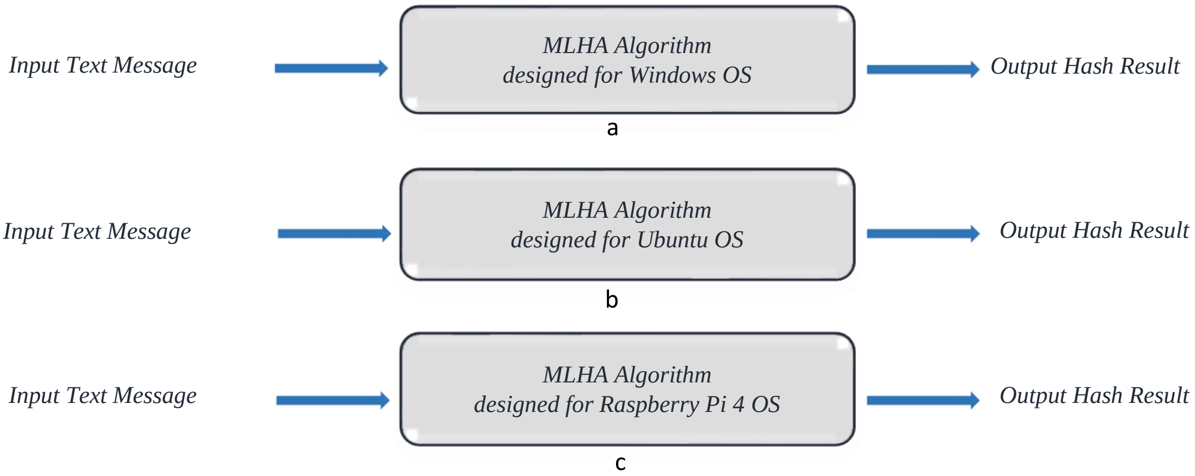
Input Text Message
MLHA Algorithm designed for Windows OS
Output Hash Result
a
Input Text Message
MLHA Algorithm designed for Ubuntu OS
Output Hash Result
b
Input Text Message
MLHA Algorithm designed for Raspberry Pi 4 OS
Output Hash Result
c

Fig 4: Organization diagram of the three depended environments:
(a) Based on Windows OS, (b) Based on Ubuntu OS, (c) Based on Raspberry Pi 4B Microcontroller.

It will be essential to find, create, and construct a number of different systems with the intention of fulfilling a specified set of criteria in order to be able to carry out this mission with any degree of success. This will include a series of different stages. It is highly encouraged that each of these responses be created in a consistent fashion making use of UML Diagrams, and the whole collection of these answers may be read in its entirety on this page. Figure 4 illustrates how the Multi-Level Hashing Algorithm, which is more often referred to as MLHA, has been developed over the course of many years and used in a wide range of settings in a number of different ways. A little examination of the figure will make this point very evident.

In order to ensure that the MLHA is durable and interoperable across a number of different platforms, it is necessary to test it on a number of different operating systems, such as Windows, Linux, and Raspberry Pi. Performing the tests on a wide variety of platforms is one way to achieve this goal. After it has been developed, the MLHA algorithm may be used in one of three separate settings: a laboratory, an industrial environment, or a military setting. Each of these settings has its own unique advantages and disadvantages:

- The Windows-based personal computer (PC1), which functioned as the major component of the first setting, was the primary component of the initial setup. In addition to that, this particular personal computer was the only one of its kind. when did the process of putting together the ecosystem officially begin, and when exactly did it start? A graphical depiction of the organizational structure that is relevant to the conditions at hand is provided in the form of Figure 4-a, which may be found at this site.
- In the second scenario, we used a computer with the identify PC2 that was operating a variant of Linux known as Ubuntu 22.04.2. image 4-b provides a graphical depiction of the information that it supplies in addition to a picture of the organizational chart that explains the make-up of this specific environment (see also image 4-a). Figure 4-a contains this chart for your perusal.
- A true Internet of Things (IoT) device or, more precisely, a hardware design that integrates the Raspberry Pi 4B Microcontroller is required for the third configuration option. These two choices are required in equal measure. Before moving on to the next step, it is

important to first satisfy both of these preconditions in order to qualify for advancement. It is probable that the next section (3.3.1), which will be located after this one, may devote a greater degree of focus to digging into the specifics of the intricate workings of the microcontroller. This is because the next section will be located after this one. The hierarchical organization chart that is shown here and can also be viewed in Figure 4-c gives an example of the structure that is present within this specific environment. It is conceivable that it will be located in this particular location.

