Lightweight Parabola Chaotic Keyed Hash Using SRAM-PUF for IoT Authentication

Nattagit Jiteurtragool[1], Jirayu Samkunta[2], Patinya Ketthong[3]
Department of Computer and Information Sciences-Faculty of Applied Science,
King Mongkut's University of Technology, North Bangkok, Bangkok, Thailand[1]
Graduate School of Science and Technology, Gunma University, Kiryu, Japan[2]
Intelligent Electronics System Laboratory (IES), Thai-Nichi Institute of Technology, Bangkok, Thailand[3]

***Abstract*—This paper introduces a lightweight and efficient keyed hash function tailored for resource-constrained Internet of Things (IoT) environments, leveraging the chaotic properties of the Parabola Chaotic Map. By combining the inherent unpredictability of chaotic systems with a streamlined cryptographic design, the proposed hash function ensures robust security and low computational overhead. The function is further strengthened by integrating it with a Physical Unclonable Function (PUF) based on SRAM initial values, which serves as a secure and tamper-resistant source of device-specific keys. Experimental validation on an ESP32 microcontroller demonstrates the function's high sensitivity to input variations, exceptional statistical randomness, and resistance to cryptographic attacks, including collisions and differential analysis. With a mean bit-change probability nearing the ideal 50% and 100% reliability in key generation under varying conditions, the system addresses critical IoT security challenges such as cloning, replay attacks, and tampering. This work contributes a novel solution that combines chaos theory and hardware-based security to advance secure, efficient, and scalable authentication mechanisms for IoT applications.**

***Keywords*—SRAM PUF; PUF key generation; chaotic keyed hash; device authentication; discrete-time chaotic**

I. Introduction

The Internet of Things (IoT) has revolutionized the way we interact with technology, connecting billions of devices globally to create a vast network of interconnected systems. IoT extends to various applications, including smart homes, healthcare, industrial automation, and transportation systems [1-2]. For instance, IoT-enabled medical devices can monitor patients in real-time, while industrial sensors optimize manufacturing processes. The ability of IoT to enhance operational efficiency, reduce costs, and enable innovative services underscores its importance in modern technological advancements. However, the rapid proliferation of IoT devices also brings significant challenges, particularly in managing the security of these interconnected networks [3-4].

Device authentication is a fundamental aspect of IoT security, ensuring that only legitimate devices can communicate within the network. Conventional methods for authentication often rely on cryptographic keys stored in memory, which are vulnerable to physical attacks, cloning, and software-based exploits. These issues highlight the need for more robust and tamper-resistant mechanisms for IoT security [5-6].

However, IoT devices face unique constraints compared to traditional IT systems. These constraints include limited computational resources, such as low processing power, minimal memory, and restricted energy availability [7]. These limitations necessitate lightweight and efficient security mechanisms that do not overwhelm the device's capabilities. Moreover, IoT devices are often deployed in physically exposed environments, making them more susceptible to tampering, theft, and side-channel attacks. Unlike traditional IT systems, which are often housed in secure, controlled environments, IoT devices may operate in untrusted or hostile settings, increasing the potential attack surface [8]. As a result, IoT security mechanisms must address these heightened risks while remaining compatible with resource-constrained hardware.

In this paper, we propose a novel approach that integrates SRAM-based PUFs with a keyed hash function based on the Parabola Chaotic Map. This combination provides a lightweight and secure mechanism for device authentication in resource-constrained IoT environments. By leveraging the inherent randomness of SRAM PUFs and the entropy-enhancing properties of the chaotic hash function, our method ensures robust security while maintaining computational efficiency over a resource-constrained environment of IoT.

The rest of this paper is organized as follows: Section II provides an in-depth discussion of the background concepts, including an overview of IoT security challenges, Physical Unclonable Functions (PUFs), and chaotic hash functions. Section III introduces the proposed system, elaborating on the integration of SRAM-based PUFs with the Parabola Chaotic Map-based keyed hash function and detailing its implementation on the ESP32 microcontroller. Section IV presents the results and analysis of the system's performance. Finally, Section V concludes the paper by summarizing key findings and proposing directions for future research.

II. Background

A. IoT Security Challenges

Security is a critical concern in IoT due to the sensitive data these devices handle and their deployment in diverse environments. Fig. 1 shows a classification of IoT security threats along with the security approaches for each IoT layer [9-10]. As IoT architectures are commonly divided into three layers [11]: The perception layer, which includes sensors and actuators, involves data collection through directly interfaces

with the physical environment, faces risks such as interception of data transmitted in and out of devices, physical tampering or node capture attacks. The network layer, which is responsible for transmitting data between devices, gateways, and cloud platforms, may have vulnerability to data interception during the transmission, or being attack by overloading the network traffic to make it unavailable. Lastly, the application layer, which able to provide the user interfacing control and processes the collected data, can be compromised by software vulnerabilities and unauthorized access. Unauthorized access to IoT devices can lead to data breaches, operational disruptions, and even physical damage in critical systems.

