A New Approach to Device Identification Using ICMetric and Chaotic Maps

Zaid Al-Khazaali, Taha A. Nuhad, and Anwar Sabah

***Abstract*—The rapid expansion of the Internet has increased security risks for connected devices. Traditional security systems rely on stored templates or cryptographic keys; if these are breached, the entire system becomes vulnerable. In addition, electronic systems have become increasingly vulnerable to identity theft, spoofing, and impersonation, compromising the system's overall integrity. To address and improve the security of the mentioned issues, various techniques and approaches have been explored, one of which is ICMetrics. This approach presents a more secure alternative by generating cryptographic keys from unique hardware or software behaviors, thereby eliminating the need to store sensitive data. However, ICMetrics can face limitations when device behaviors are too similar, reducing key uniqueness. To overcome this, the paper introduces the use of a logistic map chaotic system to enhance the randomness and strength of ICMetric-generated keys. Due to its sensitivity to initial conditions, the logistic map produces highly unpredictable sequences suitable for cryptography. Entropy analysis of keys derived from different decimal positions of the chaotic signal shows consistently high randomness, especially from the second digit onward. Experiments across key lengths from 128 to 2048 bits demonstrate near-maximal entropy values (~8 bits/byte) and improved key generation efficiency. The new approach, ICMetric-Chaotic is based on the ICM-RSA framework and achieves faster key generation without compromising security, offering a scalable and reliable method for enhancing ICMetric-based cryptographic systems, particularly for securing embedded devices, and gives a solution for the ICMetrics limitation.**

***Index terms*—ICMetric, Security, Chaotic maps, Logistic Map, Cryptographic Key Generation.**

I. INTRODUCTION

While the expansion of the Internet has introduced substantial convenience, it has also drawn the attention of hackers, scammers, and other malicious actors seeking to exploit its vulnerabilities. In the context of software, if malware injects harmful code into an application, it can be challenging for users to detect. If applications fail to properly encrypt or adequately secure data, user information may be exposed to interception by unauthorized parties [1]. In the context of hardware, although embedded devices are utilized across a diverse range of application domains, they typically execute a limited set of repetitive tasks or operate within a constrained state space. Researchers have investigated both hardware and software-based approaches to enhance the security of these devices. Among the hardware-based solutions, the Physical Unclonable Function (PUF), also referred to as hardware-intrinsic security, has been proposed as a promising mechanism to provide physical security for embedded systems [2].

The first issue in conventional security mechanisms is that they rely on storing sensitive information, such as valid samples or templates, within the system. Regardless of the approach used, if the storage or encryption is compromised, it can lead to full exposure of all protected data, highlighting a critical vulnerability in these techniques [2], [3].

The aforementioned concerns can be partially mitigated; however, enhancing security requires meeting certain critical conditions. Specifically, identifiers or signatures must possess inherent resistance to cloning, and the system should avoid storing any templates or reference samples [1].

To resolve the first security challenge, the proposed solution involves the use of the ICMetrics (Integrated Circuit Metrics) technique. This approach generates encrypted signatures based on the inherent behavioral characteristics of the software or hardware components, enabling verification of an object's authenticity and ensuring it is functioning as intended [1]. An ICMetric is associated with the cryptographic keys of a device, which implies that there is no need to store the keys on the system [4]. The ICMetrics technology operates based on a two-phase process [5], [6], [7], [8]:

Calibration Phase (Used only once within each application domain):

1. Measure the relevant feature values for each device.
2. Generate feature distributions to represent the frequency of discrete values observed per device.
3. Normalize the feature distributions and produce corresponding normalization maps.

Operation Phase (Executed whenever an encryption key is required for a specific circuit):

1. Measure the required feature values during operation.
2. Apply the normalization maps to convert the measured values into a format suitable for key generation.
3. Execute the key generation algorithm using the normalized values.

Manuscript received June 18, 2025; revised September 24, 2025. Date of publication December 18, 2025. Date of current version December 18, 2025.

The associate editor prof. Toni Perković has been coordinating the review of this manuscript and approved it for publication.

Z. Al-Khazaali and T. A. Nuhad are with the Mustansiriyah University, Iraq (e-mails: {zaid_alkhazaali, tahalzzw}@uomustansiriyah.edu.iq).

A. Sabah is with the University of Technology, Iraq (e-mail: anwar.s.sabty@uotechnology.edu.iq).

Digital Object Identifier (DOI): 10.24138/jcomss-2025-0074

ICMetrics solves issues such as system integrity and identity spoofing by generating a unique ID. Also, it eliminates the need to store sensitive data within the system. However, another issue arises: it becomes hard to distinguish one device from another if their feature values are too similar (overlap). When that happens, more complex or additional techniques are needed to improve accuracy [9]. Also, ICMetric systems that rely on simple features could be susceptible to effective analysis attacks [10].