1.3.1 Raspberry Pi 4 Microcontroller.

Throughout the course of human history, several iterations of the technology that would one day develop into the modern computer have appeared at various points in time. Iteration after iteration has resulted in a product that is both easier to work with and less expensive to produce than the one that came before it. As a direct consequence of this, the development of the modern computer was at last made possible after a lengthy period of time. Smartphones are a popular kind of pocket-sized electronic gadget, and a sizeable portion of the population in today's society constantly carries one with them. These smartphones, as well as a number of other types of electronic gadgets, such as tablets and smartwatches, combined with a few other electronic devices, along with a few other electronic devices, may be clubbed together along with a few other electronic devices to create the category of minicomputers. In countries where the economy is expanding, the number of people who have access to computers is still relatively low, despite the fact that computers are used almost everywhere. The fact that there is a gap in the distribution of computers and programming technology served as the impetus for the creation of the Raspberry Pi device, as well as the subsequent production of the product. This gap served as the inspiration for the Raspberry Pi Foundation, which was founded in 2012. In addition, the fact that there is a gap in the distribution of computers and programming technology served as the motivation for the development of the product. This gap served as the drive for the creation of the product. A further impetus for the production of the product was the realization that there is a gap in the distribution of computer technology and programming know-how, which served as a source of inspiration [32][33].

The Raspberry Pi is a single-board computer that combines an affordable price with a minimal amount of space required for installation. Additionally, it is well-known for being easily accessible. It is fairly little, around the size of a credit card, and it can be packed extremely tightly. Its measurements are presented in a format that is essentially comparable to that of a standard credit card in every respect, including the method in which its dimensions are set out. It acts as a platform that allows people from a varied range of backgrounds and various degrees of expertise to participate in computational activities. As a consequence of this, it generates possibilities for individuals to learn knowledge via the accumulation of practical experience. The Raspberry Pi foundation, which is founded in the United Kingdom and was responsible for the creation of a cutting-edge motherboard, also has its headquarters located in that nation. At this point in time, there is a universal opinion that the introduction of this mother board was directly responsible for a considerable advancement in the continuing process of the technical progression of computers. This is due to the fact that the introduction of this mother board was directly responsible for the introduction of this mother board. The minicomputer has the capacity of creating connections with a wide range of supplementary pieces of hardware, such as, among other things, a keyboard, a mouse, and a monitor, among other things. These connections may be made by the user. The extension port on the minicomputer gives the mechanism for creating these connections with other devices [34].

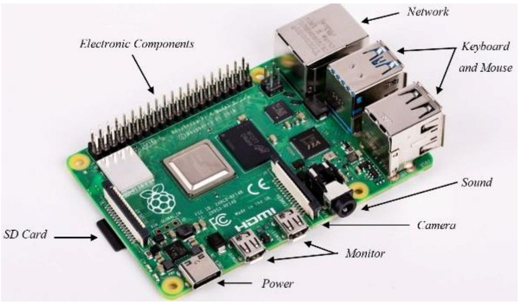

Fig 5: Raspberry Pi 4 model B.

The Raspberry Pi offers a broad variety of possible applications, some of which include the study of programming languages and the ease of managing networks. Among these potential uses is the convenience of managing networks. The object in question may be used for a number of different things, and it has received a far higher level of attention in recent years than was first anticipated [35]. The Raspberry Pi 4 model B schematic is shown in figure 5, which gives a visual depiction of the diagram.

As can be seen in Figure 6, the recommended hashing method makes use of a microcontroller based on the Raspberry Pi 4 model B. Memory capacities may now vary anywhere from 2 gigabytes of random access memory (RAM) to 8 gigabytes of random access memory (RAM) on the most recent model, which represents a major improvement over earlier versions of the product. In addition, the unit has a high-performance central processing unit (CPU) operating at 1.5 GHz and a well-balanced collection of USB ports, which includes both USB 2.0 and USB 3.0 versions. The Raspberry Pi 4B is regarded as the best model available because of its adaptability, which enables it to function well in a broad variety of contexts. It provides a significant boost in RAM capacity, which, depending on the memory resources that are available, may successfully cater to the demands of even the most dedicated programmers.

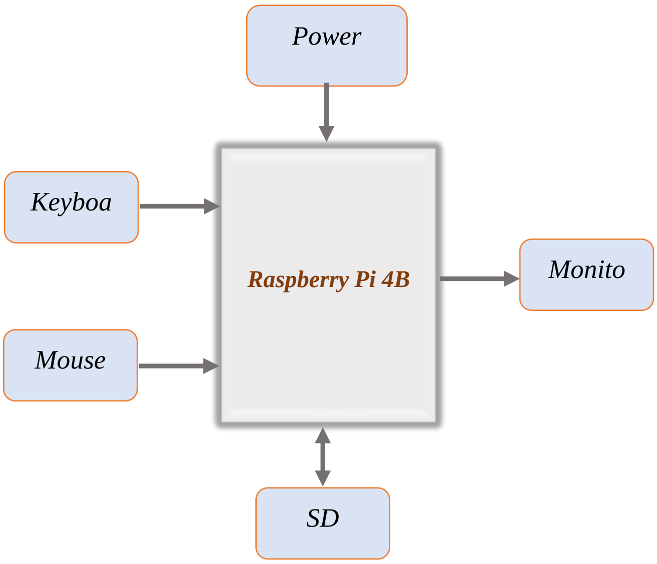

Fig 6: Block diagram of used Raspberry Pi 4 model B.

