EasyChair Preprint

№ 13647

# A HMAC Scheme Based on 6D Hyper Chaotic Maps for Enhanced Security

Omessead Benmbarak

EasyChair preprints are intended for rapid dissemination of research results and are integrated with the rest of EasyChair.

June 12, 2024

# A HMAC Scheme Based on 6D Hyper chaotic maps for Enhanced Security


***Abstract*—The HMAC (Keyed-Hash Message Authentication Code) algorithm serves as a cornerstone of security widely employed to ensure data authentication and integrity within information systems and computer networks. Essentially straightforward, HMAC relies on hash functions utilizing a secret key. The cryptographic strength of HMAC derives from its adept utilization of efficient cryptographic characteristics like balancing and the avalanche effect. In our research, we introduce a fresh HMAC algorithm incorporating hyperchaotic systems to bolster its cryptographic attributes. Our 6D-HMAC showcases resilience against MAC collision threats, displays heightened sensitivity to key, plaintext, and error propagation influences, and presents enhanced security in contrast to traditional approaches.**



***Index Terms*—MAC, Hash functions; 6D Dimensions Maps; Hyperchaotic System**


## I. Introduction

Ensuring the authenticity of data and guarding against alterations are crucial aspects of information security. Message authentication codes (MACs) play a pivotal role in achieving these objectives. Among MAC variants, HMAC [1] (Hash-based Message Authentication Code) stands out for its effectiveness due to several reasons, as noted :

- Unlike MACs employing encryption algorithms, HMAC operates with hash functions, which, while slower, offer robust security.
- Many hardware cryptographic tools are optimized for handling large data volumes, making HMAC a preferred choice.
- Licensing requirements are often associated with other cryptographic algorithms, whereas HMAC is freely available.

HMAC [2], a cornerstone of cryptographic message authentication, plays a crucial role in verifying the authenticity and data integrity of messages. Widely used in data exchange and storage applications, it ensures the legitimacy of the data source. As a specialized Message Authentication Code (MAC), HMAC utilizes a shared secret key and a hash function to generate a message digest (tag). This tag assures that the message originated from a trusted source and has not been tampered with during transmission or storage. This paper introduces a novel Message Authentication Code (MAC) proposal tailored for hyperchaotic systems. In essence, it represents an advancement over its predecessor HMAC, where the Mask, Encryptor, and compression components of this new MAC are grounded on the most chaotic 6D system.

The article is organized as follows: Section 2 provides an overview of 6-dimensional hyperchaotic systems; Section 3 presents the proposed 6D-HMAC model; Section 4 outlines the analysis conducted to assess the effectiveness of 6D-HMAC. Finally, Section 6 offers the conclusion of the article.

## II. Six dimensional (6D) hyperchaotic system

Mathematical analyses have highlighted the nonlinear nature and dynamic complexities inherent in commonly used chaotic functions, posing challenges in predicting their responses. Studies indicate that hyperchaotic functions exhibit significantly more intricate dynamic behaviors compared to their low-dimensional counterparts [30]. Chaotic systems are characterized by a minimum requirement of four dimensions, while low-dimensional chaotic functions feature a single positive Lyapunov exponent, contrasting with high-dimensional functions which typically possess at least two such exponents. Authors [3] offered the following definition for a 6-dimensional hyperchaotic system:

$$\begin{cases} x_1 &= a(x_2 - x_1) + x_4 - x_5 - x_6 \\ x_2 &= cx_1 - x_2 - x_1x_3 \\ x_3 &= -bx_3 + x_1x_2 \\ x_4 &= dx_4 - x_2x_3 \\ x_5 &= ex_6 + x_3x_2 \\ x_6 &= rx_1 \end{cases} \tag{1}$$

where a, b, c, d, e, and r are constants and $x_1$, $x_2$, $x_3$, $x_4$, $x_5$, and $x_6$ are the state variables of the 6D hyperchaotic system. In this study, a = 10, b = $\frac{8}{3}$, c = 28, d = -1, e = 10, and r = 3 were selected as constants. This ensures that the system has two positive Lyapunov exponents [4] that satisfy the condition (the sum of all exponents is negative).

Figure 1 illustrates the corresponding two-dimensional ($x_1$, $x_2$), ($x_1$, $x_3$), ($x_2$, $x_3$), and ($x_1$, $x_4$) phase portraits of the system.
The Lyapunov exponents are displayed in Table I.