To overcome the second issue that arises from using ICMetrics, a chaotic map is used and will be explained in the next section (Methodology). The chaotic map depicts a nonlinear function that is defined by its sensitivity to initial conditions, with small differences in the initial value leading to significantly divergent results. This characteristic allows chaotic systems to produce complex and unpredictable sequences, and thus, these systems are most appropriately used for purposes requiring randomness and security. With simply a small adjustment to the initial value, an infinite number of different, non-periodic sequences can be generated, which have high randomness and unpredictability. Such characteristics make chaotic maps of high relevance to fields related to cryptography, image encryption, and data hiding. Various types of chaotic maps have been studied and used to date, and these include the Tent map [11], Henon map [12], with each map having specific characteristics based on the application at hand. Under the conditions of this investigation, the logistic map is chosen as a result of its simple dynamics and efficient chaotic behavior. It should be noted, however, that a number of different maps, methods, and studies have employed advanced or optimized methods which could outstrip the performance of the logistic map under certain circumstances [13], [14].

The logistic map, identified as the simplest and well-studied nonlinear system, is a prototype example of a discrete chaotic map. It first appeared as a statistical model in [11].

The main contributions of this work can be summarized as follows:

- The proposed method increases the uniqueness and differentiation among the generated ICMetric values (by using the chaotic logistic map).
- The system achieves better identification accuracy without requiring significantly more complex or computationally expensive processes.
- The proposed method presents a new technique for deriving cryptographic keys from chaotic signals with high entropy.
- The results demonstrated that the proposed method is practically applicable in real-world scenarios due to its efficient processing time and acceptable entropy levels.

The paper is organized as follows. Section I describes using ICMetric and chaotic maps to create secure, unique device signatures without storing sensitive data. Related works are discussed in Section II. Section III (methodology) describes the proposed approach for generating secure device signatures and enhancing system accuracy. Section IV demonstrates the effectiveness of the proposed system and presents the results of processing time and achieved entropy for the generated key. Section V presents the conclusion, a discussion about the results, and suggests potential directions for future work.

II. Literature Review

Recent studies have explored various approaches to enhance the reliability and uniqueness of ICMetric-based systems for secure key generation. Researchers have investigated both hardware and software-level features to overcome challenges in entropy, stability, and device distinction. These efforts aim to improve the effectiveness of ICMetric in cryptographic applications and secure environments.

In [15], Tahir R. et al. presented a strong ICMetric-based key generation protocol capable of producing keys with entropy levels close to 8 bits per byte and of the required length. The protocol was designed to resist brute-force attacks through a two-tier architecture that prevents adversaries from accessing the original ICMetric. The proposed method successfully generated high-entropy public/private key pairs (within RSA), which is critical for enhancing the security of cryptographic applications, as such keys are significantly more resilient to exhaustive search attacks.

In [16], Kovalchuk Y. et al. examined the feasibility of using the Program Counter (PC) as an ICMetric feature for encryption key generation in electronic systems. The researchers evaluated how the number of PC samples and the tracing methods—single stepping and sampling—affected the system's ability to uniquely identify devices. Their findings demonstrated that the Program Counter could serve as a reliable ICMetric feature; however, the effectiveness of device identification was influenced by the number of samples collected and the method used, both of which impacted the consistency and quality of the generated encryption keys.

Ye et al. in [17] explored the use of software behaviors as ICMetric features for encryption key generation in cloud environments. They analyzed method invocations, loop iterations, and if statements to assess their effectiveness in uniquely identifying servers. Experimental results on three servers showed that method invocations and loops had strong classification potential. However, if statements displayed overlapping distributions, reducing their reliability for identification. Feature correlation was assessed using covariance matrices to evaluate its impact on system stability. The study confirmed that behavioral features could be viable for ICMetric but highlighted the need for more robust features. Limitations included a small-scale test environment and a narrow feature set.

III. Methodology

The proposed system aims to generate a strong ICMetric key that can be utilized for encryption and digital signature purposes. The overall structure of the system is inspired by the method proposed in the research paper [15]. However, the key generation process will be conducted using a new approach. This key will then be used to derive both the private and public keys necessary for cryptographic operations.

The new method suggests integrating and utilizing the properties of chaotic maps in the key generation process to achieve high randomness and strong security characteristics. Figure 1 illustrates the stages upon which the system is built for the key generation process:

In the first stage, both hardware and software features can be utilized as input parameters to organize and prepare the initial value required for the chaotic map. In this paper, the logistic map is selected for this purpose. Various parameters can be employed, such as the device's Serial Number (SN), Universal Product Code (UPC), software version, and other attributes of the device itself or its internal components that contribute to its unique identification.