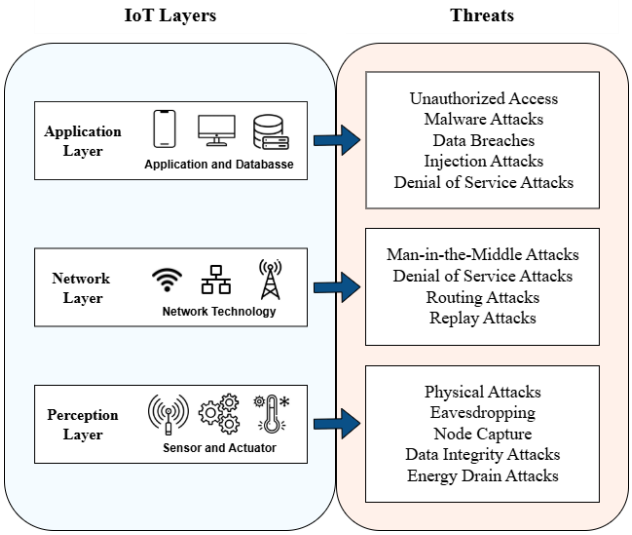

Fig. 1. IoT layers, and security threats.

B. Physical Unclonable Functions (PUFs)

Physical Unclonable Functions (PUFs) have emerged as a promising solution to IoT security challenges [12-13]. A PUF leverages the intrinsic physical variations in hardware components to generate unique and unclonable identifiers [14]. Among the various types of PUFs [15], SRAM-based PUFs are particularly attractive due to their simplicity and widespread availability in existing hardware [16-17]. When powered on, SRAM cells exhibit random initialization values influenced by manufacturing variations, making them an ideal candidate for generating device-specific keys. The stability of these values can be enhanced through error correction, enabling consistent and reliable use for cryptographic purposes, including as keys for keyed hash functions.

PUFs leverage manufacturing variations to generate unique and unclonable responses from hardware components. SRAM PUFs use the power-up state of uninitialized SRAM cells as a source of randomness. These initial values are device-specific and can be stabilized using error correction mechanisms for consistent key generation. PUFs eliminate the need to store cryptographic keys, thereby reducing risks associated with key theft or compromise. SRAM-based PUFs, specifically, are advantageous because of their availability in existing devices and their ability to function without modifications to the hardware design.

C. Hash Function

Hash functions have been implemented in the contexts of cybersecurity and information security applications [18-20], such as ensuring data integrity, authentication, digital signatures [21], protocol encryption [22], number generation [23], password security [24], or blockchain applications [25]. Hash functions are generally categorized into two types: unkeyed and keyed. Unkeyed hash functions rely solely on the input message to generate the hash value. On the contrary, Keyed hash functions, which incorporate a secret key into the hashing process, can provide enhanced security by protecting against tampering and brute-force attacks. Unkeyed hash functions such as MD5 and SHA-1, which are widely used in security applications and protocols, have primarily relied on logical operations and multi-round iterations.

Aside from hash functions, chaotic systems have gained significant traction in the field of cryptography over the last decade [26-27]. Chaotic systems are defined by their deterministic nature combined with an inherent unpredictability. These systems exhibit extreme sensitivity to initial conditions, long-term unpredictability, and complex yet non-random behavior. Such characteristics make them ideal for cryptographic applications [28], where unpredictability and high entropy are essential for ensuring secure operations. Among the tools derived from chaotic systems, chaotic maps stand out as mathematical functions capable of generating sequences with these properties. Examples include the Logistic Map, Tent Map, and Parabola Map, each of which provides unique advantages in terms of randomness and computational efficiency. The ability to generate pseudo-random values from simple mathematical operations makes chaotic maps particularly valuable for lightweight and efficient cryptographic systems, especially in resource-constrained environments like IoT.

Hashing efficiency is contingent upon inherent ciphers which necessarily require extensive computation processes. Combining input sensitivity with the entropy-generating properties of chaotic systems resulted in keyed hash functions such as the Chaotic Map-based approach [29-31], which offers a lightweight and secure solution for IoT environments. These functions ensure robust performance even in resource-constrained devices, making them a suitable choice for integrating with other security approaches such as hardware-based mechanisms like PUFs [32-33]. Moreover, the chaotic map-based approach provides an inherent randomness that enhances cryptographic strength, making it resilient against cryptanalysis. This approach also enables adaptability in various IoT applications by allowing parameter adjustments, ensuring that it can balance security needs with computational overhead effectively.