1.3.2 ER Diagram

Currently, in order to show one example of the structural components that comprise the dependent database, an entity relationship diagram, which is also often referred to as an ER diagram, is being used. The database management system is referred to as "phd_database," and the name of one of the tables that is included inside it is "bruteforce." The brute-force database has a number of columns, and the only one of those columns that has not been indexed is the message column. The information in all of the other columns has been looked over. The indexing procedure for all of the remaining columns has been finished and is now complete. Using this optimization strategy will result in a faster execution time for select queries, but it will have a significant impact on the performance of insert queries. This is because insert queries involve writing data into a table. When you use this optimization approach, the amount of time it takes to do an insert query will be cut down significantly. This is because of the rationale that was discussed earlier in this paragraph in the previous sentence. Entity connection diagrams for each table are going to be shown for your viewing enjoyment in the next subsections of this section. There is some type of connection that can be made between these diagrams, which correspond to tables, and the tables that they represent.

The bruteforce table serves as a repository for the results of testing for brute-force attacks that were carried out on each of the MLHA Algorithms. These tests were carried out on each algorithm individually. These tests were carried out on an individual basis for every algorithm. Figure 7 presents the ER diagram that shows the brute-force table for your viewing and reading convenience. You are welcome to look it over at any time. The ID Field is the one that is tasked with the responsibility of giving the table with both an auto-incrementing key as well as the main key. Before the hash function performs any actions on the plain text, the unprocessed version of the text is saved in the Message field, which has the datatype Text and is used for storing the text. This occurs before the hash function records any of the plain text. The text itself is stored using the datatype known as Text. Text datatypes are used in order to preserve the results that are generated by the MLHA functions, the names of which are as follows: MLHA1, MLHA2, MLHA3, MLHA4, MLHA5, MLHA6, MLHA7, and MLHA8. These results are saved in order to be accessed at a later time. After that, you'll have the option to retrieve these results at a later time. These outcomes may be attained by the use of the following syntax: MLHA1, MLHA2, MLHA3, MLHA4, MLHA5, MLHA6, MLHA7, and MLHA8. The results of the MLHA functions that were run on the message are preserved in these fields so that they may be accessed at a later point in time. This is done so that the results can be retrieved easily. The results of the functions are temporarily stored in these fields while they are being processed. It is necessary to do an analysis on the information that is shown in this table in order to identify instances of collisions that take place inside the various levels of the algorithm that is being studied at the moment. Examining the information that is shown in this table will lead you to these specific examples.

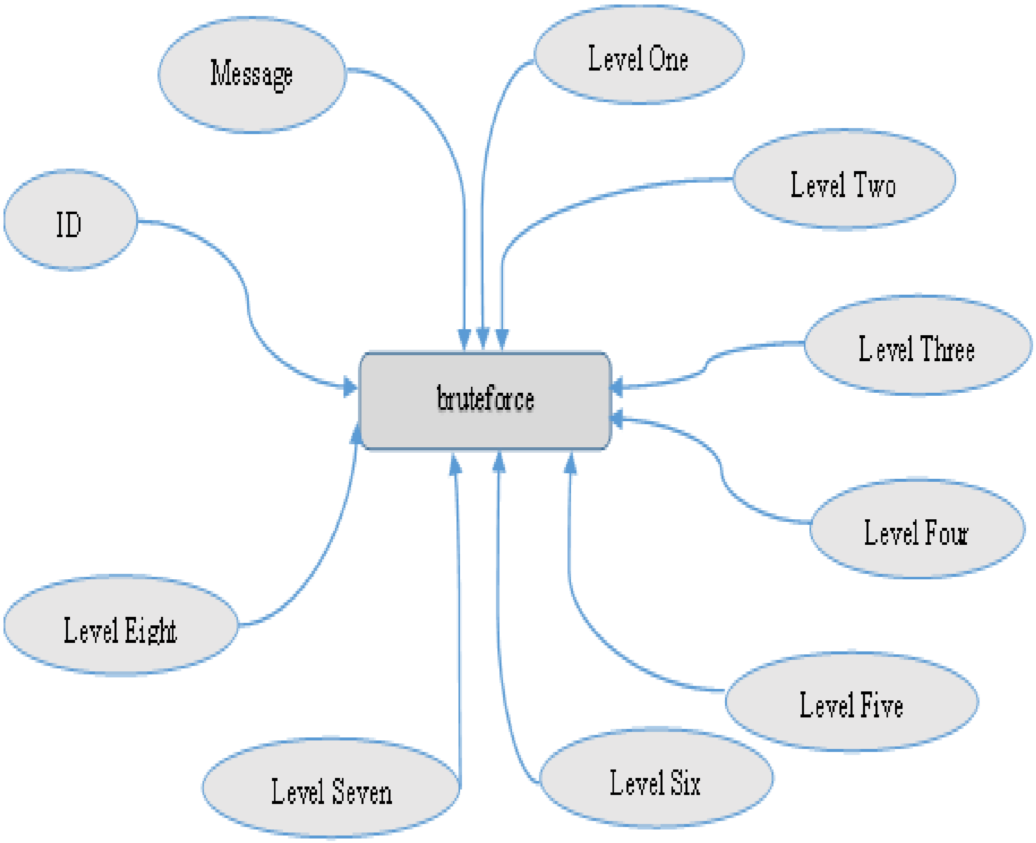

Fig 7: ER Diagram for database of Bruteforce Table in the database.