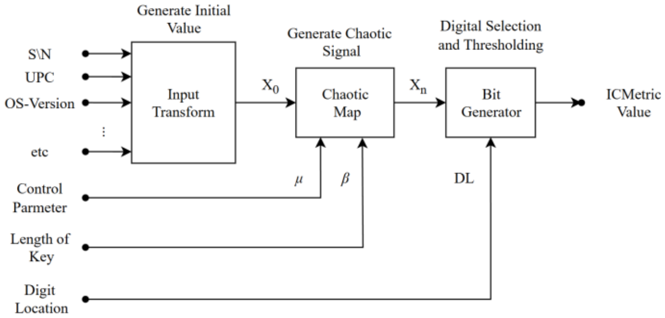


Fig. 1. Flowchart of the proposed system

These selected parameters are concatenated to construct a unique object that defines the device's identity. This object is then hashed and converted from hexadecimal to decimal format. The resulting number represents a unique identifier for the device, serving as the initial input for the chaotic system, and it will be as shown in equation (1).

$$X_o = S/N\ ||UPC\ ||\ OS - Version\ ||\ ...\ ||\ Feature_N\ . \quad (1)$$

In the second stage (Chaotic Map), the chaotic signal is generated using the logistic equation. Mathematically, the logistic map can be represented as follows:

$$x_{n+1} = \mu x_n(1 - x_n), \quad (2)$$

where $\mu \in [0,4]$ is the control parameter that governs the system's behavior, and $x_n \in [0,1]$ represents the state variable at iteration n. When the value of μ lies within range $3.57 \leq \mu \leq 4$, the system enters a chaotic regime, producing non-periodic and highly sensitive sequences, but in the other range, the system converges to a fixed point or a periodic cycle [18], [19], [20].

As illustrated in the equation, the control parameter can be predefined and stored within the device. However, the initial value must comply with the logistic map's requirement that x_0 lies between 0 and 1. Therefore, the unique number produced in the input transform stage (stage 1) is used by placing it directly after the decimal point, forming a value in the format (0.X) where x represents the generated number from stage 1. This ensures that the initial condition is suitable for generating the chaotic sequence.

The chaotic signal generated in this work is constructed from 2048 numbers, where each one is a decimal value between 0 and 1, typically containing up to 15 digits after the decimal point. In Stage 3 (Bit Generator), a specific digit position after the decimal point is selected for each generated chaotic number. The choice of digit position significantly affects the randomness of the generated bits; for instance, selecting the first digit after the decimal point would reduce randomness, as will be demonstrated in the results section later. Figure 2 shows the chaotic signal with its values. Once the digit is selected, it is passed through a thresholding process. If the digit's value exceeds a predefined threshold (set to 5 in this work), it is converted to a binary '1'; otherwise, it is converted to a binary '0'. By repeating this process, the ICMetric value is ultimately generated, consisting of 2048 bits. The thresholding process plays a vital role in converting chaotic real sequences into binary sequences. Many techniques can be used for these purposes like [21], [22], [23] and many other techniques, but the authors in this work use the method as described above. For example, consider the value at time t=0, which is equal to 0.0346557. In this work, the third digit after the decimal point is adopted and passed through the thresholding process, since the selected digit is 4, and 4<5, the resulting bit is 0. This bit represents the first of the 2048 bits that form the ICMetric value. At this stage, the ICMetric value (represented in binary form) can be converted into a specific integer.

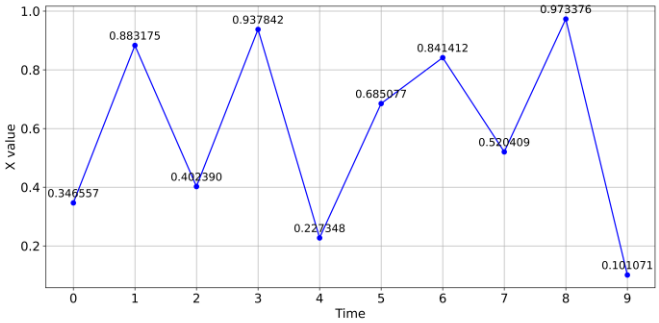


Fig. 2. Chaotic signal with its values

This resulting value serves as a unique device ID or can be further utilized to generate a key pair consisting of a private and a public key, which are essential components in digital signatures and asymmetric cryptographic systems. In this work, the ICM-RSA algorithm that is proposed in [15] is adopted and evaluated for the key generation process. It shows a strong performance, indicating that the approach can be practically applied with high effectiveness. This algorithm uses RSA along with a generated ICMetric value and a well-known Procedure to generate the RSA key pair [24], [25], [26], [27], [28]:

- Randomly pick two unique large prime numbers p & q $(p \neq q)$
- Calculate n = p × q
- Calculate $\phi_n = (p - 1)(q - 1)$
- Choose an integer e such that gcd (8(n), e) = 1;
- 1 < e < 8(n)
- Find d that fulfills the congruence condition
- d = e−1 mod 8(n); d is kept as private
- Public Key = {e, n}
- Private Key = {d, n}

But in the ICM-RSA framework [15] the key generation process begins by using the strong ICMetric value that is generated as the private key d′. Next, two large prime numbers p and q are selected such that d′ is co-prime with ϕ(n), where n=p×q. If d′ is not co-prime with ϕ(n), an offset is applied to modify d′, resulting in a new private key d = f (d′, offset) so that d′ is derived from ICMetric, and the function f could be an XOR or addition operation (in this work, the authors used the XOR operation). This adjustment ensures that the greatest common divisor gcd (d, ϕ(n)) =1. Finally, the public exponent e is computed as the modular multiplicative inverse of d (mod ϕ(n)). i.e., $e=d^{-1}$ mod ϕ(n). $d', d, p, q, \varphi(n)$ are discarded. While f and the offset are retained locally with the μ, DL, and β parameters, while the pair (n, e) is the public key. The proposed framework, as mentioned in previous studies that used ICMetric, explores this technology from two perspectives: first, as a means to prevent key theft, and second, as a foundation for generating cryptographic keys.

IV. Results

The entropy is measured to evaluate the strength of the generated key. The entropy may vary depending on whether the first, second, third, etc., number in the chaotic signal is selected. Therefore, all possible positions within the signal are tested, and the corresponding entropy values are calculated to assess the key strength and the effect of the number or digit position that is used to pass through the threshold process to construct the binary form of the key. The time required for key generation is measured, and it is demonstrated that the approach is both practical and applicable. Figure 3 shows the entropy (in bits per byte) of keys generated using a logistic chaotic map. Specifically, the key generation process involved extracting only the first digit after the decimal point from each value in the chaotic signal. The location of the digit is constant, but the key length varies (based on the chosen beta) as shown.

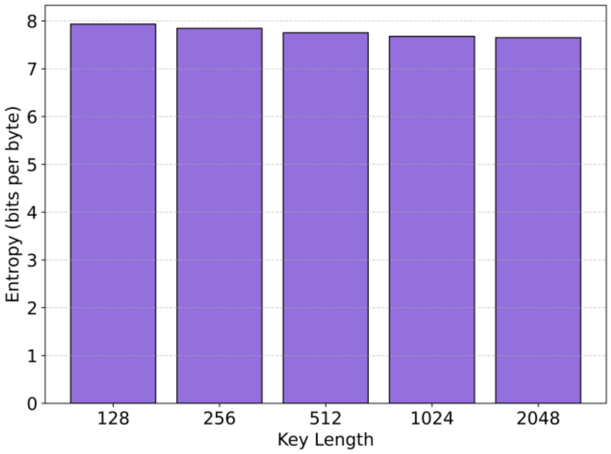


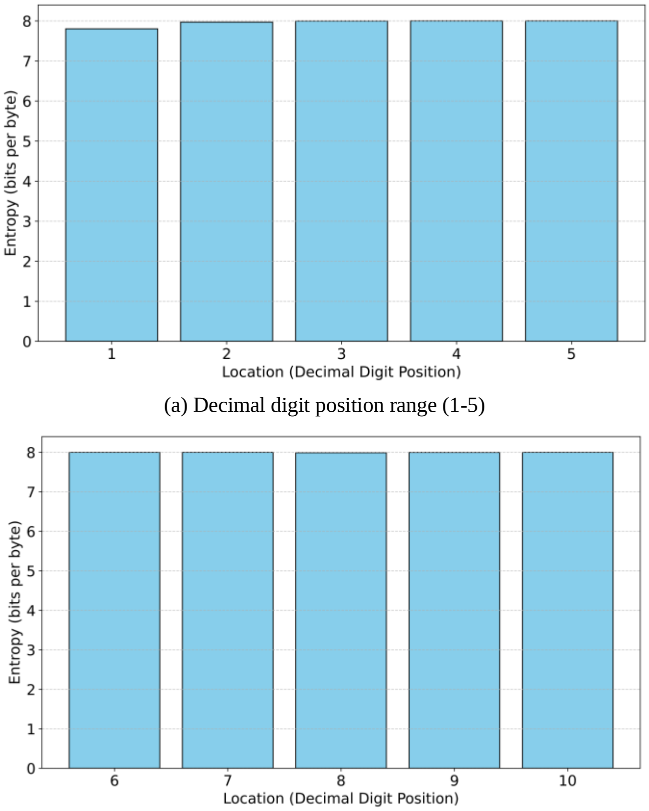
Fig. 3. Entropy at the first digit with different key lengths