III. Proposed System

A. SRAM-PUF Key Generation on ESP32

The ESP32 is a low-cost, low-power, and highly versatile microcontroller system-on-chip (SoC) designed by Espressif Systems [34]. It is widely used in Internet of Things (IoT) applications due to its powerful features, extensive connectivity options, and efficient performance. The ESP32 microcontroller features a versatile internal memory architecture that includes

multiple Static Random Access Memory (SRAM) regions. The primary SRAM, divided into three memory blocks—SRAM0, SRAM1, and SRAM2—totals 520 KB. These memory blocks are shared between instruction and data storage, enabling efficient use of memory resources. SRAM0 and SRAM1 are typically used for both instruction and data storage, while SRAM2 is primarily reserved for general-purpose data storage. The instruction storage is accessible through the instruction memory bus (IRAM), allowing executable code to run efficiently. Conversely, the data storage, accessed through the data memory bus (DRAM), is non-executable and dedicated to runtime data. Fig. 2 illustrates the internal SRAM memory architecture of the ESP32.

In this paper, the ESP32 microcontroller is used as a test platform. The device's on-chip SRAM is leveraged for PUF generation, while its computational capabilities handle the chaotic hash function. Memory block from SRAM2 is chosen for the PUF generation. This region is preferred because it remains uninitialized during the boot process, preserving its unique power-up state. These states, influenced by inherent manufacturing differences, exhibit sufficient entropy to serve as a reliable source for PUF generation. By reserving this address range exclusively for PUF purposes, the implementation ensures that other system operations do not interfere with the PUF response. This allocation also simplifies access and management of the PUF-specific memory blocks while maintaining the system's overall efficiency and stability.

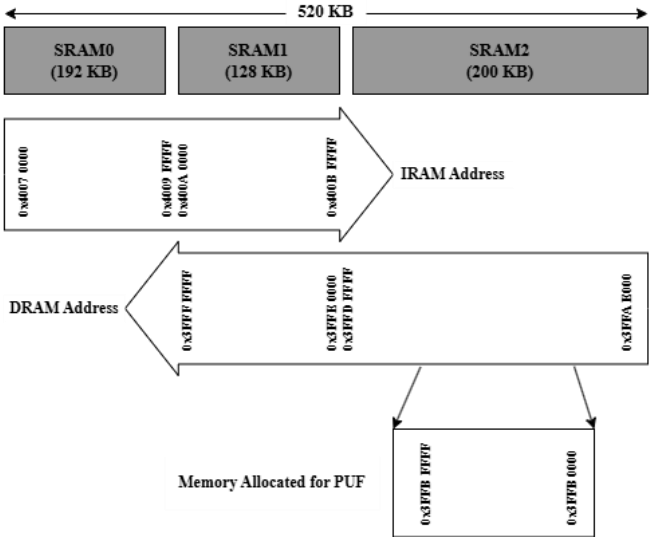

Fig. 2. Internal SRAM memory allocation of ESP32 chip.

The ESP32 microcontroller's low power consumption and computational efficiency make it an ideal platform for implementing this SRAM-based PUF mechanism. Its versatile memory architecture supports the storage and processing of the raw and stabilized PUF responses, while its dual-core processor efficiently handles the computations required for Error Correction to stabilize the output.

The key workflow of the implementation of SRAM-PUF Key Generation on ESP32 involves extracting device-specific responses from SRAM and stabilizing these responses using a Majority Vote Key instead of Error Correction Codes (ECC). This process ensures unique, tamper-resistant authentication for IoT devices, addressing the challenges of environmental variability and hardware noise. By leveraging both techniques, the system achieves a high degree of stability and reliability while maintaining computational efficiency suitable for resource-constrained IoT applications.

During device power-up, uninitialized SRAM cells generate random values due to manufacturing variations. These values are read and processed to form a raw binary response. However, environmental factors such as temperature fluctuations, voltage variations, and hardware aging can introduce inconsistencies in the raw responses across multiple power cycles. To address these challenges, the Majority Vote Key approach is applied.

With Majority Vote Key, the binary representations of the SRAM data undergo majority voting over a defined number of iterations, enhancing the robustness of the generated key against noise and instability. This process effectively averages out noise and random bit flips, ensuring that transient errors caused by environmental factors do not impact the final response. The output of this stage is a binary response that is significantly more stable and reliable than any single raw response. The stabilized output then serves as the key for the chaotic hash function.

B. Parabola Chaotic Keyed Hash Function

The Generalized Parabola Chaotic Map, developed in our previous work [35], serves as a robust randomness source for the proposed keyed hash function. This map builds upon well-known chaotic systems like the tent map, logistic map, and Gauss map, offering enhanced entropy generation and sensitivity to initial conditions. The mathematical definition of the Parabola Chaotic Map is as follows:

$$x_{n+1} = \mp A f_{\mathrm{NL}}(B x_n) \pm C \tag{1}$$

where the parameters *A*, *B* and *C* are real constants, and the $f_{NL}(x)$ is a parabola function. These parameters contribute to the map's chaotic behavior, ensuring its suitability for cryptographic applications.

