Fast Dynamic Device Authentication Based on Lorenz Chaotic Systems

Lake Bu[†], Hai Cheng[§]*, Michel A. Kinsy[†], *Member, IEEE*
[§]Department of Electronic Engineering, Heilongjiang University, Harbin, China
[†]Adaptive and Secure Computing Systems Laboratory, Boston University, Boston, USA

***Abstract*—Chaotic systems, such as Lorenz systems or logistic functions, are known for their rapid divergence property. Even the smallest change in the initial condition will lead to vastly different outputs. This property renders the short-term behavior, i.e., output values, of these systems very hard to predict. Because of this divergence feature, lorenz systems are often used in cryptographic applications, particularly in key agreement protocols and encryptions. Yet, these chaotic systems do exhibit long-term deterministic behaviors - i.e., fit into a known shape over time. In this work, we propose a fast dynamic device authentication scheme that leverages both the divergence and convergence features of the Lorenz systems. In the scheme, a device proves its legitimacy by showing authentication tags belonging to a predetermined trajectory of a given Lorenz chaotic system. The security of the proposed technique resides in the fact that the short-range function output values are hard for an attacker to predict, but easy for a verifier to validate because the function is deterministic. In addition, in a multi-verifier scenario such as a mobile phone switching among base stations, the device does not have to re-initiate a separate authentication procedure each time. Instead, it just needs to prove the consistency of its chaotic behavior in an iterative manner, making the procedure very efficient in terms of execution time and computing resources.**

Index terms — *Lorenz System, Authentication, PUF*.

I. Introduction

Chaotic systems (maps) are usually characterized by their high input sensitivity. The smallest alterations or variations in the initial condition will result in significant changes on the output values. Therefore, collecting a large set of system outputs may not – and in general does not – lead to a good prediction of the chaotic system's output values. For such a map, the system parameters are central to the system chaotic properties. The parameters need to be selected in a manner which ensures that the system divergence is highly sensitive to changes in the input values.

There are many types of chaotic systems. In this work, we focus on the 3D Lorenz systems. The original Lorenz system was introduced to describe and model fluid or air applications that are uniformly heated from bottom and cooled from above [1]. However, due to its divergence property, it has also been used as a cryptographic primitive or component. Its most common application is in encryption. Researchers have suggested using the chaotic functions in block ciphers [2], as well as image encryption [3]. Unlike conventional secret key-based encryptions, which use one key to obfuscate the entire piece of data, chaotic systems are able to spawn a new key or random vector for each block, which can be precisely re-generated during decryption. This block-based obfuscation technique only requires one set of pre-shared system parameters. Another common use is to facilitate key agreements [4]. Other applications include image digest algorithms [5] and random number generators [6].

However, two issues are often overlooked when developing or implementing chaotic systems like Lorenz:

- *Convergence*: Besides their divergence property, Lorenz systems do have a convergence property as well. Although, this convergence property is rarely used. With a given set of system parameters, all the output values are centralized around two attractors on the function trajectory. Although the outputs of the Lorenz system are still highly unpredictable with varying initial conditions, the trajectory is determined statistically according to the system parameters.
- *Static System Parameters*: Unlike keyed cryptographic schemes where the keys can be any vectors, the system parameters of chaotic functions cannot be arbitrarily chosen. They must be selected in a way that ensures both convergence and divergence. Thus, they cannot be arbitrarily updated as the public key systems.

To address these two issues, we propose a novel authentication scheme based on Lorenz systems. The major contributions of the paper are:

1. We leverage both the convergence and divergence properties of the Lorenz systems in the authentication procedure. The convergence property is used to provide a fast but rough verification of the function output values. The divergence of the function is used for slower but more accurate authentication. We combine the two properties to implement an adaptive verification scheme and improve the efficiency of the procedure;
2. To enhance the security of the Lorenz system-based scheme, we outline an approach for dynamically updating the system parameters while still maintaining all the chaotic nature of the system;
3. The proposed scheme works specially well for use-cases when a device needs to be authenticated frequently by multiple verifiers. An illustrative scenario is shown in Fig. 1. The device only needs to prove the consistency of its chaotic behavior in an iterative manner, instead of re-initiating a new authentication session each time.

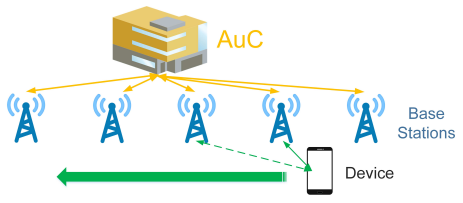


Fig. 1: While a phone is switching among the base stations (or a vehicle being authenticated by other automobiles in the network), it needs to be authenticated at every transition. All base stations rely on an authentication center (AuC) for this process, which shares the same secret with the device.