TABLE I
VALUES OF LYAPUNOV EXPONENTS

| | |
|---|---|
| $l_{LE1}$ | 0.362485 |
| $l_{LE2}$ | 0.24709 |
| $l_{LE3}$ | 0 |
| $l_{LE4}$ | -0.225698 |
| $l_{LE5}$ | -10.0017 |
| $l_{LE6}$ | -15.0708 |

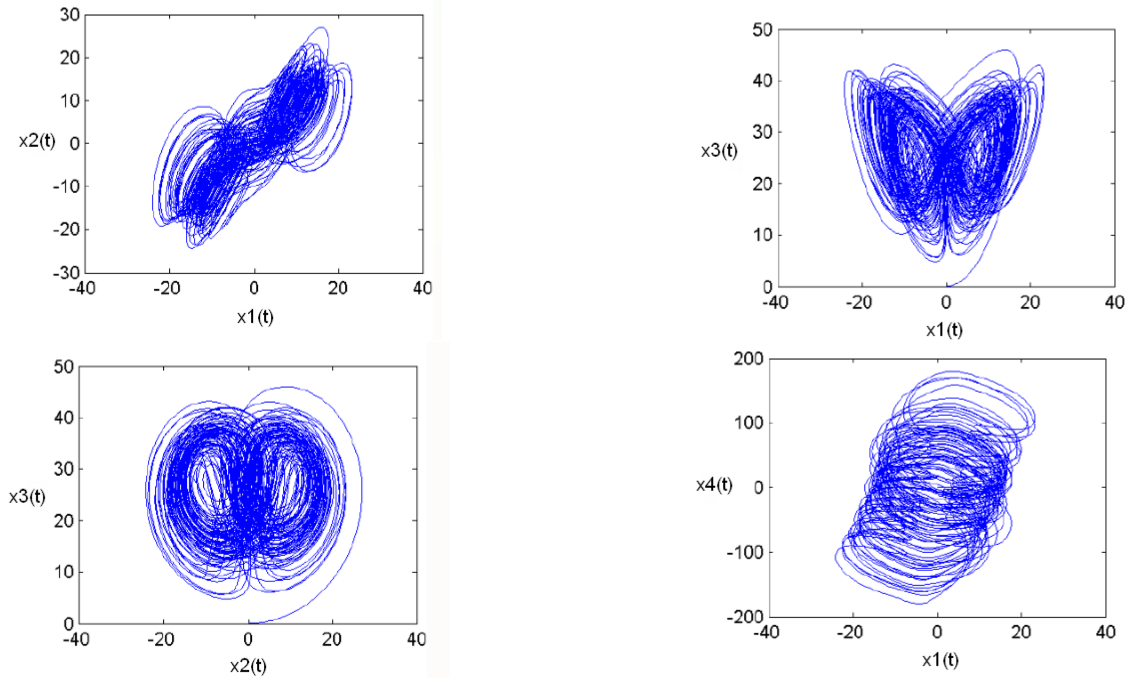


Fig. 1. Hyperchaotic attractor of system 1 in 2-D spaces with (a, b, c, d, e, r) = (10,$\frac{8}{3}$, 28,-1, 10, 3) and initial condition ($x_{1_0}$, $x_{2_0}$, $x_{3_0}$, $x_{4_0}$, $x_{5_0}$, $x_{6_0}$ ) = (0.1, 0.1, 0.1, 0.1, 0.1, 0.1).

## III. PROPOSED 6D-HMAC

Network users must verify that messages have not been altered, accidentally or maliciously. The recommended method to protect against undetected alterations is to develop a key-based message authentication code (HMAC). The flowchart, in the graph 2, effectively captures the essential steps in generating an HMAC, highlighting the importance of message padding, key preparation, and a hash function to ensure message authenticity and integrity.

