A COMPREHENSIVE ANALYSIS OF LIGHTWEIGHT HASH FUNCTIONS IN THE IOT ENVIRONMENT.

[1]Amita V Shah, [2]Dr. Sanjay M Shah, [3]Namit V Shah

[1]Ph.D Scholar, Computer/IT Engineering, Gujarat Technological University, Gujarat, India
[2]Professor & Head, Computer Engineering Dept., Government Engineering College, Rajkot, Gujarat, India
[3]Student, Computer Engineering Dept., L D College of Engineering, Ahmedabad, Gujarat, India

ABSTRACT

The exponential growth of the Internet of Things (IoT) in recent years is attributed to its profound impact on various aspects of daily life, particularly in critical applications such as healthcare, smart homes, smart cities, and broader smart infrastructure. This surge has drawn the attention of industries and researchers, prompting them to delve into the development of IoT technology. However, the vulnerability of IoT devices is evident due to their limited network capacities, minimal computing power, short battery life, and constrained data storage capabilities. The primary challenge hindering the widespread adoption of IoT systems is the imperative need for lightweight and energy-efficient security solutions. These solutions aim to safeguard smart devices and the sensitive data they store. Therefore, this study explores the integration of lightweight hash algorithms to verify data integrity and enhance the security of IoT systems. Given that hashing plays a pivotal role in establishing a resilient IoT framework, we opt to implement various hash techniques on a Raspberry Pi/Arduino/ESP32 device. In conclusion, this research offers a quantitative analysis to assess the performance of renowned hash functions applicable to lightweight IoT framework and resource constrained IoT devices.

Keyword : - *Lightweight IoT Framework, Hash Algorithms, SHA, Lightweight Cryptography*

1. INTRODUCTION

Embedded systems are purpose-built to execute specific functions while meeting real-time processing demands [1]. With the widespread availability and progress in internet connectivity, a new paradigm has surfaced—intertwining and connecting embedded devices with the internet. This phenomenon, commonly referred to as the Internet of Things (IoT), is gaining considerable momentum.

The Internet of Things functions as a framework where physical objects in the tangible world can establish connections with the web, creating a bidirectional communication channel. It entails integrating physical objects with electronics and corresponding software, enabling these objects to be remotely sensed, controlled, and analyzed within the existing network infrastructure. This dynamic fusion fosters direct interaction between the physical and digital realms, resulting in heightened efficiency, precision, and significant economic advantages.

From everyday consumer products to industrial machinery, commonplace items now feature internet connectivity, enabling robust data analytics and device control. This transformative shift is reshaping every aspect of human life, delivering unparalleled insights and capabilities across diverse domains.

Information security assumes a crucial role [1,2]. While IoT technology brings numerous advantages, it also introduces various security threats, including vulnerabilities in hard-coded security keys and the leakage of user privacy information [3,4]. Prior efforts have aimed to mitigate these security threats, employing methods such as static or dynamic analysis of firmware and source code in IoT devices to identify and address potential vulnerabilities [5]. Frameworks like Interference Mitigation Risk Aware (IMRA) have been proposed to counteract interference imposed by intruders, safeguarding the proper operation of passive RFID networks [6]. Study have also focused on designing new lightweight algorithms and optimizing existing cryptography algorithms [7,8].

2. SECURITY CONCERNS IN IOT

Amidst the myriad advantages associated with the Internet of Things (IoT), there are concurrent risks and safety concerns that necessitate robust security measures. The security techniques implemented must align with the intrinsic characteristics of IoT [2] [8] [9] [10] [11], taking into account the following key factors:

2.1 Long Device Life

To effectively address IoT security, it is imperative to devise mechanisms and techniques that are future-proof, ensuring sustained security over an extended device lifespan.

2.2 Lightweight Solution

Recognizing the inherent constraints of computational and power capabilities in embedded IoT devices, security arrangements should be lightweight. They must consume minimal power, meet memory requirements, and not impede processor performance.

2.3 Configurability

Given the widespread deployment of IoT devices across diverse locations, the security framework should exhibit dynamic configurability. This adaptability is essential for adjusting security requirements during execution.

2.4 Privacy during Communication

Establishing a secure communication framework within IoT is paramount. Robust mechanisms are needed to prevent unauthorized access, eavesdropping, and external interference to safeguard privacy.

2.5 Authentication

In a network of interconnected devices, each object must possess the capability to accurately identify and authenticate other objects. Authentication mechanisms are fundamental to ensuring the integrity and trustworthiness of the IoT ecosystem.