Because it involves the hashing of a wordlist file with an expected size of around 15 gigabytes, which contains frequently used passwords from various locations all over the globe, the procedure at hand is predicted to take a large amount of time, spanning several months. This is due to the fact that it includes commonly used passwords from all over the world. This is as a result of the fact that the file includes passwords that are used in several countries and regions around the globe. In addition to that, there is a possibility that this table may be used for purposes linked with penetration testing at some point in the future.

1.3.3 Pseudo Code Samples of MLHA

In order to illustrate the efficiency of the proposed hashing algorithm, we have selected first and last levels of the new MLHA to be produced in the form of Pseudo Code. As illustrated bellow:

a. Pseudo Code Implementation of MLHA1 (Level One)

- Initialization of constants (K) and initial hash values (H).
- Message preparation by converting to ASCII and binary, adding padding, and incorporating message length.
- Processing the message in successive 512-bit blocks.
- Preparing a message schedule and initializing working variables.
- Executing a compression loop within each block.
- Updating hash values.
- Returning the final hash result in hexadecimal format.

This process is depicted in the flowchart diagram in Figure 8.

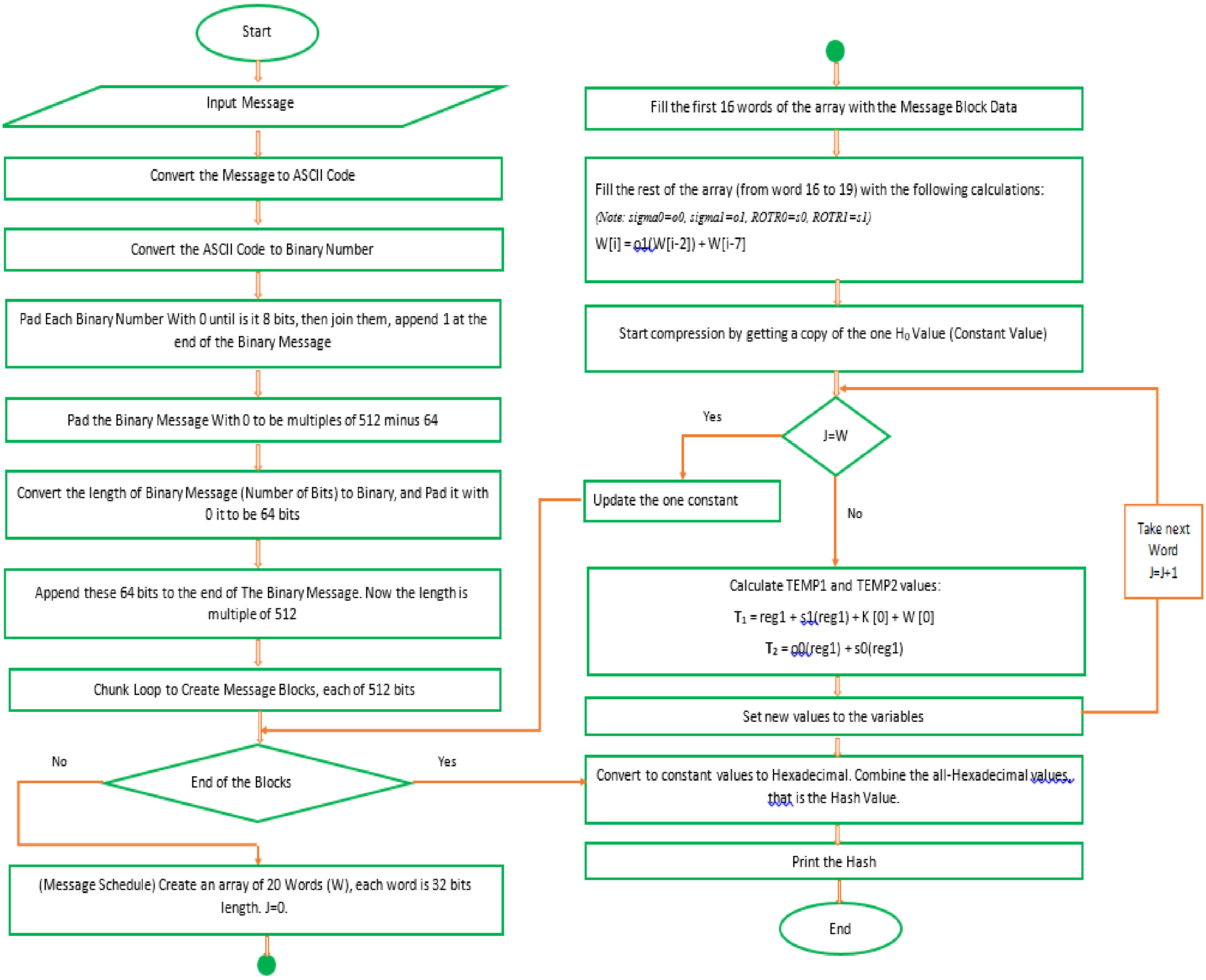
Start
Input Message
Convert the Message to ASCII Code
Convert the ASCII Code to Binary Number
Pad Each Binary Number With 0 until is it 8 bits, then join them, append 1 at the end of the Binary Message
Pad the Binary Message With 0 to be multiples of 512 minus 64
Convert the length of Binary Message (Number of Bits) to Binary, and Pad it with 0 it to be 64 bits
Append these 64 bits to the end of The Binary Message. Now the length is multiple of 512
Chunk Loop to Create Message Blocks, each of 512 bits
No
End of the Blocks
Yes
(Message Schedule) Create an array of 20 Words (W), each word is 32 bits length. J=0.
Fill the first 16 words of the array with the Message Block Data
Fill the rest of the array (from word 16 to 19) with the following calculations:
(Note: sigma0=o0, sigma1=o1, ROTR0=s0, ROTR1=s1)
W[i] = o1(W[i-2]) + W[i-7]
Start compression by getting a copy of the one H0 Value (Constant Value)
Yes
J=W
Update the one constant
No
Take next Word J=J+1
Calculate TEMP1 and TEMP2 values:
T1 = reg1 + s1(reg1) + K [0] + W [0]
T2 = o0(reg1) + s0(reg1)
Set new values to the variables
Convert to constant values to Hexadecimal. Combine the all-Hexadecimal values, that is the Hash Value.
Print the Hash
End