TABLE I
Entropy at The First Digit With Different Key Lengths

Key Length	Entropy
128	7.9308
256	7.8440
512	7.7509
1024	7.6746
2048	7.6469

TABLE II
Entropy of Keys Generated From Different Decimal Digit Positions

Location	Entropy
1	7.8023
2	7.9714
3	7.9988
4	7.9994
5	7.9993
6	7.9999
7	7.9998
8	7.9903
9	7.9980
10	7.9997



(a) Decimal digit position range (1-5)

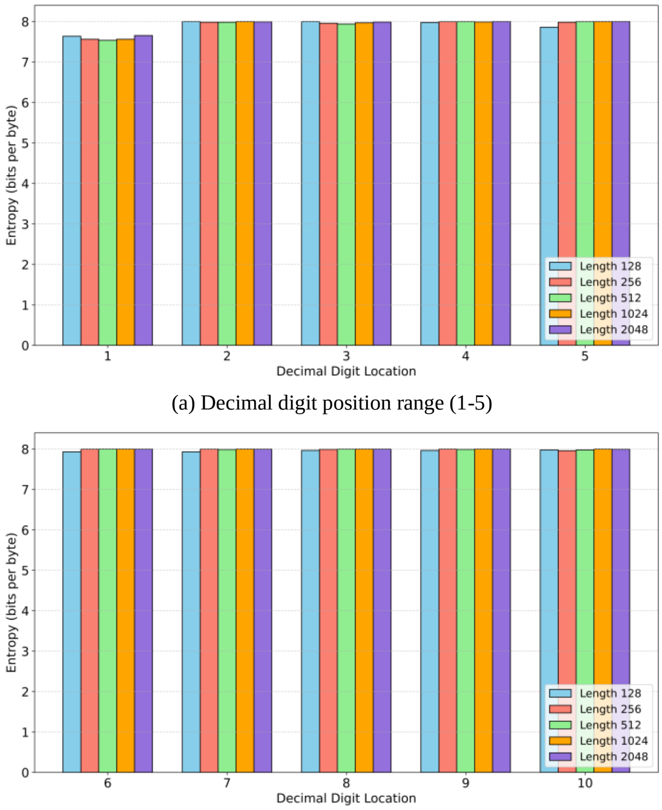
(b) Decimal digit position range (6-10)

Fig. 4. Entropy of keys generated from different decimal digit positions.

Each bar corresponds to a different key length and shows the entropy of the resulting key, and the entropy values across all key lengths are consistently high, ranging between approximately 7.65 and 7.8 bits/byte, which is close to the maximum value of 8 bits/byte.

Figure 4 shows that the randomness of the key increases significantly after the first decimal digit. The first digit is slightly less random, which is expected because the early digits in logistic chaotic sequences may show some deterministic patterns due to limited precision or initialization artifacts. Starting from the second digit onward, the entropy is consistently very high (close to the ideal 8 bits/byte), indicating excellent randomness. So, selecting digits from later positions (2nd to 10th) in the chaotic number is more effective for generating cryptographically strong keys.

Figure 5 presents a comparative entropy analysis of cryptographic keys generated from different decimal digit positions (1 to 10) of the logistic chaotic signal, across multiple key lengths (128, 256, 512, 1024, and 2048 bits). Each bar group represents a specific digit position extracted from the chaotic signal, while the color-coded bars within each group correspond to different key lengths.



(a) Decimal digit position range (1-5)

(b) Decimal digit position range (6-10)

Fig. 5. Entropy analysis of keys generated across decimal digit positions

TABLE III
ENTROPY ANALYSIS OF KEYS GENERATED ACROSS DECIMAL DIGIT POSITIONS

		Length				
		128	256	512	1024	2048
Position	1	7.6355	7.563	7.5373	7.563	7.6553
	2	7.9986	7.9827	7.9827	7.9989	7.9916
	3	7.9986	7.9573	7.9404	7.9698	7.9889
	4	7.9774	7.9968	7.9978	7.9893	7.9991
	5	7.8585	7.9827	7.9978	7.9992	7.9996
	6	7.9308	7.9968	7.9999	7.9996	7.9976
	7	7.9308	7.9968	7.9893	7.9992	7.9968
	8	7.9647	7.9912	7.9986	8	7.9989
	9	7.9647	7.9996	7.9912	7.9996	7.9988
	10	7.9774	7.9573	7.9774	8	7.996