* Hai Cheng participated in this research while he was a visiting scholar with the Adaptive and Secure Computing Systems Laboratory.

978-1-5386-8398-9/18/$31.00 ©2018 IEEE

The rest of the paper is organized as follows: Section II presents the relevant background on Lorenz chaotic functions. Section III introduces the hardware primitive required by the scheme. Section IV outlines the authentication protocol.

II. Preliminaries of the Lorenz Chaotic Systems

In this section we will introduce the concept and property of the Lorenz chaotic systems. To better facilitate the presentation, we define the following notations:

- σ, β, ρ: the system parameters of Lorenz functions;
- $p_n = (x_n, y_n, z_n)$: an output of a Lorenz system after n iterations, which is a point on the trajectory with coordinates (x_n, y_n, z_n);
- $LF_{\sigma,\beta,\rho}(p_0, n)$: a Lorenz function with system parameters σ, β, ρ. Where p_0 is the initial condition and p_n the outputs after n iterations.

A. Lorenz Chaotic Systems

There exist many types of chaotic systems with different dimensions. For example, there are one-dimensional (1D) logistic map, two-dimensional (2D) Van der Pol system, and three-dimensional (3D) Chua circuit. In this work, we focus on the Lorenz system, which is a 3D chaotic map. The Lorenz functions can be formulated by a system of differential equations with three parameters as shown in the equation below:

$$\begin{cases} \dfrac{\mathrm{d}x}{\mathrm{d}t} = \sigma(y - x) \\ \dfrac{\mathrm{d}y}{\mathrm{d}t} = x(\rho - z) - y \\ \dfrac{\mathrm{d}z}{\mathrm{d}t} = xy - \beta z \end{cases} \tag{1}$$

where σ, β, and ρ are the system parameters. The Lorenz system was introduced originally to model the consequent bidirectional convection of air or fluid. And the three system parameters were defined as follows:

- $\sigma = 10$ as the Prandtl number, to denote the ratio of thermal conductivity and viscosity of the studied material;
- $\rho = 28$ as the Rayleigh number, to represent the difference between the system's top and bottom temperatures;
- $\beta = 2.6667$ as the ratio of the area's width and height in which the air or fluid convection is formed.

Fig 2 shows the chaotic map of the Lorenz function with the above parameters. Since then Lorenz systems have been applied in many fields such as dynamos, lasers, chemical reactions, and cryptography, these parameters can take on other values as long as the convergence and divergence are maintained.

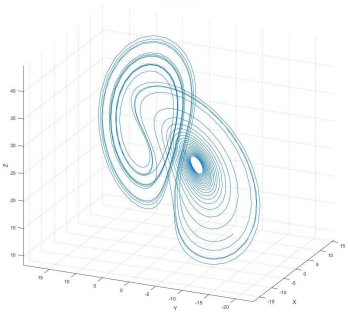


Fig. 2: The trajectory of a 3D Lorenz system, which is usually in a pattern of a butterfly or "8".

For digital applications of Lorenz functions, the discrete equations are given below:

$$\begin{aligned} x_{n+1} &= x_n + \sigma(x_n - y_n)\triangle t \\ y_{n+1} &= y_n + (\rho x_n - x_n z_n - y_n)\triangle t \\ z_{n+1} &= z_n + (x_n y_n - \beta z_n)\triangle t \end{aligned} \tag{2}$$

where $\triangle t$ determines the resolution of the map. In the proposed design and its FPGA implementation, we adopt the fixed-point multiplication version of [Eq. 2].

B. The Key Properties of Lorenz Chaotic Systems

1) *Stationary points*: In [Eq. 2], when $\rho > 1$, there are two distinct stationary points:

$$C1, C2 = (\pm\sqrt{\beta(\rho-1)}, \pm\sqrt{\beta(\rho-1)}, \rho - 1) \tag{3}$$

These two points are located at mirror symmetry with respect to the vertical plane $x + y = 0$. Although $C1$ and $C2$ are not physically on the trajectory, they serve as the *attractors* that balance out and kill off the initial transients, and evolve the system towards its typical behavior. It should be highlighted that σ does not determine the space location of the two attractors, but rather the size of the map.