Fig 8: Flowchart of MLHA1.

b. Pseudo Code Implementation of MLHA8 (Level Eight)

- Initialization of constants (K) and initial hash values (H).
- Message preparation by converting to ASCII and binary, adding padding, and incorporating message length.
- Processing the message in successive 512-bit blocks.
- Preparing a message schedule and initializing working variables.
- Executing a compression loop within each block.
- Updating hash values.
- Returning the final hash result in hexadecimal format

This process is depicted in the flowchart diagram in Figure 9.

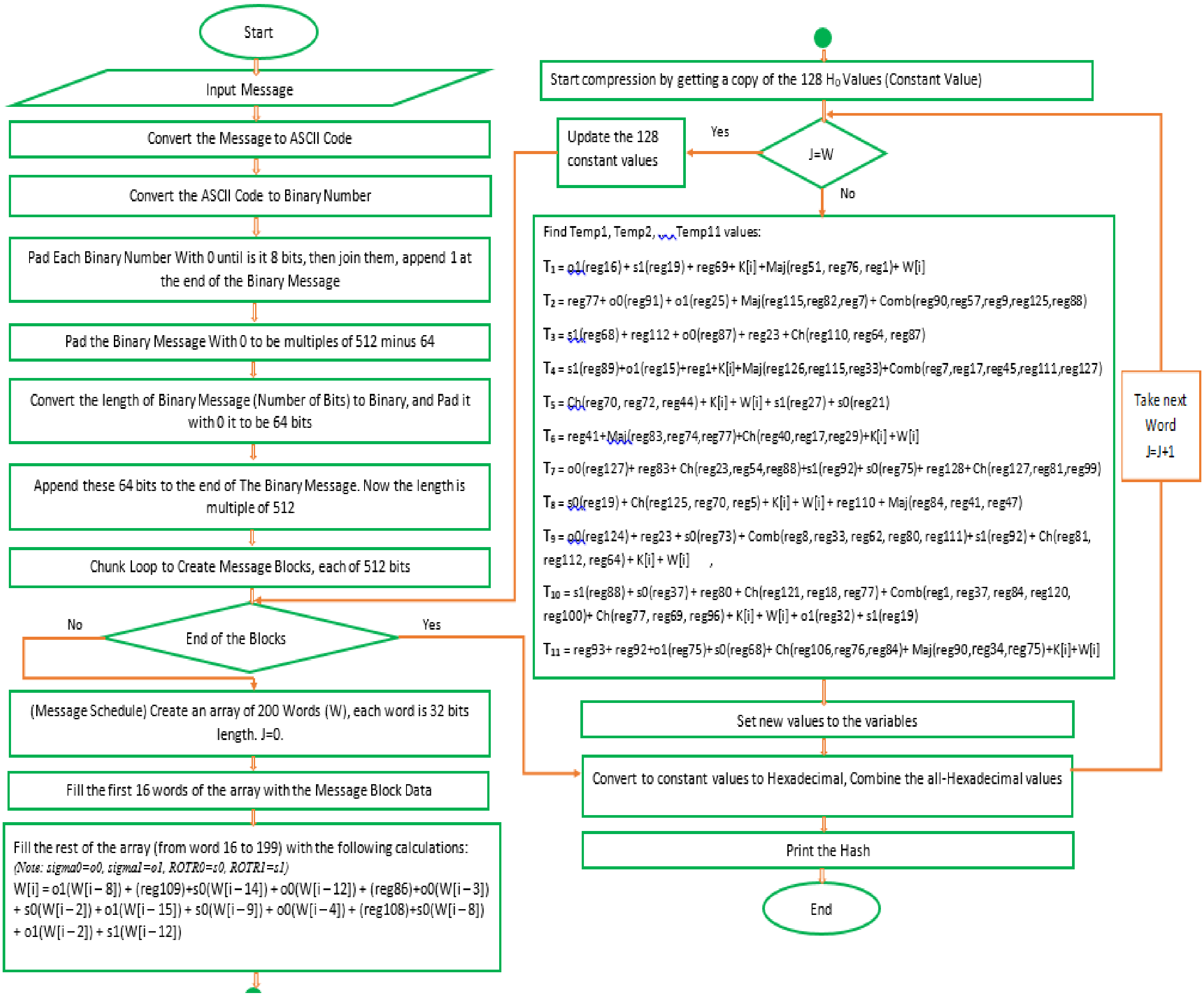
Start
Input Message
Convert the Message to ASCII Code
Convert the ASCII Code to Binary Number
Pad Each Binary Number With 0 until is it 8 bits, then join them, append 1 at the end of the Binary Message
Pad the Binary Message With 0 to be multiples of 512 minus 64
Convert the length of Binary Message (Number of Bits) to Binary, and Pad it with 0 it to be 64 bits
Append these 64 bits to the end of The Binary Message. Now the length is multiple of 512
Chunk Loop to Create Message Blocks, each of 512 bits
No
End of the Blocks
Yes
(Message Schedule) Create an array of 200 Words (W), each word is 32 bits length. J=0.
Fill the first 16 words of the array with the Message Block Data
Fill the rest of the array (from word 16 to 199) with the following calculations:
(Note: sigma0=o0, sigma1=o1, ROTR0=s0, ROTR1=s1)