The entropy is a key indicator of randomness and cryptographic strength. The results show that for all key lengths, the entropy increases with digit position, reaching nearly maximum entropy (~8 bits/byte) starting from position 2 onward. Keys generated from the first decimal digit exhibit slightly lower entropy, particularly for shorter key lengths like 128 bits, indicating weaker randomness. Beyond the second digit, entropy values stabilize and approach the ideal maximum regardless of key length, suggesting consistent and high randomness in deeper decimal places of the chaotic sequence. This confirms that both the choice of digit position and key length influence the entropy, with middle-to-late digit positions offering optimal randomness even at shorter key lengths.

Figure 6 shows the comparison between the proposed method and the ICM-RSA framework method in terms of processing time. ICM-RSA proposed that a hash algorithm run 100,000 iterations to generate a strong ICMetric key. The proposed approach shows significant reductions in the processing time, highlighting how computationally efficient it is. This implies that the chaotic method has the potential to be used as an efficient method with ICMetric to generate a strong ICMetric key.

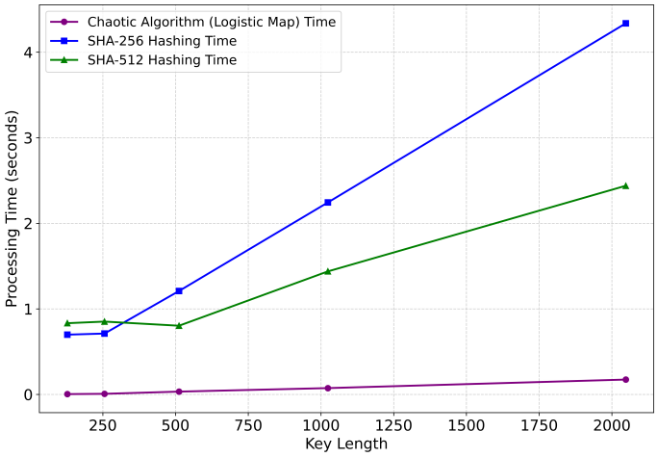


Fig. 6. Processing Time Comparison

The PC's CPU that used to run these algorithms and record the time is AMD Ryzen 9 4900HS, and the code is written in Python.

TABLE IV
PROCESSING TIME COMPARISON (TIME IN SECONDS)

Key Length	Chaotic Algorithm	SHA-256	SHA-512
128	0.0041	0.6989	0.8328
256	0.0074	0.7118	0.8516
512	0.0332	1.2089	0.8031
1024	0.0743	2.2441	1.4382
2048	0.1740	4.3368	2.4388

V. CONCLUSION, DISCUSSION, AND FUTURE WORK

This paper improves the ICMetric framework by integrating chaotic dynamics, specifically the logistic map, to increase the variability and uniqueness of the identifiers generated by the device. The proposed method demonstrates that small chaotic systems can reach near maximal entropy.

The proposed approach enhances the ICMetric framework; it inherently depends on the characteristics of the logistic map. Consequently, the security and performance of the generated key rely on the properties of this chaotic system and the chosen parameters. The logistic map is characterized by entropy and randomness, and even small deviations and badly chosen control parameters lead to poorly defined entropy, degradation of randomness, or even periodic behaviors. Such dependencies pose potential drawbacks for the ICMetric key. Therefore, careful parameter tuning and validation are essential to ensure that the generated keys consistently achieve the desired level of unpredictability and security.

The proposed method introduces a novel approach to derive cryptographic keys from chaotic signals with the promise of high entropy and strong cryptographic security characteristics. It also improves identification accuracy without introducing significant computational overhead. Experimental results confirm the practicality of the approach, demonstrating efficient processing time alongside acceptable entropy levels, which makes the method suitable for real-world deployment in resource-constrained environments. The approach also enabled the generation of a large number of unique identifiers, exceeding the conventional limits of ICMetric-based identification. The generated ICMetric is successfully integrated into encryption systems, and the method proved to be effective in securing communications by using it along with a specific Asymmetric algorithm to generate private and public keys.

For future work, many chaotic maps can be tested with our system to increase the complexity and robustness, for example, the Henon map, Tent map, or Lorenz system. Cascade or hybrid approaches, whereby multiple chaotic maps are linked such that only a single ICMetric value results, could bring higher unpredictability as well as cryptanalytic attack resistance.