2) *Convergence*: The attractors bring in the convergence property of a chaotic system. In other words, even if the initial state p_0 is not a point on the trajectory, it will converge to the orbit in a finite number of iterations. In addition, although the coordinates of individual Lorenz system outputs are or seem highly unpredictable in the short-range, over time they all conform to the butterfly pattern.
Another straightforward representation of the convergence property is the z-axis ($x = y = 0$). All trajectories which start on the z-axis, will remain on it and tend to evolve towards the origin $(0, 0, 0)$ in a clockwise direction.
The convergence attribution can be described by Hausdorff dimension $dim_H K$ [7] bounded by:

$$dim_H K \leq 3 - \frac{2(\sigma + \beta + 1)}{\sigma + 1 + \sqrt{(\sigma-1)^2 + 4\rho\sigma}} \tag{4}$$

3) *Divergence*: The divergence property has been adopted in many cryptographic applications because of its high unpredictability. With a tiny variation of the initial condition p_0, the output p_n will largely fluctuate. Lyapunov exponent can be used to evaluate the divergence of a given chaotic system:

$$|\delta(p)| \approx |\delta(0)|e^{\lambda p}, \tag{5}$$

where for a trajectory $T(p)$'s nearby orbit $T(p) + \delta(p)$, $\delta(p)$ is a vector with infinitesimal initial length. The maximal λ is known to be approximately 0.9056 [8].

III. Lorenz System-Based Authentication Primitive (LAP)

Before introducing the proposed fast authentication technique, we will first present its core building block, termed the Lorenz System-Based Authentication Primitive (LAP). The LAP primitive (cf., Fig. 3) anchors the security of the scheme in the uniqueness of the device hardware. It is used to dynamically update the system parameters and generate unique tokens for authentication.

A. The Three Basic Units of a LAP

As shown in Fig. 3, there are three major components in a LAP: U1, U2, and an input control unit (ICU). U1 consists of a physical unclonable function (PUF) and a Lorenz

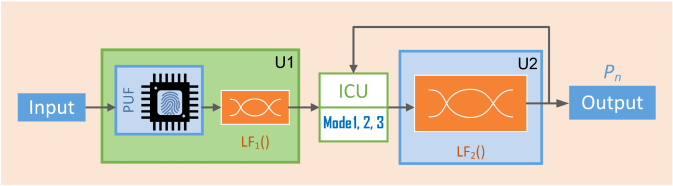


Fig. 3: Here we use $LF_1() = LF_{\sigma_1,\beta_1,\rho_1}()$, $LF_2() = LF_{\sigma_2,\beta_2,\rho_2}()$ for simplicity. The PUF box contains a PUF sizing at least 16-bit. Larger sized PUFs can be added to increase the system variation.

function module $LF_1()$. The purpose of U1 is to establish the security of the LAP based on the device's hardware uniqueness. Particularly, the PUF reflects the LAP's intrinsic randomness and $LF_1()$ serves as the randomness magnifier.

A PUF [9] is a piece of hardware that produces unpredictable responses upon challenges due to their manufacturing variations. Each PUF's output (response) is a non-linear function of the outside stimulation (challenge) and the PUF's own physical, intrinsic, and unique diversity. Even under exactly the same circuit layout and manufacturing procedure, two pieces of hardware will still have distinct behaviors.

Ideally, the "raw" responses of a PUF should be sufficient to demonstrate its distinction from others. However, in real implementations the responses need to be further randomized due to the limited uniqueness of the PUF. In earlier designs, one-way hash functions are used for the randomization process. Recently, because of their lower implementation cost, chaotic systems have been proposed as an efficient substitute for one-way hash functions [10], [11]. This is the design logic behind the U1 unit in the LAP. The U2 Lorenz system is in charge of generating the final authentication tokens sent to the verifier. It is the core unit in the LAP design. The authentication procedure has three phases or modes. The mode switches in the LAP is managed by the input control unit (ICU).

B. Mode 1: Dynamic Parameter Configuration

First, the challenge and response pairs (CRPs) of the PUF, as well as the system parameters $(\sigma_1, \beta_1, \rho_1)$ of $LF_1()$, are pre-stored at the verifier's end. While U1 is only the magnifier of the PUF, the actual system parameters of the LAP are then U2's $(\sigma_2, \beta_2, \rho_2)$.

Before a device is deployed for use, $(\sigma_2, \beta_2, \rho_2)$ will be dynamically configured. These three parameters are all 64-bit vectors, where the first 8 bits represent the integer digits, and the last 56 bits are decimal. As mentioned in Section I, $(\sigma_2, \beta_2, \rho_2)$ cannot be arbitrarily chosen since they need to provide to the Lorenz system an appropriate level of the chaotic behavior [12]. However, through an extensive experimental process, we have found that there does exist a certain (but limited) degree of freedom in the parameter configuration as shown in Fig. 4.

Fig. 5 shows three examples when only the 48 LSBs of (ρ, β, σ) are modified respectively. New Lorenz chaotic maps are generated, while they still preserve both their convergence and divergence properties.