W[i] = o1(W[i – 8]) + (reg109)+s0(W[i – 14]) + o0(W[i – 12]) + (reg86)+o0(W[i – 3]) + s0(W[i – 2]) + o1(W[i – 15]) + s0(W[i – 9]) + o0(W[i – 4]) + (reg108)+s0(W[i – 8]) + o1(W[i – 2]) + s1(W[i – 12])
Start compression by getting a copy of the 128 H_0 Values (Constant Value)
Update the 128 constant values
Yes
J=W
No
Find Temp1, Temp2, ...Temp11 values:
T_1 = o1(reg16) + s1(reg19) + reg69+ K[i] +Maj(reg51, reg76, reg1)+ W[i]
T_2 = reg77+ o0(reg91) + o1(reg25) + Maj(reg115,reg82,reg7) + Comb(reg90,reg57,reg9,reg125,reg88)
T_3 = s1(reg68) + reg112 + o0(reg87) + reg23 + Ch(reg110, reg64, reg87)
T_4 = s1(reg89)+o1(reg15)+reg1+K[i]+Maj(reg126,reg115,reg33)+Comb(reg7,reg17,reg45,reg111,reg127)
T_5 = Ch(reg70, reg72, reg44) + K[i] + W[i] + s1(reg27) + s0(reg21)
T_6 = reg41+Maj(reg83,reg74,reg77)+Ch(reg40,reg17,reg29)+K[i] +W[i]
T_7 = o0(reg127)+ reg83+ Ch(reg23,reg54,reg88)+s1(reg92)+ s0(reg75)+ reg128+ Ch(reg127,reg81,reg99)
T_8 = s0(reg19) + Ch(reg125, reg70, reg5) + K[i] + W[i] + reg110 + Maj(reg84, reg41, reg47)
T_9 = o0(reg124) + reg23 + s0(reg73) + Comb(reg8, reg33, reg62, reg80, reg111)+ s1(reg92) + Ch(reg81, reg112, reg64) + K[i] + W[i] ,
T_{10} = s1(reg88) + s0(reg37) + reg80 + Ch(reg121, reg18, reg77) + Comb(reg1, reg37, reg84, reg120, reg100)+ Ch(reg77, reg69, reg96) + K[i] + W[i] + o1(reg32) + s1(reg19)
T_{11} = reg93+ reg92+o1(reg75)+ s0(reg68)+ Ch(reg106,reg76,reg84)+ Maj(reg90,reg34,reg75)+K[i]+W[i]
Take next Word J=J+1
Set new values to the variables
Convert to constant values to Hexadecimal, Combine the all-Hexadecimal values
Print the Hash
End

Fig 9: Flowchart of MLHA8.