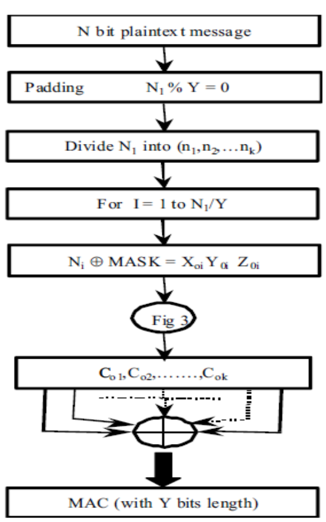


Fig. 2. HMAC Process

### A. Initial Mask Generating Process

A MASK [5] functions as an alias to ensure the anonymity of real sub-messages. This is achieved by applying a MASK to each sub-message (using XOR with the initial MASK) before creating the sub-MACs. These sub-MACs are then XORed together to produce the final message authentication code. User can employ his secret initial points $x_{1_0}$, $x_{2_0}$, $x_{3_0}$,

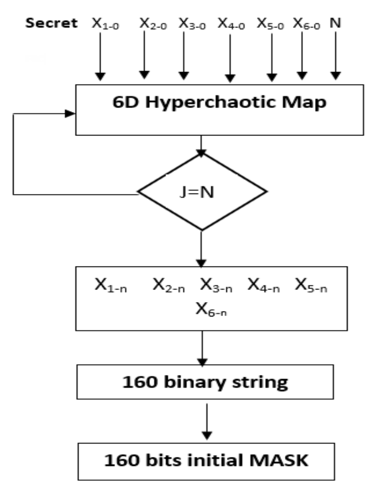


Fig. 3. Initial MASK Generation Process

$x_{4_0}$, $x_{5_0}$, $x_{6_0}$ and N in the 6D map equation to calculate the initial MASK in the figure 3 as follows :

$x_{1_n} = x_{1_0}, x_{2_n} = x_{2_0}, x_{3_n} = x_{3_0}, x_{4_n} = x_{4_0}, x_{5_n} = x_{5_0}, x_{6_n} = x_{6_0}$
For i = 1 to R Do
$x_{1_{(n+1)}} = x_{1_n} + h * (a(x_2 - x_1) + x_4 - x_5 - x_6)$
$x_{2_{(n+1)}} = x_{1_n} + h * (cx_1 - x_2 - x_1 x_3)$
$x_{3_{(n+1)}} = x_{3_n} + h * (-bx_3 + x_1 x_2)$
$x_{4_{(n+1)}} = x_{4_n} + h * (dx_4 - x_2 x_3)$
$x_{5_{(n+1)}} = x_{5_n} + h * (ex_6 + x_3 x_2)$
$x_{6_{(n+1)}} = x_{6_n} + h * (rx_1)$
Change $x_{1_n}, x_{2_n}, x_{3_n}, x_{4_n}, x_{5_n} and x_{6_n}$ to binary strings
Concatenate $x_{1_n}, x_{2_n}, x_{3_n}, x_{4_n}, x_{5_n} and x_{6_n}$ binary strings into Y binary characters string, held in Buffer.
Change the Buffer into Y bits initial MASK.

Where

- (a,b,c,d,e,r) are control parameters (10,$\frac{8}{3}$,28,-1,10,3)
- N is the number of the iterations of the 6D map.
- Y is the MAC lengths (160 bits).
- Buffer is a one-dimensional array with Y bit lengths.

### *B. HMAC Generating Process*

According to the figure 2, the HMAC generating process can be done as follows:

1) $N_1$ bits message is produced, after the padding process on the N bits plain-text message,
2) $N_1$ bits message is divided into $n_k$ sub-message, each with Y bits length where
   a) Y = 160 bits
   b) K = $N_1$/Y
3) For I = 1 to K Do
4) Each sub-message $n_i$ is masked by XORing it with the initial MASK.
5) By using the same steps as in figure 3, we can find the sub-MACs $C_{O_i}$
6) $MAC = C_{O_1} \oplus C_{O_2} \oplus .... \oplus C_{O_k}$

The MAC generation process can be conducted concurrently with encryption, following which the MAC bits are appended to the end of the encrypted text and transmitted to the receiver as a complete encrypted message. Upon reception, Y MAC bits are retrieved, and the encrypted text is decrypted to obtain the plaintext message. Subsequently, the MAC is recomputed and compared to the received MAC. Upon equality, the recipient is assured that the received message has not undergone any inadvertent or malicious corruption.