The Parabola Chaotic Map-based keyed hash function is specifically designed for resource-constrained environments, making it an ideal choice for IoT applications. This lightweight cryptographic mechanism capitalizes on the pseudo-random behavior and high entropy characteristics of the chaotic map, which enhances its resistance to attacks such as collision, preimage, and differential cryptanalysis. Furthermore, the efficient computational requirements ensure compatibility with IoT devices that often have limited processing power and energy resources.

The hash function operates through the following steps, illustrated in Fig. 3:

1) Input handling: The input data and the stabilized keys from the SRAM-PUF are prepared. If the input length is not a multiple of 8, it is padded with zeros to meet the length requirement.

2) Initialization: A chaotic sequence is initialized, with the number of iterations determined by the length of the padded input.

3) Chaos mapping loop: The chaotic sequence is iteratively calculated, where each new value is derived from the previous value and a portion of the input bits.

4) Continuation of chaos mapping: Additional values are generated as needed to complete the hash output.

5) Output generation: The final chaotic sequence is processed to extract the relevant portion, forming the hash output.

The streamlined structure of the hash function ensures high efficiency while maintaining robust cryptographic properties. The nonlinear dynamics of the Parabola Chaotic Map enable the generation of highly unpredictable outputs, providing a secure foundation for cryptographic applications in IoT environments.

C. IoT Device Authentication Mechanism

The combination of the parabola chaotic map-based keyed hash function with SRAM-PUF technology is aimed to create a secure and lightweight authentication mechanism. The SRAM-PUF, implemented on an ESP32 microcontroller, generates a stabilized output key derived from the intrinsic physical variations of the SRAM memory. This stabilized key serves as the secret input to the keyed hash function, which, in turn, generates unique and unpredictable authentication tokens. The system leverages these tokens to implement a secure authentication mechanism.

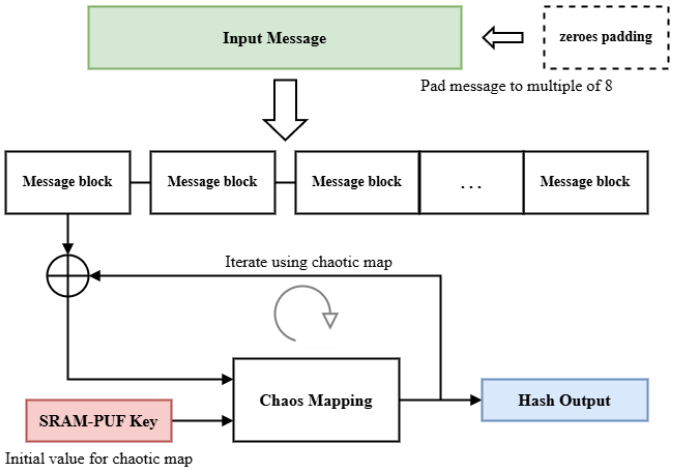

Fig. 3. SRAM-PUF based chaotic keyed hash.

As challenge-response pair (CRP) was chosen as an authentication mechanism, the authenticator sends a challenge to the device during an authentication session. The device combines this challenge with its stabilized SRAM-PUF key and processes through the chaotic keyed hash function. The result is an authentication token that is unique to the device and the specific challenge. This response is then sent back to the authenticator for authentication. Since the SRAM-PUF-derived key is unclonable and never directly transmitted, the system ensures that even if an attacker intercepts the challenge or response, they cannot reconstruct the key or generate valid tokens. The authentication mechanism is shown in Fig. 4.

The verifier, having the original challenge and a stored copy of the expected key or hash behavior, computes the expected token using the same keyed hash function. If the computed token matches the received token, the device is authenticated successfully. This process ensures the following:

1) Device uniqueness: The unclonable nature of the SRAM-PUF key ensures that each device produces a unique response, making it nearly impossible to replicate the authentication behavior of another device.

2) Resistance to replay attacks: Since the response is dynamically generated for each challenge, replaying a previously captured response will fail to authenticate the device.

3) Lightweight operation: The combination of SRAM-PUF and chaotic keyed hash function provides a computationally efficient authentication mechanism suitable for IoT devices with limited resources.

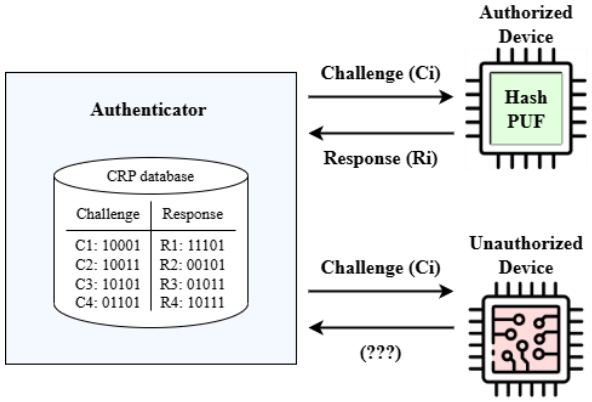

Fig. 4. The authentication mechanism.