4. Conclusion

This work has suggested a novel hashing method for IoT devices based on multi-level hashing and the SHA family, termed the Multi-Level Hashing method (MLHA). The MLHA consists of eight levels, going from a lightweight level to a heavy level. Each level is based on a distinct modified SHA function, with the higher levels employing more sophisticated SHA functions. The MLHA was assessed on a range of IoT devices and was proved to be both secure and energy-efficient. The MLHA is a potential new way to hashing for IoT devices. The algorithm is efficient, secure, and energy-efficient. It is also adaptable and scalable, making it suited for a broad variety of IoT devices, from devices with little resources to devices with complicated resources. The MLHA has the ability to greatly enhance the security and energy efficiency of IoT devices. This is significant because IoT devices are rapidly being utilized in a broad variety of applications, including critical infrastructure applications. As the number of IoT devices continues to expand, it is necessary to create new security and energy management solutions that can handle the particular difficulties of these devices. The MLHA is one step towards designing more secure and energy-efficient IoT devices. The MLHA's performance on the Raspberry Pi 4B is especially notable. The Raspberry Pi 4B is a low-cost, low-power IoT device that is widely utilized in resource-constrained applications.

References

[1] Sadeeq, Mohammed AM, et al. "Internet of Things security: a survey." 2018 International Conference on Advanced Science and Engineering (ICOASE). IEEE, 2018.

[2] Zeebaree, Subhi RM, et al. "Characteristics and analysis of hadoop distributed systems." Technology Reports of Kansai University 62.4 (2020): 1555-1564.

[3] Chataut, Robin, Alex Phoummalayvane, and Robert Akl. "Unleashing the Power of IoT: A Comprehensive Review of IoT Applications and Future Prospects in

Healthcare, Agriculture, Smart Homes, Smart Cities, and Industry 4.0." Sensors 23.16 (2023): 7194.

[4] Sami, Teba Mohammed Ghazi, Subhi RM Zeebaree, and Sarkar Hasan Ahmed. "A Comprehensive Review of Hashing Algorithm Optimization for IoT Devices." International Journal of Intelligent Systems and Applications in Engineering 11.6s (2023): 205-231.

[5] Tamrakar, Ashish, et al. "extended review on internet of things (IoT) and its characterisation." Int. J. Health Sci 10 (2022): 8490-8500.

[6] Ansere, James Adu, et al. "Dynamic Resource Optimization for Energy-Efficient 6G-IoT Ecosystems." Sensors 23.10 (2023): 4711.

[7] Abdullah, Pavel Y., et al. "An hrm system for small and medium enterprises (sme) s based on cloud computing technology." International Journal of Research-GRANTHAALAYAH 8.8 (2020): 56-64.

[8] Sadeeq, Mohammed Mohammed, et al. "IoT and Cloud computing issues, challenges and opportunities: A review." Qubahan Academic Journal 1.2 (2021): 1-7.

[9] Raja Santhi, Abirami, and Padmakumar Muthuswamy. "Industry 5.0 or industry 4.0 S? Introduction to industry 4.0 and a peek into the prospective industry 5.0 technologies." International Journal on Interactive Design and Manufacturing (IJIDeM) 17.2 (2023): 947-979.

[10] Syed, Abbas Shah, et al. "IoT in smart cities: A survey of technologies, practices and challenges." Smart Cities 4.2 (2021): 429-475.

[11] Aljuboury, Anwer S., et al. "A New Nonlinear Controller Design for a TCP/AQM Network Based on Modified Active Disturbance Rejection Control." Complexity 2022 (2022).

[12] Tariq, Usman, et al. "A Critical Cybersecurity Analysis and Future Research Directions for the Internet of Things: A Comprehensive Review." Sensors 23.8 (2023): 4117.

[13] Ageed, Zainab Salih, et al. "A state of art survey for intelligent energy monitoring systems." Asian Journal of Research in Computer Science 8.1 (2021): 46-61.

[14] Uddin, Md Ashraf, et al. "A survey on the adoption of blockchain in iot: Challenges and solutions." Blockchain: Research and Applications 2.2 (2021): 100006.

[15] Kumar, K. Pradeep Mohan, et al. "Privacy Preserving Blockchain with Optimal Deep Learning Model for Smart Cities." CMC-COMPUTERS MATERIALS & CONTINUA 73.3 (2022): 5299-5314.

[16] Lee, Seungjin, et al. "Energy-Efficient AP Selection Using Intelligent Access Point System to Increase the Lifespan of IoT Devices." Sensors 23.11 (2023): 5197.

[17] Sadeeq, Mohammed AM, and Subhi Zeebaree. "Energy management for internet of things via distributed systems." Journal of Applied Science and Technology Trends 2.02 (2021): 59-71.