## IV. 6D-HMAC ANALYSIS

In the 6D-HMAC analysis, three types of analyses were conducted on the generated MAC:

1) **Key-ciphertext avalanche effect:** This analysis evaluates the sensitivity of the MAC to changes in the key and ciphertext, measuring how small modifications in the key or ciphertext lead to significant changes in the MAC output.
2) **Plaintext-ciphertext avalanche effect:** This analysis examines the MAC's ability to conceal changes in the plaintext by producing substantially different outputs for slightly different plaintexts. It measures the effect of changes in plaintext on ciphertext.
3) **Strength Analysis:** This analysis will be based on calculating the key length and the time (Tbreak) required to break the encryption by brute force.

### *A. Key-ciphertext avalanche effect*

The concept of the ciphertext key avalanche effect revolves around modifying the key value and examining its effect on the encrypted text. In 6D-HMAC, the key comprises six distinct initial points of the hyperchaotic system: $x_{1_0}$, $x_{2_0}$, $x_{3_0}$, $x_{4_0}$, $x_{5_0}$, and $x_{6_0}$.

The tables (II to VII) display the outcomes of the avalanche effect analysis on encrypted text with variations in $x_{1_0}$, $x_{2_0}$, $x_{3_0}$, $x_{4_0}$, $x_{5_0}$, and $x_{6_0}$. In this investigation, a fixed message "Testing our new HMAC based on hyperchaotic systems" is employed, with control parameters a = 10, b = $\frac{8}{3}$, c = 28, d = -1, e = 10, and r = 3, and iteration count N=250.

### *B. Plaintext-ciphertext avalanche effect*

The data indicates that even slight alterations in the initial conditions values lead to significant variations in encryption techniques.
This reformulation clarifies the purpose of analyzing how changes in the original message (input) affect the encrypted message (output). It avoids mentioning the "avalanche effect" directly, focusing on the core concept of input sensitivity.

Table VIII presents the results of the avalanche effect between plaintext and ciphertext with different variable message lengths. This analysis utilizes the following initial values: $x_{1_0}$ = 0.100000000000001, $x_{2_0}$ = 0.1, $x_{4_0}$ = 0.11111111111111, $x_{5_0}$ = 0.145635778, $x_{6_0}$ = 0.2035777777777 and N=250.
The 6D-HMAC algorithm demonstrates remarkable sensitivity to even minor alterations within the message. This is elucidated by Table VIII, which offers insightful examples. For instance, removing just one letter, such as 's', yields a drastically different outcome, underscoring the algorithm's acute responsiveness to textual modifications.

### *C. Strength Analysis*

The 6D-HMAC secret relies on six distinct initial points $x_{1_0}$, $x_{2_0}$, $x_{3_0}$, $x_{4_0}$, $x_{5_0}$, and $x_{6_0}$. Depending on the data type used (embedded REAL) and the chosen precision, there are $2^{64}$x $2^{64}$ x $2^{64}$ x $2^{64}$ x $2^{64}$ x $2^{64}$ possible combinations for $x_{1_0}$, $x_{2_0}$, $x_{3_0}$, $x_{4_0}$, $x_{5_0}$, and $x_{6_0}$. This corresponds to a key length of 384 bits. Considering that the variable 'h' in the 6D hyperchaotic system is a randomly selected double value, the key length will increase to 512 bits. Assuming that the computing power of computers is $10^{20}$ operations per second.The value of Tbreak, the time required to break the encryption by brute force. It

TABLE II

KEY ($x_{1_0}$)-CIPHERTEXT AVALANCHE EFFECT (WHERE ($x_{2_0}, x_{3_0}, x_{4_0}, x_{5_0}, x_{6_0}$) = (0.1, 0.1, 0.1, 0.1, 0.1,) AND N=250)