By integrating the parabola chaotic map-based keyed hash function with SRAM-PUF, the proposed system achieves a high degree of security and efficiency, ensuring robust authentication for IoT devices in diverse and potentially hostile environments.

IV. Results and Analysis

This section evaluates the performance of the Parabola Chaotic Map-based keyed hash function and the SRAM-PUF mechanism implemented on the ESP32 microcontroller on security, efficiency, and robustness in IoT authentication.

A. Sensitivity and Reliability Analysis

The reliability of the SRAM-PUF is critical factor in determining the robustness of the proposed keyed hash function. In this study, experimental test was conducted using two ESP32 boards with the same model (ESP32WROOM32).

Reliability measures the consistency of the PUF response to the same challenge across multiple instances and varying environmental conditions, such as temperature fluctuations and power supply variations. The proposed system leverages the Majority Vote Key technique instead of traditional error correction codes, ensuring consistent key generation without significant computational overhead.

The evaluation of reliability involved extensive testing with two ESP32 boards. To assess reliability, a set of predefined challenges was repeatedly presented to the PUF, and the responses were collected over 100 iterations per challenge. The

collected responses were then analyzed to identify any inconsistencies.

Remarkably, the PUF demonstrated a reliability of 100%, consistently producing identical responses to the same challenges across all tests. This perfect reliability validates the robustness of the Majority Vote Key mechanism, which effectively mitigated the effects of noise and variability. Over prolonged operational periods, the mechanism maintained its performance, demonstrating resilience against potential aging effects in the SRAM hardware. This result validates the unparalleled reliability of the proposed SRAM-PUF mechanism, reinforcing its suitability for an integration with the parabola chaotic map-based keyed hash function.

Consequently, to demonstrate exceptional sensitivity to minor changes in the input message of the proposed keyed hash function, a series of experiments were conducted such as character replacements, additions, or deletions.

Case1: The Original message: “This is a simple absolute chaos based keyed hash function."

Case2: Replace the first character of the original message ‘‘T’’ by ‘‘t’’.

Case3: Add a space to the end of the original message.

Case4: Delete the full stop symbol at the end of the message.

The corresponding 128-bit hash values for each input message are presented in hexadecimal format as follows.

Case1: FB47 2C55 6481 D81D 379B 13F4 7616 733C

Case2: 80FB 471D 72B4 71C6 A735 7FA0 7868 CF77

Case3: 81E9 1D90 FB39 0FB4 5028 1F38 00D1 55AF

Case4: 90E9 1D71 D72B 5639 6F4B 75F9 195B 2CD1

The graphical display of binary sequences is shown in Fig. 5.

The results clearly demonstrate the proposed hash function exceptional sensitivity to such alterations and its suitability for tamper-resistant IoT authentication systems and dependable performance across diverse scenarios.

B. Collision analysis

The collision test consists in computing the difference between the hash values obtained from the original message and from a modified version of this message. The difference is computed based on the following formula:

$$d = \sum_{i=1}^{N} |dec(m_i) - dec(m_i')| \qquad (2)$$

where m_i and m_i' is the ith ASCII character of the original and the new hash value respectively, while the *dec*(.) converts an ASCII character to its decimal value.

Table I, show the minimum, maximum and mean values of the absolute difference of original and new hash values, where simulation repeat N = 10,000 time.

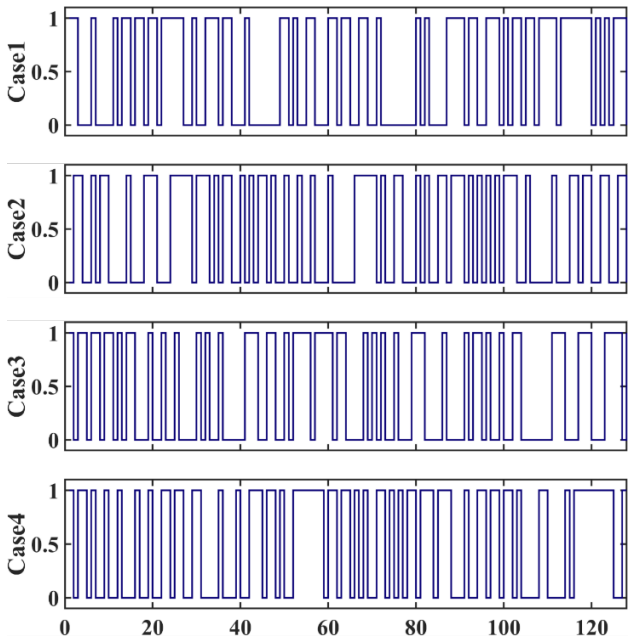

Fig. 5. The 128-bit hash values under different conditions.