2.6 Integrity

Maintaining data integrity is critical during data exchange among different IoT devices. Ensuring the accuracy of data, verifying the sender's authenticity, and preventing tampering, whether intentional or unintentional, are essential safeguards.

2.7 Heterogeneity

Protocols designed for IoT security should be versatile enough to adapt to the diverse range of devices and situations encountered within the IoT landscape. This consideration acknowledges the inherent heterogeneity in IoT deployments.

Fig -1: Basic Security Concerns [1]

Addressing these factors collectively ensures the development of resilient security solutions that align with the unique challenges and requirements posed by the Internet of Things.

3. RELATED WORK

Cryptanalysis of these algorithms like MD5, SHA-0, SHA-1, SHA-2, SHA-3, is done and it was found that these algorithms are vulnerable to several attacks like collision resistance, birthday attack etc. only SHA-2 and SHA-3 algorithms came in to existence as till now and no attacks have been reported against these algorithms. SHA-256 is another name for SHA-2 or SHA-256 creates a longer, and thus more complex, hash. MD5, SHA-1 offers weaker security as it sometimes gives the same digest for two different data values, while SHA-2 produces a unique digest for every data value as a large number of combinations are possible in it. It is based on the cryptographic concept "Merkle–Damgård construction" and is considered highly secure. SHA-2 is published as official crypto standard in the United States. Regarding performance, SHA-2 has been widely adopted and optimized over the years, making it faster and more efficient on many platforms. On the other hand, SHA-3 is a relatively new algorithm, and its implementations may not be as mature or optimized as SHA-2[15]. However, as SHA-3 gains more adoption and optimization efforts increase, its performance is expected to improve.
In summary, both SHA-2 and SHA-3 are secure hash algorithms used for data integrity verification and other cryptographic applications. SHA-2 is a well-established algorithm family with different hash functions, while SHA-3 represents the latest addition to the Secure Hash Algorithm family. Researcher can choose SHA-2 for optimization such that they can use it in lightweight IoT environment. These all hashing algorithms are based on MD5 and basic structure is very similar to MD5.so, first we will see working of MD5.

3.1 MD5

The MD5 message-digest algorithm is a widely used hash function producing a 128-bit hash value. MD5 was designed by Ronald Rivest in 1991 to replace an earlier hash function MD4, and was specified in 1992 as RFC 1321. A 2013 attack by Xie Tao, Fanbao Liu, and Dengguo Feng breaks MD5 collision resistance in 218 time.With its distinctive features, MD5 proves instrumental in addressing security concerns within the Internet of Things (IoT). is It is evaluated for its compatibility and adherence to these imperative security features. The conclusion drawn asserts the viability and efficacy of employing MD5 in the embedded system, affirming its role in bolstering security within the IoT landscape [12].

The working of the MD5 algorithm can be understood clearly from Fig. 2. The following steps provide the details about the working of the algorithm [14].

Step 1. Append Padding Bits
Step 2. Append Message Length Information
Step 3. Initialize MD Buffer
Step 4. Process Message in 512-bit Blocks

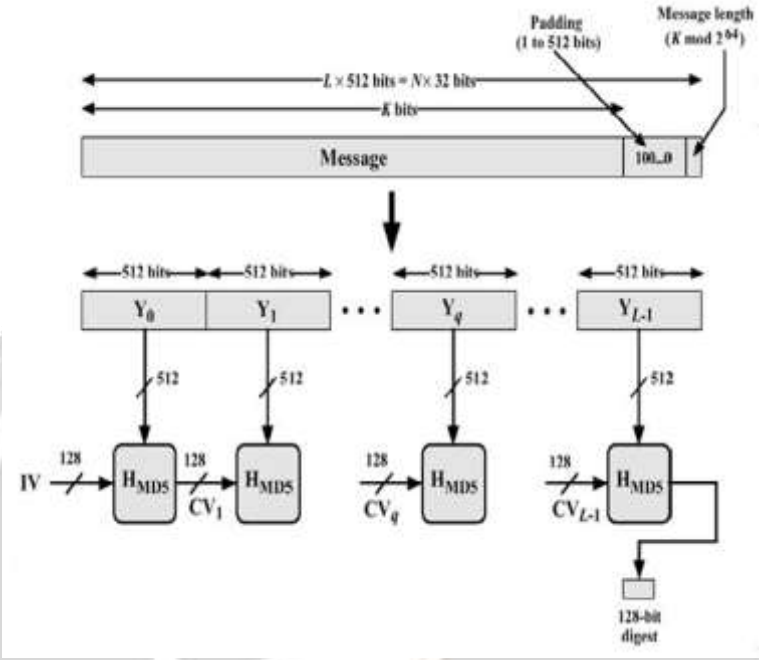

Fig -2: Basic architecture of MD5 [14]