[18] Rashid, Zryan N., et al. "Distributed and Parallel Computing System Using Single-Client Multi-Hash Multi-Server Multi-Thread." 2021 1st Babylon International Conference on Information Technology and Science (BICITS). IEEE, 2021.

[19] Rejeb, Abderahman, et al. "The Internet of Things and the circular economy: A systematic literature review and research agenda." Journal of Cleaner Production 350 (2022): 131439.

[20] Al-Obaidi, Karam M., et al. "A review of using IoT for energy efficient buildings and cities: A built environment perspective." Energies 15.16 (2022): 5991.

[21] Alsharif, Mohammed H., et al. "Green IoT: A review and future research directions." Symmetry 15.3 (2023): 757.

[22] S. Benhamaid, A. Bouabdallah, and H. Lakhlef, "Recent advances in energy management for Green-IoT: An up-to-date and comprehensive survey," Journal of Network and Computer Applications, vol. 198, p. 103257, 2022.

[23] Ibrahim, R., S. R. Zeebaree, and Karwan Jacksi. "Semantic Similarity for Document Clustering using TFIDF and K-mean." Master’s Thesis (2020).

[24] S. K. Mousavi, A. Ghaffari, S. Besharat, and H. Afshari, "Security of internet of things based on cryptographic algorithms: a survey," Wireless Networks, vol. 27, pp. 1515-1555, 2021.

[25] P. P. Pittalia, "A comparative study of hash algorithms in cryptography," International Journal of Computer Science and Mobile Computing, vol. 8, no. 6, pp. 147-152, 2019.

[26] S. Long, "A comparative analysis of the application of hashing encryption algorithms for MD5, SHA-1, and SHA-512," in Journal of Physics: Conference Series, 2019, vol. 1314, no. 1: IOP Publishing, p. 012210.

[27] Razaq, Hussam Hatem Abdul, et al. "Designing and implementing an arabic programming language for teaching pupils." Journal of Southwest Jiaotong University 54.3 (2019).

[28] P. P. Pittalia, "A comparative study of hash algorithms in cryptography," International Journal of Computer Science and Mobile Computing, vol. 8, no. 6, pp. 147-152, 2019.

[29] A. Sideris, T. Sanida, and M. Dasygenis, "Hardware acceleration of SHA-256 algorithm using NIOS-II processor," in 2019 8th International Conference on Modern Circuits and Systems Technologies (MOCAST), 2019: IEEE, pp. 1-4.

[30] Sami, Teba Mohammed Ghazi, Subhi RM Zeebaree, and Sarkar Hasan Ahmed. "A Novel Multi-Level Hashing Algorithm to Enhance Internet of Things Devices’ and Networks’ Security." International Journal

of Intelligent Systems and Applications in Engineering 12.1s (2024): 676-696.

[31] B. U. I. Khan, R. F. Olanrewaju, M. A. Morshidi, R. N. Mir, M. L. B. M. Kiah, and A. M. Khan, "Evolution and analysis of secured hash algorithm (SHA) family," Malaysian Journal of Computer Science, vol. 35, no. 3, pp. 179-200, 2022.

[32] F. Hasmawati, Y. T. Samiha, A. Razzaq, and M. Anshari, "Understanding nomophobia among digital natives: Characteristics and challenges," Journal of Critical Reviews, vol. 7, no. 13, pp. 122-131, 2020.

[33] Alanazi, Turki M. "Embedded System Based Raspberry Pi 4 for Text Detection and Recognition." Intelligent Automation & Soft Computing 36.3 (2023).

[34] S. E. Mathe, A. C. Pamarthy, H. K. Kondaveeti, and S. Vappangi, "A review on raspberry pi and its robotic applications," in 2022 2nd International Conference on Artificial Intelligence and Signal Processing (AISP), 2022: IEEE, pp. 1-6.

[35] S. Monk, Raspberry pi cookbook. " O'Reilly Media, Inc.", 2022.

[36] Juan Garcia, Guðmundsdóttir Anna, Johansson Anna, Maria Jansen, Anna Wagner. Machine Learning for Predictive Maintenance and Decision Making in Manufacturing. Kuwait Journal of Machine Learning, 2(4). Retrieved from http://kuwaitjournals.com/index.php/kjml/article/view/212

[37] Prasad, A. K. ., M, D. K. ., Macedo, V. D. J. ., Mohan, B. R. ., & N, A. P. . (2023). Machine Learning Approach for Prediction of the Online User Intention for a Product Purchase. International Journal on Recent and Innovation Trends in Computing and Communication, 11(1s), 43–51. https://doi.org/10.17762/ijritcc.v11i1s.5992

[38] Tripathi, R.C., Gupta, P., Anand, R., Jayashankar, R.J., Mohanty, A., Michael, G., Dhabliya, D. Application of information technology law in India on IoT/IoE with image processing (2023) Handbook of Research on Thrust Technologies? Effect on Image Processing, pp. 135-150.