Protocol III.1. CHL_i denotes the i^{th} challenge to the PUF in the LAP. RSP_i is the corresponding response to CHL_i. The iteration number of $LF_1()$ is fixed at m. The dynamic configuration of the 48 LSBs of $(\sigma_2, \beta_2, \rho_2)$ is as follows:

1) The verifier acquires k pairs of CRPs from the PUF (Fig. 3). Before the device is released, the 16 MSBs of $(\sigma_2, \beta_2, \rho_2)$ are fixed and stored as (Σ, B, P) at both the verifier and device ends. $(\Sigma||\Theta, B||\Theta, P||\Theta)$ are tested and proved to be a set of proper chaotic parameters for Lorenz systems, where Θ stands for a 48-bit vector of all 0's and $||$ the concatenation operator;
2) After selecting the last 48 bits of $(\sigma_2, \beta_2, \rho_2)$, the verifier selects three arbitrary responses RSP_{i_0}, RSP_{i_1}, RSP_{i_2} from the PUF and computes:
$$\begin{aligned} \sigma_d &= LF_1(RSP_{i_0}, m) \\ \beta_d &= LF_1(RSP_{i_1}, m) \\ \rho_d &= LF_1(RSP_{i_2}, m) \end{aligned} \tag{6}$$
3) For this round, the 3-tuple $(\sigma_2, \beta_2, \rho_2)$ is calculated as:
$$\sigma_2 = (\Sigma||\sigma_d), \quad \beta_2 = (B||\beta_d), \quad \rho_2 = (P||\rho_d). \tag{7}$$

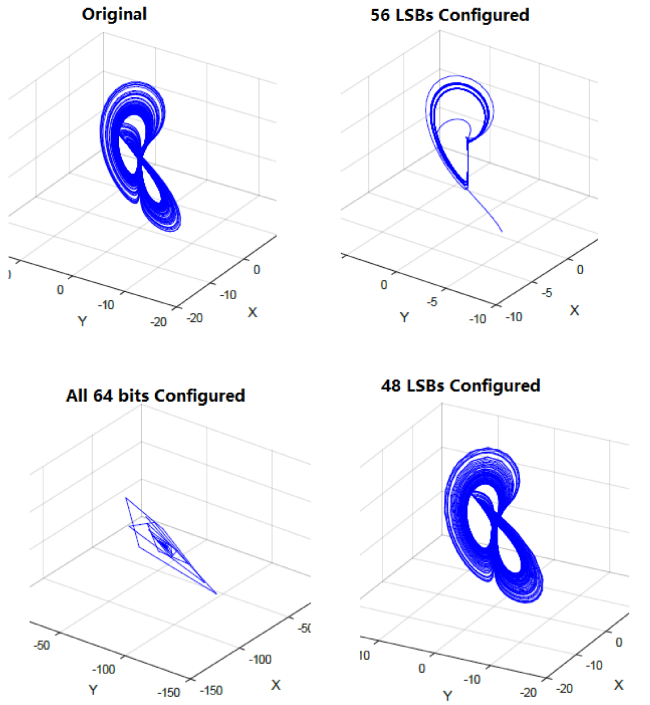
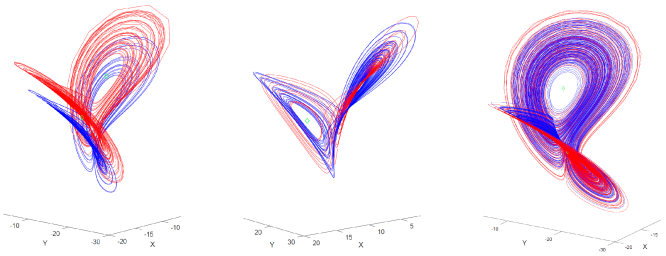


Fig. 4: For a given Lorenz map (upper left) with carefully-selected three 64-bit system parameters, if all the 64-bits are arbitrarily modified, then the new pattern will easily end up as a non-chaotic trajectory (bottom left). Similar situation happens when the 56 least significant bits (LSBs) are arbitrarily modified (upper right). Only when the configuration is restricted to 48 LSBs (bottom right) or lower, can the new generated Lorenz map remain chaotic.



(a) ρ (b) β (c) σ

Fig. 5: As theoretically analyzed in [Eq. 3], the modification of ρ and β will lead to the re-location of the attractors. As shown in (a) and (b), the old (blue) and new (red) Lorenz maps do not share the same $C1, C2$. On the other hand as presented in (c), modifying σ does not change the coordinates of the attractors, but the areas of the old and new butterfly maps are different.

4) To configure a device with the selected system parameters, the verifier sends the challenges CHL_{i_0}, CHL_{i_1}, CHL_{i_2} to the device's LAP.
5) The ICU switches from idle to Mode 1 and accepts the three outputs from the U1 unit to set $(\sigma_2, \beta_2, \rho_2)$ locally in U2 using [Eq. 7].