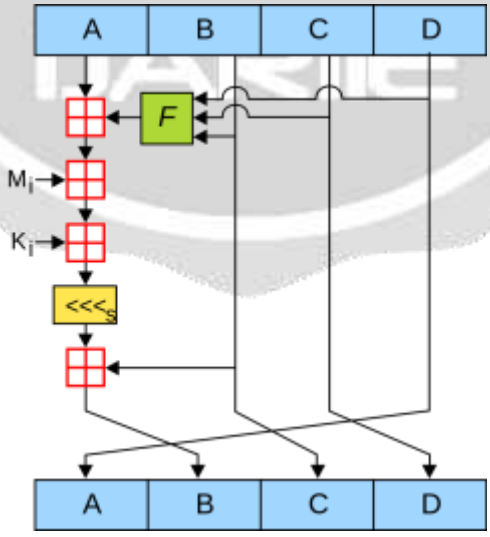

Fig -3: Main Hash Function (H_{MD5}) of MD5 [14]

Since 2004, MD5 has been largely replaced by more secure hashing algorithms like SHA, primarily due to vulnerabilities associated with collision attacks. However, the proposed system aims to reintegrate the MD5

algorithm for securing communication within the Internet of Things (IoT). By leveraging this system, the limitations posed by collision attacks can be mitigated, allowing the continued use of MD5 for IoT security while preserving its inherent advantages. This approach seeks to rejuvenate the utility of MD5 in specific contexts, offering a nuanced solution that balances both its strengths and weaknesses.

3.2 SHA-2

The SHA-2 family stands as a standardized hash collection endorsed by NIST and detailed in FIPS PUB 180-4 [18]. Comprising various algorithms, these are iterative, one-way hash functions designed to process input messages and generate condensed representations known as message digests. Each algorithm undergoes two key stages: preprocessing and hash computation. During preprocessing, the message is padded, parsed into m-bit blocks, and initialization values are set for subsequent hash computation. The hash computation process, stemming from the padded message, generates a message schedule. This schedule, coupled with functions, constants, and word operations, iteratively produces a series of hash values. The final hash value defines the message digest. Algorithms within SHA-2 differ in block sizes, word data, or message digest sizes, offering a spectrum of security levels [18]. Basic architecture is same as MD5 but there is difference in main hash function.

Discussed in Ref. [19], only SHA-256 and SHA-512 can be deemed original designs among all SHA-2 functions; others are variants with distinct initial hash values and truncated digests. Consequently, this work focuses on these two SHA-2 variants. Both adhere to the Merkle–Damgård structure with a Davies–Meyer compression function. SHA-256 operates on 32-bit words, processing a 512-bit message block to produce a 256-bit digest, providing 128-bit security against collisions [18]. SHA-512, operating on 64-bit words, processes a 1024-bit message block to yield a 512-bit digest, ensuring 256-bit security against collisions [18]. Both algorithms use eight working variables, with lengths of 32- and 64-bit in SHA-256 and SHA-512, respectively as [18].

SHA-256 is widely used in crypto currency systems, notably in Bitcoin mining [20], NIST considers it insecure for long-term use due to the constrained input small output (CISO) problem observed in Bitcoin mining, impacting hashing speed and cost [21]. This issue has been scrutinized in various studies [21,22], and SHA-3 (Keccak) has been proposed as a solution. Additionally, SHA-512 is anticipated to gain significance in the future, particularly with the potential emergence of quantum cryptanalysis threats [19], making it crucial in high-performance computing or the IoT field.

3.2 SHA-3

In the realm of security, both SHA-2 and SHA-3 stand out as highly secure and resilient against a variety of cryptographic attacks. However, SHA-3, with its distinctive design and underlying principles, exhibits greater resistance to specific types of attacks, notably collision attacks and length extension attacks, in comparison to SHA-2 [23].

Consider more secure than SHA-2, SHA-3 and its variants (SHA3-224, SHA3-256, SHA3-384, SHA3-512) are particularly esteemed for securing embedded subsystems, sensors, consumer electronic devices, and other systems utilizing symmetric key-based message authentication codes (MACs). Notably, SHA-3 demonstrates superior speed compared to its predecessors, boasting an average speed of 12.5 cycles per byte on an Intel Core 2 processor [23].