TABLE I. COLLISION ANALYSIS OF HASH ALGORITHMS

Hash Algorithm	Min	Max	Mean
MD5(128bit)	590	2074	1304
SHA-1(160bit)	795	2730	1603
Proposed scheme (128bit)	549	2590	1387
Proposed scheme (160bit)	602	2702	1647
Proposed scheme (256bit)	1509	4005	2710

C. Statistical Analysis

Confusion and diffusion are critical metrics for evaluating the performance of hashing algorithms. To conduct a statistical analysis of these properties, the following experiments were performed:

- A random message of length L = n*50 was generated, where n represents the desired hash value length.
- A single bit of the original message was randomly selected and flipped, and a new n-bit hash value was calculated.
- The two n-bit hash values were compared to determine the number of changed bits.
- This simulation was repeated N times for different values of N (256, 512, 1024, 2048, and 10,000) and varying hash value lengths n (128, 256, and 512).

The following statistics were collected from the results obtained (see Tables II, III and IV):

- Minimum number of changed bits (B_{min}):
- Maximum changed bit (B_{max}):
- Mean changed bit $\overline{(B)}$:
- Mean changed probability (P):

- Standard deviation of the changed bit (ΔB):
- Standard deviation (ΔP):

TABLE II. STATISTICAL RESULT FOR A 128-BIT HASH VALUE

128-bit	*N* times				
	256	512	1024	2048	10000
B_{min}	48	41	38	21	8
B_{max}	80	83	83	96	86
$\overline{B}$	64.40	63.93	63.73	63.76	63.83
P (%)	50.31	49.95	49.79	49.82	49.87
ΔB	5.33	6.05	6.11	6.27	6.27
ΔP (%)	4.16	4.73	4.77	4.9	4.89

TABLE III. STATISTICAL RESULT FOR A 256-BIT HASH VALUE

256-bit	*N* times				
	256	512	1024	2048	10000
B_{min}	95	95	86	82	80
B_{max}	151	153	154	154	158
$\overline{B}$	127.93	127.80	127.35	127.40	127.41
P (%)	49.97	49.92	49.75	49.77	49.77
ΔB	8.95	8.99	9.28	8.93	9.16
ΔP (%)	3.50	3.51	3.63	3.49	3.58

TABLE IV. STATISTICAL RESULT FOR A 512-BIT HASH VALUE

512-bit	*N* times				
	256	512	1024	2048	10000
B_{min}	195	195	195	195	175
B_{max}	288	288	288	296	296
$\overline{B}$	255.17	254.87	254.71	254.82	254.80
P (%)	49.84	49.78	49.75	49.77	49.76
ΔB	12.64	12.52	12.61	12.86	12.94
ΔP (%)	2.47	2.45	2.46	2.51	2.53

The distribution of the number of changed bits can be analyzed using both a plot of the distribution of changed bits and a histogram as shown in Fig. 6 (a) and (b), respectively. This illustrates the distributions for n = 256 and N = 10000. The results indicate that the number of changed bits is consistently close to n/2 or 50%, and the histograms exhibit a normal distribution with a mean of n/2. This corresponds to a mean changed bit number and a mean changed probability that are very close to the ideal values of n/2 bits and 50%, respectively.

To achieve high security and prevent major attacks, the parabola chaotic keyed hash function based on SRAM-PUF key demonstrate high sensitivity to input changes, with minor variations in the input producing significant differences in the hash output. Statistical analyses confirm that the reliability of the proposed system again a single bit changes in the challenge would result in nearly 50% change in responses, which is an ideal state.