| $x_{1_0}$ | **MAC in Hex** | **MASK in Hex** |
|---|---|---|
| 0.1 | 012HJD2358674JHVBXSD3FF77HJKKTRED8A8D01B4HJY | 7EDXCV2251B9F962KLN93VBLM1547FKLMO21F5D1E7MLK8 |
| 0.11 | 8F3D6E7B12A9C5F044B6A71E29D8C3A5F1B7E6D9YTEG4 | F8J7N4D5W6L3B9C0K5V8N9G4M1T3R6F4L2J82H9B5N4M7C3 |
| 0.111 | A6B713F0C4E4FAD46C27UFEEE0B7C4D4KLO9644E534F0 | 37C9B2E4D20139A4E5F5928F5A7E6C13AAD4970QSAZERTKB |
| 0.1111 | F7B1D2CH93B50ADBF405F0B1TRDSC59E3BDFAC1750E98 | 449F01122A61FA6175915AC3B62D236EABDC8D9CJHJFERTTB |
| 0.11111 | A5063C3E89KJXAD0B59BA8HJGFDDA0CE079HGF1395B68 | 6BAE213D5E9B0F31C8A5DCB45F6E79BCA8D3E12JGRXVBV5 |
| 0.111111 | 5A7426DAD480AHJ3E9F5F75A7B10JD4HHJYT63AA89250A | 2C8B3F6E117C6A0B5ED8E79H23A02B86CF2B857BDHYBF254 |
| 0.1111111 | 91A4A26A9BCF542HS0U1A6A9C544B6F1AAB4ML621495C | 7B16E3B537AD8D8C91E5A37B49C6B28503DE4A7QSZVBL3H |

TABLE III

KEY ($x_{2_0}$)-CIPHERTEXT AVALANCHE EFFECT (WHERE ($x_{1_0}, x_{3_0}, x_{4_0}, x_{5_0}, x_{6_0}$) = (0.1, 0.10000000001, 0.1, 0.1, 0.1,) AND N=250)

| $x_{2_0}$ | **MAC in Hex** | **MASK in Hex** |
|---|---|---|
| 0.1 | 012HJD2358674JHVBXSD3FF77HJKKTRED8A8D01B4HJY | 7EDXCV2251B9F962KLN93VBLM1547FKLMO21F5D1E7MLK8 |
| 0.11 | 8F3D6E7B12A9C5F044B6A71E29D8C3A5F1B7E6D9YTEG4 | F8J7N4D5W6L3B9C0K5V8N9G4M1T3R6F4L2J82H9B5N4M7C3 |
| 0.111 | A6B713F0C4E4FAD46C27UFEEE0B7C4D4KLO9644E534F0 | 37C9B2E4D20139A4E5F5928F5A7E6C13AAD4970QSAZERTKB |
| 0.1111 | F7B1D2CH93B50ADBF405F0B1TRDSC59E3BDFAC1750E98 | 449F01122A61FA6175915AC3B62D236EABDC8D9CJHJFERTTB |
| 0.11111 | A5063C3E89KJXAD0B59BA8HJGFDDA0CE079HGF1395B68 | 6BAE213D5E9B0F31C8A5DCB45F6E79BCA8D3E12JGRXVBV5 |
| 0.111111 | 5A7426DAD480AHJ3E9F5F75A7B10JD4HHJYT63AA89250A | 2C8B3F6E117C6A0B5ED8E79H23A02B86CF2B857BDHYBF254 |
| 0.1111111 | 91A4A26A9BCF542HS0U1A6A9C544B6F1AAB4ML621495C | 7B16E3B537AD8D8C91E5A37B49C6B28503DE4A7QSZVBL3H |

TABLE IV

KEY ($x_{3_0}$)-CIPHERTEXT AVALANCHE EFFECT (WHERE ($x_{1_0}, x_{2_0}, x_{4_0}, x_{5_0}, x_{6_0}$) = (0.100000000000001, 0.1, 0.11111111111111, 0.145635778, 0.2035777777777,) AND N=250)

| $x_{3_0}$ | **MAC in Hex** | **MASK in Hex** |
|---|---|---|
| 0.1 | KJF893NF834NF3984FN3904CN034NC430N4C903NC09N | F1BF239F1BDE7A60E6B7F1CB21349DPOA4C2UI648JHF81B |
| 0.11 | 32JN49D24NB5W7Y04NC30N5C09N3C09N34C903N43NC3 | G39NGF2N39FN34F03NF034NF0CN430N4C903NC4N09N3C09 |
| 0.111 | BJD94NC30NC903N4C903NC430NCTRZI903NC43GRDT2 | N3904C903N9N3C4N39C903N340NC3N0N3C09J4N903CKL30 |
| 0.1111 | JHFNJ4N3904NC903NC430N903NC430NC340N34C90C4 | 294JN34N09N30NC903N4C903NC340N34C903NC4309N3C09 |
| 0.11111 | C9JN3FN3904NC903N4C903NC430N34C903NC43JHG043 | 390N4C9N0393NC9N340NC903N390NC3N90N4NFGX3C90V3N9 |
| 0.111111 | A24FN39FN304NC903N4C903NC430N34C903NC4309NC4 | 9N03N490C3N903NC0N3903NC903N34C3903NC90DSA3N340 |
| 0.1111111 | KJN34N3904NC903N4C903NC430N34C903NC4309NC430 | 5C94N4C903N30N30NC903C9J0N340CN3NC903N3C09N4J309N |