It should be emphasized that the above procedure is secure against eavesdropping since CHL_i leaks no information of RSP_i. Furthermore, the U1 PUF could be made larger to increase the system's unpredictability as shown in Fig. 6.

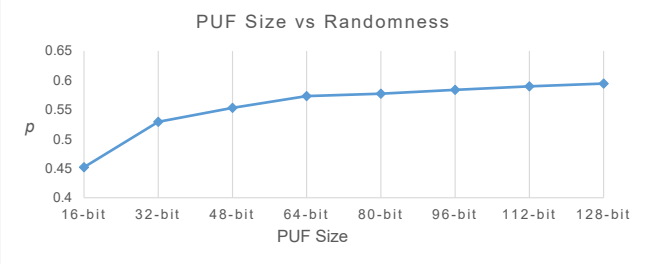


Fig. 6: As the size of the PUF increases from 16 to 128 bits, the variation of $LF_1()$'s inputs also enlarges. This leads to the increasing randomness of U1's output. The randomness is evaluated by the p-value of the National Institute of Standards and Technology (NIST) SP 800-22 test [13].

C. Mode 2: Trajectory Landing

When the ICU is in this mode, the goal is to produce the first point p_1 that lands on the trajectory. The verifier sends two random numbers CHL_{R_1} and CHL_{R_2} to the device as the request. The PUF uses the request, i.e., challenge, to generate two outputs:

$$\begin{aligned} p_0 &= LF_1(RSP_{R_1}, m) \\ n &= LF_1(RSP_{R_2}, m) \end{aligned} \tag{8}$$

The (p_0, n) is then fed into $LF_2()$ as the initial condition by the ICU. The U2 unit can generate its first output:

$$p_1 = LF_2(p_0, [n]), \tag{9}$$

where $[n]$ is defined as:

$$[n] = n \bmod (v - u) + u, \tag{10}$$

and u and v are the lower and upper bounds of n.

The derivation of p_1 in U2 using [Eq. 9] has to ensure that p_1 lands on the trajectory featured by $(\sigma_2, \beta_2, \rho_2)$. Therefore, the U2 operation may take multiple iterations to meet this convergence requirement. The lower bound u of iterations is determined by the resolution variable $\triangle t$ in [Eq. 2]. Essentially, it regulates how fast an arbitrary point can be attracted to the trajectory, as shown in Fig 7.

In this work, we use a high resolution $\triangle t = 0.00001$, which has the lower bound of $u = 220$ iterations for the landing. On the other hand, v limits the maximum number of iterations to save computation resources and time. Here we choose the bounds to be $220 \leq [n] \leq 250$.

D. Mode 3: Authentication Token Generation

Once U2 in the LAP outputs the first point p_1 on the orbit, the ICU switches the initial condition of U2 from p_0 to p_j, where $j > 0$. Thus the LAP's outputs will be:

$$p_j = LF_2(p_{j-1}, [n]), \tag{11}$$

where the set $\{p_j\}$ will be used for fast authentication among multiple verifiers.

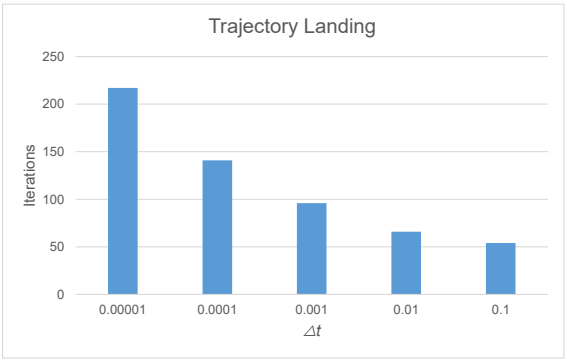


Fig. 7: The smaller $\triangle t$ is, the more iterations are needed for an arbitrary point to enter into the trajectory.

IV. Fast Dynamic Authentication Based on Lorenz Chaotic Systems

First, we briefly outline the protocol, then we describe in the subsections in more detail each of the steps. The advantages of the proposed technique are:

1) Unlike conventional authentication schemes that require at least two messages to be communicated between every verifier and device, the proposed scheme uses two messages for the first verifier, and only one message for each of the consequent verifications;
2) The technique rests on the fact that it is hard for an adversary to predict the correct responses to a challenge because to the Lorenz function's divergence and the dynamic configuration of the system parameters, yet, it is easy for a verifier to authenticate because of the function does converge under the right criteria;
3) The proposed protocol only involves fixed-point multiplication and addition (rather than exponential operations in conventional approaches). Moreover, the authentication is carried out in an adaptive manner for better efficiency.