References

[1] M. Haciosman, B. Ye, and G. Howells, "Protecting and Identifiying Smartphone Apps Using Icmetrics," in *2014 Fifth International Conference on Emerging Security Technologies*, IEEE, Sep. 2014, pp. 94–98. doi: 10.1109/EST.2014.28.

[2] X. Zhai *et al.*, "Exploring ICMetrics to detect abnormal program behaviour on embedded devices," *Journal of Systems Architecture*, vol. 61, no. 10, pp. 567–575, Nov. 2015, doi: 10.1016/j.sysarc.2015.07.007.

[3] A. B. T. Hopkins, K. D. McDonald-Maier, E. Papoutsis, and W. G. J. Howells, "Ensuring data integrity via ICmetrics based security infrastructure," in *Second NASA/ESA Conference on Adaptive Hardware and Systems (AHS 2007)*, IEEE, Aug. 2007, pp. 75–81. doi: 10.1109/AHS.2007.48.

[4] R. Tahir, H. Tahir, K. McDonald-Maier, and A. Fernando, "A novel ICMetric based framework for securing the Internet of Things," in *2016 IEEE International Conference on Consumer Electronics (ICCE)*, IEEE, Jan. 2016, pp. 469–470. doi: 10.1109/ICCE.2016.7430694.

[5] R. Tahir, Huosheng Hu, Dongbing Gu, K. McDonald-Maier, and G. Howells, "Resilience against brute force and rainbow table attacks using strong ICMetrics session key pairs," in *2013 1st International Conference on Communications, Signal Processing, and their Applications (ICCSPA)*, IEEE, Feb. 2013, pp. 1–6. doi: 10.1109/ICCSPA.2013.6487307.

[6] S. Yadav and G. Howells, "Secure Device Identification Using Multidimensional Mapping," in *2019 Eighth International Conference on Emerging Security Technologies (EST)*, IEEE, Jul. 2019, pp. 1–5. doi: 10.1109/EST.2019.8806218.

[7] Y. Kovalchuk, K. McDonald-Maier, and G. Howells, "Overview of ICmetrics Technology – Security Infrastructure for Autonomous and Intelligent Healthcare System," *International Journal of u-and e-Service, Science and Technology*, vol. 4, Sep. 2011.

[8] K. M. A. Alheeti, F. Khaled Alarfaj, M. Alreshoodi, N. Almusallam, and D. Al Dosary, "A hybrid security system for drones based on ICMetric technology," *PLoS One*, vol. 18, no. 3, p. e0282567, Mar. 2023, doi: 10.1371/journal.pone.0282567.

[9] B. Ye, "Analysis of ICmetrics features requirements in Cloud environment," 2014. [Online]. Available: https://api.semanticscholar.org/CorpusID:211164998

[10] S. J. Alsunaidi and A. M. Almuhaideb, "Investigation of the optimal method for generating and verifying the Smartphone's fingerprint: A review," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 5, pp. 1919–1932, May 2022, doi: 10.1016/j.jksuci.2020.06.007.

[11] R. M. May, "Simple mathematical models with very complicated dynamics," *Nature*, vol. 261, no. 5560, pp. 459–467, Jun. 1976, doi: 10.1038/261459a0.

[12] M. Hénon, "A two-dimensional mapping with a strange attractor," *Commun Math Phys*, vol. 50, no. 1, pp. 69–77, Feb. 1976, doi: 10.1007/BF01608556.

[13] Y. Zhou, L. Bao, and C. L. P. Chen, "A new 1D chaotic system for image encryption," *Signal Processing*, vol. 97, pp. 172–182, Apr. 2014, doi: 10.1016/j.sigpro.2013.10.034.

[14] K. C. Murthy*, Mahalinga. V. Mandi, and R. Murali, "Chaotic Binary Sequence Generator based on Logistic Map," *International Journal of Recent Technology and Engineering (IJRTE)*, vol. 8, no. 4, pp. 7351–7355, Nov. 2019, doi: 10.35940/ijrte.D5297.118419.

[15] R. Tahir, S. Tahir, H. Tahir, K. McDonald-Maier, G. Howells, and A. Sajjad, "A novel ICMetric public key framework for secure communication," *Journal of Network and Computer Applications*, vol. 195, p. 103235, Dec. 2021, doi: 10.1016/j.jnca.2021.103235.

[16] Y. Kovalchuk *et al.*, "Investigation of Properties of ICmetrics Features," in *2012 Third International Conference on Emerging Security Technologies*, IEEE, Sep. 2012, pp. 115–120. doi: 10.1109/EST.2012.22.