TABLE V

KEY ($x_{4_0}$)-CIPHERTEXT AVALANCHE EFFECT (WHERE ($x_{1_0}, x_{2_0}, x_{3_0}, x_{5_0}, x_{6_0}$) = (0.100000000000001, 0.1, 0.1, 0.145635778, 0.2035777777777) AND N=250)

| $x_{4_0}$ | **MAC in Hex** | **MASK in Hex** |
|---|---|---|
| 0.1 | C7R0A5F9T8V3W6X2Y4Z1BQNEMJPLKOIHTUGFVCDXSWZAR | D8S1G6U3V2W5X4Y7Z0BHNEMJPLKOIHTUGFVCDXSWZAR |
| 0.11 | E9T2H7V4W3X6Y0Z5CINEMJPLKOIHTUGFVCDXSWZARQSB | F0U3I8W5X4Y1Z6DJNEMJPLKOIHTUGFVCDXSWZARQSBTG |
| 0.111 | F0U3I8W5X4Y1Z6DJNEMJPLKOIHTUGFVCDXSWZARQSBTG | G1V4J9X6Y5Z2EKINEMJPLKOIHTUGFVCDXSWZARQSBTGU |
| 0.1111 | M7B0PK5D8E3F9IQMJPLKOIHTUGFVCDXSWZARQSBTGWL | N3903N49C03N9C390N4N903N390N4C90N34N03N490N3 |
| 0.11111 | I3X6L1Z7A4B5GMNEMJPLKOIHTUGFVCDXSWZARQSBTGWI | J4Y7M2A5B0C6HNEMJPLKOIHTUGFVCDXSWZARQSBTGWIX |
| 0.111111 | K5Z8N3B6C1D7IOEMJPLKOIHTUGFVCDXSWZARQSBTGWJY | L6A9O4C7D2E8JPENMJPLKOIHTUGFVCDXSWZARQSBTGWK |
| 0.1111111 | KJN34N3904NC903N4C903NC430N34C903NC4309NC430 | 5C94N4C903N30N30NC903C9J0N340CN3NC903N3C09N4J309N |

TABLE VI

KEY ($x_{5_0}$)-CIPHERTEXT AVALANCHE EFFECT (WHERE ($x_{1_0}, x_{2_0}, x_{3_0}, x_{4_0}, x_{6_0}$) = (0.1, 0.1, 0.10000000000001, 0.222222222222, 0.1) AND N=250)

| $x_{5_0}$ | **MAC in Hex** | **MASK in Hex** |
|---|---|---|
| 0.1 | C7R0A5F9T8V3W6X2Y4Z1BQNEMJPLKOIHTUGFVCDXSWZAR | H8LN5GM9NM9YO4QST7UV2WZB7D4HE1IF0JG5MD3N0XA |
| 0.11 | F6JL3EK7LK7WM2POQR5ST0UXA5B2FC9GD8HE3JKB1LY | G7KM4FL8ML8XN3QPRS6TU1VYA6C3GD0HE9IF4KLC2MZ |
| 0.111 | H8LN5GM9NM9YO4QST7UV2WZB7D4HE1IF0JG5MD3N0XA | I9MO6HN0ON0ZP5TUV8WX3YAC8E5IF2JG1KH6NE4O1YB |
| 0.1111 | O5SU2NT6UT6FV1ZAB4CG4K1OL8PM5TN2KU6V0W47EH | P6TV3OU7VU7GW2BC5DH5L2PM9QN6UO3LV7W1X58FI3 |
| 0.11111 | Q1FT4GO9H2I7M3QOJPLKOIHTUGFVCDXSWZARQSBTGWP | R2GU5HP0I3J8N4RPJPLKOIHTUGFVCDXSWZARQSBTGWQ |
| 0.111111 | S3HV6IQ1J4K9O5SQJPLKOIHTUGFVCDXSWZARQSBTGWR | T4IW7JR2K5L0P6TRJPLKOIHTUGFVCDXSWZARQSBTGWS |
| 0.1111111 | U5JX8KS3L6M1Q7USJPLKOIHTUGFVCDXSWZARQSBTGWT | V6KY9LT4M7N2R8VTJPLKOIHTUGFVCDXSWZARQSBTGWU |