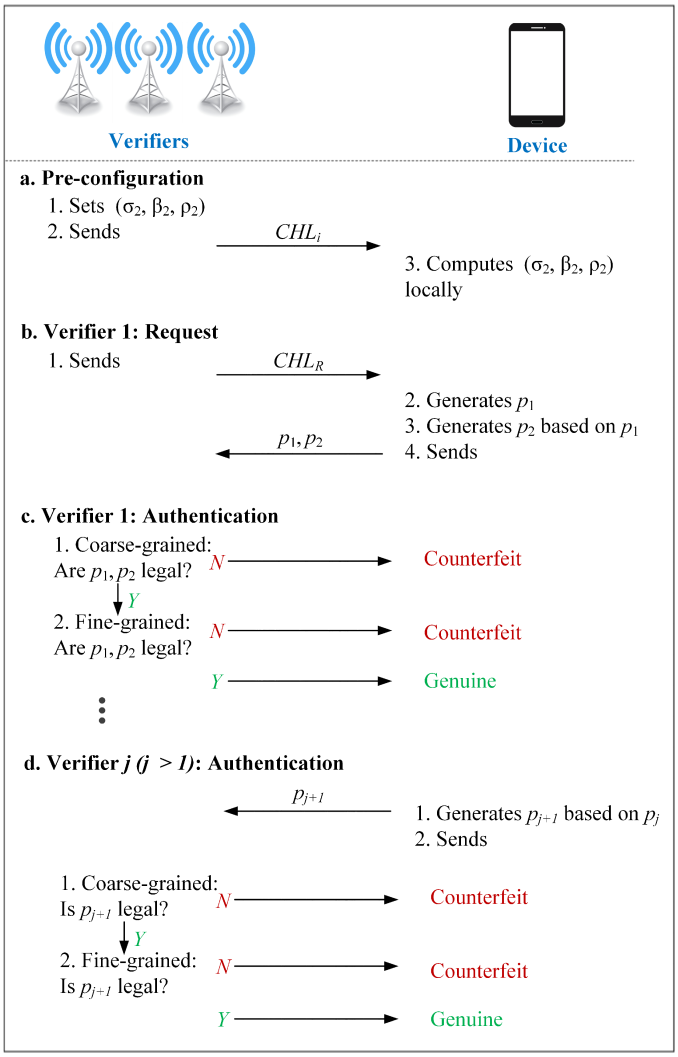
The adaptive authentication protocol using the Lorenz chaotic system is outlined in Protocol IV.1.

In the pre-configuration step, the verifier determines the Lorenz system parameters of the device to be authenticated. It then notifies the device in an eavesdrop-resistant manner using the PUF's CRP feature. The device switches to *Mode 1* to dynamically configure its $(\sigma_2, \beta_2, \rho_2)$ accordingly. It is worth noting that the pre-configuration does not have to be run every time an authentication procedure is initiated. Once a set of system parameters is determined, the PAL can generate $2^{48 \cdot 3}$ unique authentication tokens under this set, which is sufficient for most applications. The system parameters only need to be re-set in the event of an information leakage, in other words, when the verifier or the device suspects that the current $(\sigma_2, \beta_2, \rho_2)$ 3-tuple may have been compromised.

When the first verifier tries to authenticate the device, it sends two arbitrary CHL_{R_1} and CHL_{R_2} to the device. The device transitions into *Mode 2* to generate an on-trajectory point p_1, and then shifts to *Mode 3* to generate p_2 based on p_1. At that moment, the two points are sent to the verifier for their fast authentication.

When the device encounters another verifier, let us say verifier j, it only needs to generate one point p_{j+1} for authentication, and does not need to wait for the verifier's request. p_{j+1} can be authenticated based on the p_j sent to the previous verifier $(j-1)$. All verifiers share this correlated

Protocol IV.1.



information and carry out the authentication by communicating with the authentication center (AuC), as shown in Fig. 1.

We have presented the details of steps a and b of the protocol IV.1 in Section III when we introduced the LAP and its three modes. Steps c and d deal with the adaptive nature of the authentication and are described in the following subsection.

A. Adaptive Authentication

According to the convergence property mentioned in Section II-A, although the outputs of a Lorenz system are highly unpredictable with a small variation in the initial condition, the long-term statistic behavior is always a trajectory in butterfly patterns. Representationally, there exist two stationary points (attractors) C1 and C2 that center all the points located on the two curved surfaces (the two wings of the "butterfly") of the function (c.f. [Eq. 3]). From the formulation [Eq. 2], one can observe that starting from a certain point on the trajectory, the output of the equation always ends up with another point on the trajectory as well.