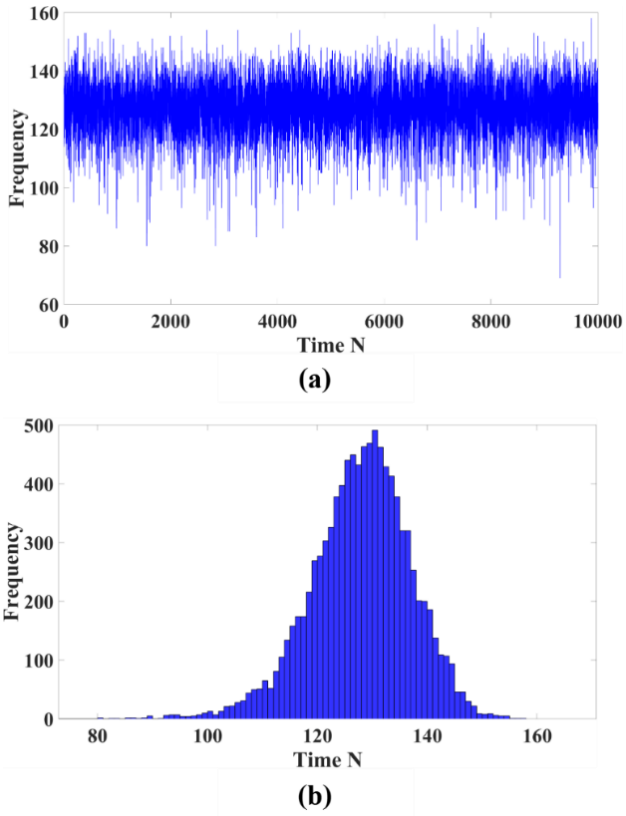

Fig. 6. Plot distribution of the number of bits changed (*a*) and Histogram distribution of the number of changed bits (*b*) for n = 256 and N = 10000.

V. CONCLUSION

This paper introduces a novel authentication mechanism that combines the hardware-level uniqueness of SRAM-based Physical Unclonable Functions (PUFs) with the cryptographic robustness of a Parabola Chaotic Map-based keyed hash function. This integration is designed to meet the stringent requirements of security, reliability, and computational efficiency for Internet of Things (IoT) applications operating in resource-constrained environments.

The proposed approach leverages the inherent physical variations in SRAM to generate unclonable keys, ensuring device-specific uniqueness without the need for external key storage. The chaotic hash function, built on the properties of the Parabola Chaotic Map, enhances security through its high sensitivity to input changes and entropy-generating capabilities, offering strong resistance to cryptographic attacks such as collision, preimage, and differential analysis.

Experimental evaluations confirm the system's effectiveness. The Majority Vote Key technique, employed to stabilize SRAM responses, demonstrates 100% reliability. Statistical analyses further validate the hash function's robustness, achieving near-ideal diffusion and confusion properties with a mean bit-change probability close to 50% in response to input variations. These results underscore the system's capability to deliver robust security and consistent performance.

This work contributes to the field of IoT security by providing a lightweight, tamper-resistant solution that mitigates risks associated with cloning, replay attacks, and physical tampering. The simplicity of the proposed design ensures compatibility with the limited resources of IoT devices, while its high reliability and security make it a promising candidate for broader adoption.

Future research will explore further optimizations of the system's computational and energy efficiency. Additionally, extending the application of this approach to other cryptographic contexts and implementing hardware prototypes for large-scale testing will provide further validation and refinement of the proposed solution.

REFERENCES

[1] Minerva, Roberto, Abyi Biru, and Domenico Rotondi, "Towards a definition of the Internet of Things (IoT). *IEEE Internet Initiative*, vol. 1, pp. 1-86, 2015

[2] M. Mohammadi, M. Aledhari, A. Al-Fuqaha, Internet of things: a survey on enabling technologies, protocols and applications. *IEEE Commun. Surveys Tuts*. vol. 17, pp. 2347–2376, 2015

[3] Shafiq, M., Gu, Z., Cheikhrouhou, O., Alhakami, W., & Hamam, H. (2022). The Rise of "Internet of Things": Review and Open Research Issues Related to Detection and Prevention of IoT-Based Security Attacks. *Wireless Communications and Mobile Computing*, *2022*(1), 8669348.

[4] Abosata, N., Al-Rubaye, S., Inalhan, G., & Emmanouilidis, C. (2021). Internet of things for system integrity: A comprehensive survey on security, attacks and countermeasures for industrial applications. *Sensors*, *21*(11), 3654.

[5] S. Dargaoui, et al., "Internet of Things Authentication Protocols: Comparative Study," *Computers, Materials & Continua*, vol. 79, no. 1, 2024.

[6] M. Trnka, et al., "Systematic review of authentication and authorization advancements for the Internet of Things," *Sensors*, vol. 22, no. 4, p. 1361, 2022.

[7] F. Pereira, et al., "Challenges in resource-constrained IoT devices: Energy and communication as critical success factors for future IoT deployment," *Sensors*, vol. 20, no. 22, p. 6420, 2020.

[8] T. Sasi, et al., "A comprehensive survey on IoT attacks: Taxonomy, detection mechanisms and challenges," *Journal of Information and Intelligence*, vol. 2, no. 6, pp. 455-513, 2024.

[9] S. Shin and T. Kwon, "A privacy-preserving authentication, authorization, and key agreement scheme for wireless sensor networks in 5G-integrated Internet of Things," IEEE Access, vol. 8, pp. 67555–67571, 2020.

[10] N. Abosata, S. Al-Rubaye, G. Inalhan, and C. Emmanouilidis, "Internet of things for system integrity: A comprehensive survey on security, attacks and countermeasures for industrial applications," *Sensors*, vol. 21, no. 11, p. 3654, 2021.

[11] Institute of Electrical and Electronics Engineers, "IEEE Standard for an Architectural Framework for the Internet of Things (IoT)," IEEE Std 2413-2019, pp. 1-269, 2020.