Among various hashing algorithms, SHA-3 is chosen as the default algorithm, derived from Keccak, which has given birth to the new Secure Hash Algorithm-3. It encompasses different SHA-3 models, including SHA-3(224-bit, 512-bit, 384-bit, and 256-bit), each characterized by distinct rounds and logical operations per round. The sponge function, a fundamental component, facilitates the absorption of input and subsequent squeezing to produce the desired output [23]

4. COMPARITIVE ANALYSIS OF VARIOUS HASHING ALGORITHMS

For instance, a novel tiny symmetric encryption algorithm (NTSA) enhances security for text file transfers in IoT networks by introducing dynamic key confusions for each encryption round [9]. Another example is the Function-

based Access Control scheme in IoT (IoT-FBAC), which uses an Identity-based Encryption (IBE) scheme [10]. While data masking and encryption algorithms are common solutions to protect sensitive information, In much of the aforementioned work, hash functions, belonging to one-way encryption algorithms that compress messages with arbitrary lengths into fixed-length digests [11], are frequently applied. Typical hash algorithms include MD5, SHA-0, SHA-1, SHA-2, SHA-3, and SM3 [12–16]. Due to security concerns, SHA-0, SHA-1, and MD5 are excluded from discussion. Table 1 presents a comparison between SHA-0, SHA-1, SHA-2 and SHA-3 [17]. To ensure hash algorithm security, the digest size should not be too short, as indicated in Table 1.

Table -1: SHA Comparison [12-16]

SHA	Block Size(bits)	Digest Size(bits)	Strength
MD5	512	128	Weak
SHA-0	512	160	Weak
SHA-1	512	160	Weak
SHA-2	512,1024	256,512	Strong
SHA-3	1600(Varies)	256,512	Strong

5. CONCLUSIONS

The experimental findings reveal that, in terms of hashing speed, SHA-2 functions demonstrate superior performance compared to other hash functions. Additionally, it conclude with the requirement of advancing more efficient lightweight hash functions, aligning with the recommendations put by NIST's lightweight cryptography project. In terms of performance, SHA-2 has achieved widespread adoption and optimization over the years, rendering it faster and more efficient across various platforms. Conversely, SHA-3 is a comparatively newer algorithm, and its implementations may not be as mature or optimized as those of SHA-2. Nevertheless, with the increasing adoption of SHA-3 and growing efforts in optimization, its performance is anticipated to improve over time. As the algorithm becomes more established and undergoes refinement, it is likely to bridge the performance gap with its predecessor, SHA-2.This recognition points to an open issue that warrants further investigation and exploration within the field.

6. REFERENCES

[1] Zhou, J.; Cao, Z.; Dong, X.; Vasilakos, A.V. Security and privacy for cloud-based IoT: Challenges. IEEE Commun. Mag. 2017, 55, 26–33.
[2] Surendran, S.; Nassef, A.; Beheshti, B.D. A survey of cryptographic algorithms for IoT devices. In Proceedings of the 2018 IEEE Long Island Systems, Applications and Technology Conference (LISAT), Bahawalpur, Pakistan, 23–25 October 2018; pp. 1–8.
[3] Hwang, Y.H. IoT Security and Privacy: Threats and Challenges. In Proceedings of the Acm Workshop on Iot Privac, Singapore, 14–17 April 2015.
[4] Zhou, W.; Jia, Y.; Peng, A.; Zhang, Y.; Liu, P. The Effect of IoT New Features on Security and Privacy: New Threats, Existing Solutions, and Challenges Yet to Be Solved. IEEE Internet Things J. 2018, 6, 1606–1616.
[5] Electronics 2019, Davidson, D.; Moench, B.; Ristenpart, T.; Jha, S. FIE on firmware: finding vulnerabilities in embedded systems using symbolic execution. In Proceedings of the 22nd USENIX conference on Security, Washington, DC, USA, 14–16 August 2013.
[6] Tsiropoulou, E.E.; Baras, J.S.; Papavassiliou, S.; Qu, G. On the Mitigation of Interference Imposed by Intruders in Passive RFID Networks. In Decision and Game Theory for Security, Proceedings of the International Conference, New York, NY, USA, 2–4 November 2016; Springer: Berlin/Heidelberg, Germany, 2016.