[17] B. Ye, G. Howells, and M. Haciosman, "Investigation of Properties of ICmetric in Cloud," in *2013 Fourth International Conference on Emerging Security Technologies*, IEEE, Sep. 2013, pp. 107–108. doi: 10.1109/EST.2013.36.

[18] M. Alawida, "Enhancing logistic chaotic map for improved cryptographic security in random number generation," *Journal of Information Security and Applications*, vol. 80, p. 103685, Feb. 2024, doi: 10.1016/j.jisa.2023.103685.

[19] S. E. Borujeni and M. S. Ehsani, "Modified Logistic Maps for Cryptographic Application," *Appl Math (Irvine)*, vol. 06, no. 05, pp. 773–782, 2015, doi: 10.4236/am.2015.65073.

[20] Z. Alqadi, "Analysis of Chaotic Logistic Map used to Generate Secret Keys," *International Journal of Computer Science and Mobile Computing*, vol. 13, no. 4, pp. 25–40, Apr. 2024, doi: 10.47760/ijcsmc.2024.v13i04.004.

[21] P. G S and R. S, "Pseudo Random Binary Sequences Obtained Using Novel Chaos Based Key Stream Generator and their Auto-correlation Properties," *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 11, no. 5, pp. 141–147, May 2023, doi: 10.17762/ijritcc.v11i5.6588.

[22] R. Naik, A. Pal, C. Kataria, T. Gosavi, and V. Rathish, "A Pseudo Random Binary Sequence Generator Based on Chaotic Logistic Maps," *SSRN Electronic Journal*, 2021, doi: 10.2139/ssrn.3865940.

[23] S. S. and S. S. V., "Generation of chaotic random binary sequences for cryptographic applications," *Concurr Comput*, vol. 34, no. 1, Jan. 2022, doi: 10.1002/cpe.6497.

[24] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Commun ACM*, vol. 21, no. 2, pp. 120–126, Feb. 1978, doi: 10.1145/359340.359342.

[25] O. F. A. Wahab, A. A. M. Khalaf, A. I. Hussein, and H. F. A. Hamed, "Hiding Data Using Efficient Combination of RSA Cryptography, and Compression Steganography Techniques," *IEEE Access*, vol. 9, pp. 31805–31815, 2021, doi: 10.1109/ACCESS.2021.3060317.

[26] P. Cotan and G. Teşeleanu, "Small Private Key Attack Against a Family of RSA-Like Cryptosystems," 2024, pp. 57–72. doi: 10.1007/978-3-031-47748-5_4.

[27] M. Rahmani, A. Nitaj, and M. Ziane, "Improved Cryptanalysis of Some RSA Variants," *Algorithms*, vol. 18, no. 4, p. 223, Apr. 2025, doi: 10.3390/a18040223.

[28] Q. Chang, T. Ma, and W. Yang, "Low power IoT device communication through hybrid AES-RSA encryption in MRA mode," *Sci Rep*, vol. 15, no. 1, p. 14485, Apr. 2025, doi: 10.1038/s41598-025-98905-0.

Zaid H. Al-Khazaali received his B.Sc. in Computer Communication Engineering from Al-Rafidain University College in 2015 and his M.Sc. in Electrical Engineering (Electronics and Communications) from Mustansiriyah University in 2020. From 2018 to 2023, he gained professional experience in networking and holds certifications, including CCNP ENCOR, CCNP Advanced Routing and Enterprise, and CEH Master from EC-Council. Since April 2023, he has been an Assistant Lecturer at the College of Engineering, Mustansiriyah University, Baghdad, Iraq, where he also works in the Department of Construction and Projects. His current research interests include network security, intrusion detection systems, deep learning-based traffic classification, and cryptography. He is an active participant in cybersecurity events and has authored several research papers in his field.

Taha A. Nuhad received his B.Sc. in Electrical Engineering from Mustansiriyah University in 2020 and his M.Sc. in Electrical Engineering (Electronics and Communications) from Mustansiriyah University in 2023. Since 2023, he has been an Assistant Lecturer at Mustansiriyah University, Baghdad, Iraq, where he also works in the Department of Construction and Projects. His current research interests include chaotic systems, network security, anti-jamming systems, and wireless communication.

Anwar Sabah received her B.Sc. in Electrical Engineering from Mustansiriyah University College in 2015 and her M.Sc. in Electrical Engineering (Electronics and Communications) from Mustansiriyah University in 2020. From 2020 to 2023, she worked in the Network and Network technologies. Since April 2023, she has been an Assistant Lecturer at the College of Engineering, University of Technology, Baghdad, Iraq, where she also works in the Department of Laser and Optoelectronics Engineering.