[12] C. Herder, M. -D. Yu, F. Koushanfar and S. Devadas, "Physical Unclonable Functions and Applications: A Tutorial," in *Proceedings of the IEEE*, vol. 102, no. 8, pp. 1126-1141, Aug. 2014

[13] A. Shamsoshoara, et al., "A survey on physical unclonable function (PUF)-based security solutions for Internet of Things," *Computer Networks*, vol. 183, p. 107593, 2020.

[14] U. Rührmair et al., "PUFs: Myth, fact or busted? A security evaluation of physically unclonable functions (PUFs) cast in silicon," *IEEE Trans. Dependable Secure Comput*., vol. 10, no. 3, pp. 193–206, 2013

[15] F. Farha, H. Ning, K. Ali, L. Chen, and C. D. Nugent, "SRAM-PUF based entities authentication scheme for resource-constrained IoT devices," *IEEE Internet of Things Journal*, vol. 0, pp. 1-10, 2020.

[16] G. E. Suh and S. Devadas, "Physical unclonable functions for device authentication and secret key generation," in *Proc. 44th Annu. Design Autom. Conf*., pp. 9-14, 2007.

[17] Böhm, Christoph, Maximilian Hofer, and Wolfgang Pribyl. "A microcontroller sram-puf." in *2011 5th International Conference on Network and System Security*, pp. 269-273. 2011.

[18] ISO/IEC 10118-3:2004, "Information technology - Security techniques - Hash-functions - Part 3: Dedicated hash-functions," 2004.

[19] A. Menezes, P. Van Oorschot, and S. Vanstone, "Handbook of Applied Cryptography," CRC Press, 1996.

[20] D. R. Stinson, "Cryptography: Theory and Practice," CRC Press, 2005.

[21] J. Buchmann, E. Dahmen, and M. Szydlo, "Hash-based digital signature schemes," in *Post-quantum cryptography*, Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 35-93, 2009.

[22] Y. Yang, et al., "Research on the hash function structures and its application," *Wireless Personal Communications*, vol. 94, pp. 2969-2985, 2017.

[23] Van der Leest, Vincent, Erik Van der Sluis, Geert-Jan Schrijen, Pim Tuyls, and Helena Handschuh. "Efficient implementation of true random number generator based on SRAM PUFs." In *Cryptography and Security: From Theory to Applications: Essays Dedicated to Jean-Jacques Quisquater on the Occasion of His 65th Birthday*, pp. 300-318, 2012.

[24] Q. Guo, et al., "PUFPass: A password management mechanism based on software/hardware codesign," *Integration*, vol. 64, pp. 173-183, 2019.

[25] P. Velmurugadass, et al., "Enhancing Blockchain security in cloud computing with IoT environment using ECIES and cryptography hash algorithm," *Materials Today: Proceedings*, vol. 37, pp. 2653-2659, 2021.

[26] S. N. Elaydi, Discrete Chaos: With Applications in Science and Engineering. Chapman and Hall/CRC, 2007.

[27] S. Boccaletti, C. Grebogi, Y.-C. Lai, H. Mancini, and D. Maza, "The control of chaos: theory and applications," *Physics Reports*, vol. 329, no. 3, pp. 103-197, 2000.

[28] N. Jiteurtragool, T. Masayoshi, and W. San-Um, "Robustification of a one-dimensional generic sigmoidal chaotic map with application of true random bit generation," *Entropy*, vol. 20, no. 2, p. 136, 2018.

[29] P. Ayubi, S. Setayeshi, and A.M. Rahmani, "Chaotic complex hashing: A simple chaotic keyed hash function based on complex quadratic map," *Chaos, Solitons & Fractals*, vol. 173, pp. 113647, 2023.

[30] N. Jiteurtragool, et al., ''A topologically simple keyed hash function based on circular chaotic sinusoidal map network,'' in *15th International Conference on Advanced Communications Technology* (ICACT), pp. 1089–1094, 2013.

[31] H Liu, et al., "Keyed hash function using hyper chaotic system with time-varying parameters perturbation," *IEEE Access*, vol. 7, pp.37211-37219, 2019.

[32] A. Mostafa, S. J. Lee, and Y. K. Peker, "Physical unclonable function and hashing are all you need to mutually authenticate IoT devices," *Sensors*, vol. 20, no. 16, p. 4361, 2020.

[33] A. Braeken, "PUF based authentication protocol for IoT," *Symmetry*, vol. 10, no. 8, p. 352, 2018.

[34] Espressif Systems, "ESP32 Technical Reference Manual," 2023.

[35] N. Jiteurtragool, "Generalized parabola chaotic map for pseudorandom random number generator," in *26th International Conference on Advanced Communications Technology (ICACT)*, pp. 53-56, , 2024.