[7] Shi, Y.; Wei, W.; He, Z.; Fan, H. An ultra-lightweight white-box encryption scheme for securing resource-constrained IoT devices. In Proceedings of the the 32nd Annual Conference, Los Angeles, CA, USA, 5–9 December 2016.
[8] Buchmann, J.; Göpfert, F.; Güneysu, T.; Oder, T.; Pöppelmann, T. High-performance and lightweight lattice-based public-key encryption. In Proceedings of the 2nd ACM International Workshop on IoT Privacy, Trust, and Security, Xi'an, China 30 May–3 June 2016; pp. 2–9.
[9] Rajesh, S.; Paul, V.; Menon, V.G.; Khosravi, M.R. A secure and efficient lightweight symmetric encryption scheme for transfer of text files between embedded IoT devices. Symmetry 2019, 11, 293.
[10] Yan, H.; Wang, Y.; Jia, C.; Li, J.; Xiang, Y.; Pedrycz, W. IoT-FBAC: Function-based access control scheme using identity-based encryption in IoT. Future Gener. Comput. Syst. 2019, 95, 344–353
[11] State Cryptography Administration of China. Specification of SM3 Cryptographic Hash Function; State Cryptography Administration of China: Beijing, China, 2010.
[12] Zhao Yong-Xia, and Zhen Ge, "MD5 Research," in IEEE 2nd International Conference on Multimedia and Information Technology, pp. 271 - 273, April 2010.
[13] Hu, Y.; Wu, L.; Wang, A.; Wang, B. Hardware design and implementation of SM3 hash algorithm for financial IC card. In Proceedings of the 2014 Tenth International Conference on Computational Intelligence and Security, Yunnan, China, 15–16 November 2014; pp. 514–518.
[14] Priya.P, and B. Gopinathan, "Improving Security Based on Detecting Selfish Nodes using MD5 Encryption Algorithm in Manets," International Journal of Advanced Technology in Engineering and Sciences, vol. 4, pp. 1311-1317, Feb. 2016.
[15] Juliato M , Gebotys C H . Tailoring a Reconfigurable Platform to SHA-256 and HMAC through Custom Instructions and Peripherals. In Proceedings of the International Conference on Reconfigurable Computing and Fpgas, Cancun, Mexico, 9–11 December 2009.
[16] Algredo-Badillo, I.; Feregrino-Uribe, C.; Cumplido, R.; Morales-S, Oval, M. FPGA-based implementation alternatives for the inner loop of the Secure Hash Algorithm SHA-256. Microprocess. Microsyst. 2013, 37, 750–757. [CrossRef]
[17] Federal Information Processing Standards Publication 180-2. Announcing the Secure Hash Standard; US DoC/NIST: Gaithersburg, MD, USA, 2002.
[18] Q.H. Dang, Secure Hash Standard (SHS), Federal Information Processing Standards (FIPS-180-4), National Institute of Standards and Technology, Gaithersburg, MD, USA, 2015, https://doi.org/10.6028/NIST.FIPS.180-4.
[19] H. Cheng, D. Dinu, J. Großsch€adl, Efficient implementation of the SHA-512 hash function for 8-bit AVR microcontrollers, in: J.L. Lanet, C. Toma (Eds.), SECITC 2018: Innovative Security Solutions for Information Technology and Communications, Springer, Cham, Switzerland, 2018, pp. 273–287.
[20] X. Wang, X. Zha, W. Ni, et al., Survey on blockchain for Internet of things, Comput. Commun. 136 (2019) 10–29.
[21] N.T. Courtois, M. Grajek, R. Naik, Optimizing SHA256 in Bitcoin mining, in: Z. Kotulski, B. Księzopolski, K. Mazur (Eds.), CSS 2014: Cryptography and Security _ Systems 448, Springer, Berlin, Heidelberg, Germany, 2014, pp. 131–144.
[22] L.V.T. Duong, N.T.T. Thuy, L.D. Khai, A fast approach for bitcoin blockchain cryptocurrency mining system, Integration 74 (2020) 107–114.
[23] M.J. Dworkin, SHA-3 standard: permutation-based hash and extendable-output functions, National Institute of Standards and Technology FIPS-202 (2015).
[24] G. Bertoni, J. Daemen, M. Peeters, G. Van Assche, Keccak sponge function family main document 3 (30) (2009) 320–337. Submiss. to NIST (Round 2). [25] J. Guo, T. Peyrin, A. Poschmann, The PHOTON family of lightweight hash functions, in: P. Rogaway (Ed.), Advances in Cryptology—CRYPTO 2011, Springer, Berlin, Heidelberg, Germany, 2011, pp. 222–239.