Therefore, we take advantage of this convergence property to propose a fast authentication protocol. This protocol works in an adaptive manner. It first verifies that p_j roughly belongs to the space covered by the curved surface. This is a fast but coarse-grained testing of the fitness of p_j. Next, it passes it through a fine-grained verification step, which is to check that p_{j+1} can indeed be computed by the Lorenz functions [Eq. 2] using p_j as the initial state. This step is an accurate authentication. If either verification step fails, the device is identified as a counterfeit. Due to the lack of knowledge of the system parameters, most counterfeits will be spotted quickly at the coarse-grained check. Even if the adversaries are able to make a very close estimation on $(\sigma_2, \beta_2, \rho_2)$ to bypass the coarse-grained verification, they will be detected at the fine-grained stage.

1) Coarse-grained Verification: For a Lorenz system, once a set of system parameters is given, the related trajectory is confined within a certain space. The space occupied by the two curved surfaces can be approximated by two cylinders as shown in Fig. 8. The dimensions of the two cylinders can be conveniently estimated by taking a relatively large $\triangle t$ in [Eq. 2], say $\triangle t = 0.1$ (cf., Fig. 9). The approximate range of any in-orbit point's coordinates (x, y, z) can be calculated this way. The range is denoted as $\{[\bar{x}, \underline{x}], [\bar{y}, \underline{y}], [\bar{z}, \underline{z}]\}$ where $\bar{}$ is the maximum value of the coordinate and $\underline{}$ the minimum. The following equation estimates the fitness or authenticity of a point:

$$Authentic = (x \in [\bar{x}, \underline{x}] \ \& \ y \in [\bar{y}, \underline{y}] \ \& \ z \in [\bar{z}, \underline{z}])?1 : 0 \quad (12)$$

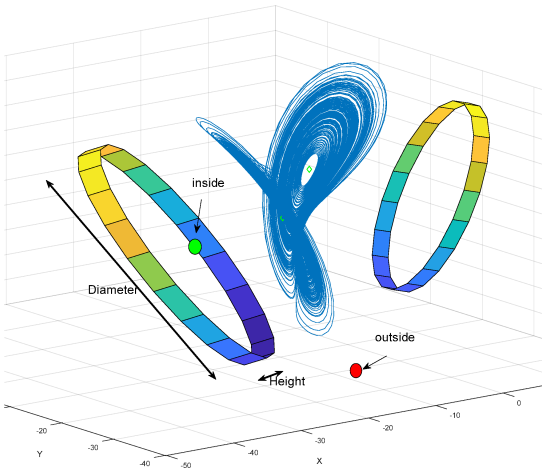


Fig. 8: The two cylinders are an approximation of the space taken by the two curved surfaces of the Lorenz system's trajectory. Given their diameters and heights, one can use [Eq. 12] to quickly evaluate if a given point is within this space or not.

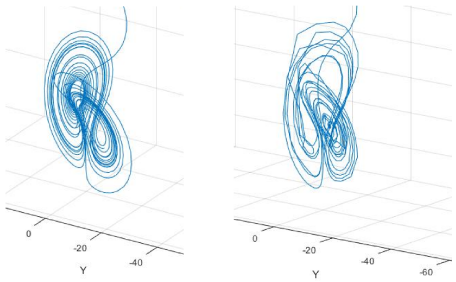


Fig. 9: By adopting a a relatively large $\triangle t$, a low-resolution frame of the chaotic map can be quickly sketched. (left) is a Lorenz map drawn with 10^6 points under $\triangle t = 0.00001$, and (right) is its approximation with 100 points under $\triangle t = 0.1$.

2) Fine-grained Authentication: Like other security cruxes, there have been efforts to reverse engineer the system parameters [14], [15] with the observed outputs. In the proposed

protocol, if an adversary can have good enough estimates of $(\sigma_2, \beta_2, \rho_2)$, they can generate a similar butterfly pattern, which can produce p_j and bypass the coarse-grained verification step. Therefore, we propose the fine-grained verification stage to check if:

$$p_j \stackrel{?}{=} LF_2(p_{j-1}, [n]). \tag{13}$$

[Eq. 13] is derived from [Eq. 2] under the principle that once a point is on the trajectory, if it is used as the initial condition, all subsequent points will remain on the trajectory regardless of the number of iterations. This is an accurate and robust verification mechanism that can be derived from the convergence property of the Lorenz systems. Unless an adversary acquires the exact system parameters $(\sigma_2, \beta_2, \rho_2)$, it is nearly impossible to forge a p_j satisfying [Eq. 13].

Fig. 10 is an example depicting the case where a device, using p_{j-1} (green), submits p_j (red) to a verifier $(j-1)$. However, after $[n]$ iterations in [Eq. 13] with p_{j-1} being the initial condition, the verifier reaches the blue point $p'_j \neq p_j$, indicating that the device is a counterfeit.

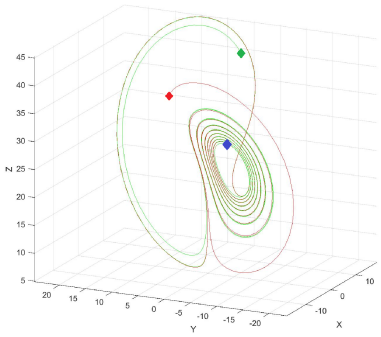


Fig. 10: The visual representation of [Eq. 13]'s verification. Without the accurate knowledge of the dynamic $(\sigma_2, \beta_2, \rho_2)$, it is infeasible for an adversary to generate a legitimate p_j for verifier $(j-1)$.

B. Evaluation

We evaluate the hardware and timing costs of the proposed scheme by comparing to the conventional secret key signature based authentication, which requires exponential operations in large finite fields. Theses comparisons are done using the Xilinx Vertex 7 XC7VX330T FPGA board.

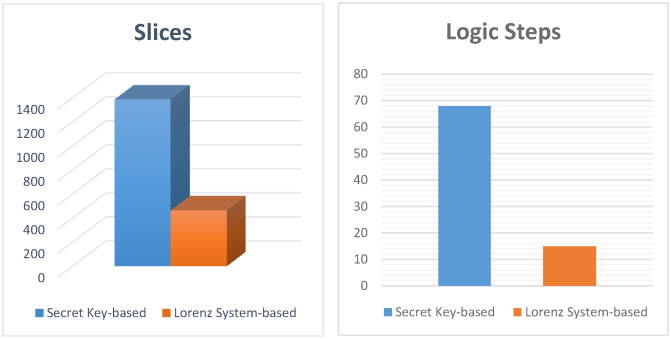


Fig. 11: The proposed authentication scheme saves 67.1% of the slices used on FPGA, and 77.9% on the timing cost over the conventional approach. It is both fast and resource-efficient.

V. Conclusion

In this paper, we introduce our work on using Lorenz chaotic functions for fast dynamic device authentication. The major contributions of this scheme are 1) it is able to achieve a time and resource-saving authentication procedure using Lorenz systems' properties; 2) unlike conventional chaotic system applications with static system parameters, the proposed scheme configures these secrets in a dynamic manner to enhance the security of the scheme; 3) it is able to accomplish authentication with one message rather than a two step request-response procedure, making the scheme perfect for frequent device authentication by multiple verifiers.

Acknowledgment

This research is partially supported by the NSF grant (No. CNS-1745808).

References

[1] F. C. Moon, *Chaotic and fractal dynamics: introduction for applied scientists and engineers*. John Wiley & Sons, 2008.

[2] G. Jakimoski and L. Kocarev, "Chaos and cryptography: block encryption ciphers based on chaotic maps," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 2001.

[3] H. Kwok and W. K. Tang, "A fast image encryption system based on chaotic maps with finite precision representation," *Chaos, solitons & fractals*, 2007.

[4] C. Guo and C.-C. Chang, "Chaotic maps-based password-authenticated key agreement using smart cards," *Communications in Nonlinear Science and Numerical Simulation*, 2013.

[5] H. Yang, K.-W. Wong, X. Liao, W. Zhang, and P. Wei, "A fast image encryption and authentication scheme based on chaotic maps," *Communications in Nonlinear Science and Numerical Simulation*, 2010.

[6] M. François, T. Grosges, D. Barchiesi, and R. Erra, "Pseudo-random number generator based on mixing of three chaotic maps," *Communications in Nonlinear Science and Numerical Simulation*, vol. 19, no. 4, pp. 887–895, 2014.

[7] A. Y. Pogromsky, G. Santoboni, and H. Nijmeijer, "An ultimate bound on the trajectories of the lorenz system and its applications," *Nonlinearity*, 2003.

[8] D. Viswanath, "Lyapunov exponents from random fibonacci sequences to the lorenz equations."

[9] B. Gassend *et al.*, "Silicon physical random functions," *Proceedings of the Computer and Communications Security Conference*, 2002.

[10] L. Lin, H. Huang, and S. H, "Lorenz chaotic system based carbon nanotube physical unclonable functions," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2017.

[11] L. Chen, "A framework to enhance security of physically unclonable functions using chaotic circuits," *Physics Letters*, 2018.

[12] L. Bu, H. Cheng, and M. A. Kinsy, "Adaptive and dynamic device authentication based on lorenz chaotic systems," 2018.

[13] S. NIST, "800-22," *A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications*, vol. 120, 2000.

[14] P. Akritas, I. Antoniou, and V. Ivanov, "Identification and prediction of discrete chaotic maps applying a chebyshev neural network," *Chaos, Solitons & Fractals*, vol. 11, no. 1-3, pp. 337–344, 2000.

[15] H. Abarbanel, *Analysis of observed chaotic data*. Springer Science & Business Media, 2012.