TABLE VII

KEY ($x_{6_0}$)-CIPHERTEXT AVALANCHE EFFECT (WHERE ($x_{1_0}, x_{2_0}, x_{3_0}, x_{4_0}, x_{5_0}$) = (0.1000000001, 0.2222222222, 0.10111111111, 0.1, 0.1) AND N=250)

| $x_{6_0}$ | **MAC in Hex** | **MASK in Hex** |
|---|---|---|
| 0.1 | W7ZX0Y4CA8VB3ND2MKER3FG5HI9JO1PLQ6SU2WY5TX7ZR | X8AY1Z5DB9WC4OE3NLF4GH6IJ0KP2MQ7TV3XZ6UY8ASR |
| 0.11 | Y9BZ2CE6XD0PF5OMG5HI7JK1LQ3NR8UW4YV9TZ1AX0BS | Z0C1DF7YE1QG6PNH6IJ8KL2MR4OS9VX5ZW0AU3YB2CTD |
| 0.111 | C3GI0BH4IH4TJ9SMNL1OP2RU7VW2Y8CZ3DX6AE5BG0FW | D4HJ1CI5JI5UK0TNMO2PQ3SV8WX3Z9D0EA7BF6CH1GXY |
| 0.1111 | A1E2FG8ZF2RH7QOJ7KL9MN3PS5TU0WY6AX1BV4ZC3DUE | B2FH9AG3HG3SI8RPLK0MN1QT6UV1XZ7BY2CW5AD4EZFV |
| 0.11111 | E5IK2DJ6KJ6VL1UONP3QR4TW9XY4A1EB8FC7DG2IJA0Z | F6JL3EK7LK7WM2POQR5ST0UXA5B2FC9GD8HE3JKB1LY |
| 0.111111 | S9WY2BE6CG0XF5OH4JM8KP1LQ3NR7TV4WX9ZC0AD2YB3ZD | T0XZ3CF7DH1YG6PI5KO9LQ2MS4NU8WX5YDA1B2ZE3XC4FE |
| 0.1111111 | Q7UW0ZC4AE8VD3MF2GK6HN3IO9JP1LQ5RT2SY6UX4WB7XZ | R8VX1AD5BF9WE4NG3IL7JO0KP2MQ6SU3TX7VY5WZ8XC1YA |

TABLE VIII

PLAINTEXT-CIPHERTEXT AVALANCHE EFFECT

| **Message** | **Length MAC** | **MAC in HeX** |
|---|---|---|
| Testing our new HMAC based on hyperchaotic systems | 45 | KJN34N3904NC903N4C903NC430N34C903NC4309NC430 |
| Testing our new HMAC based on hyperchaotic system | 44 | FBFFC92A0AE81471B40A256B79B49D2F33DA2ERB13E |
| Testing HMAC based on hyperchaotic | 33 | F5D9A28E615C07B436B3E192H8G2J0A7K3L9M4N5O6PE |
| hyperchaotic | 12 | FR3W5E1T9Y7U2I0O8P6A4S3D2F1G5H4J7K9L8ZXC4V6 |

is necessary to try all possible combinations until the correct one is found.

- Tbreak = (Total number of possible keys) * (Computation time per key)
- Tbreak = $2^{512}$ * 1 operation / key
- Tbreak ≈ 3.4028236692093846 * $10^{154}$ seconds

With a key length of 512 bits and a theoretical computational power of $10^{20}$ operations per second, the estimated time to break the encryption by brute force (Tbreak) is on the order of $10^{154}$ seconds. This figure underscores the exponentially increasing difficulty of breaking encryption by brute force with the lengthening of the key. It emphasizes the critical importance of using long keys and robust encryption methods to ensure data security.

## V. CONCLUSION

This study introduces a novel hybrid message authentication method combining hash functions with 6D hyperchaotic maps, utilizing a large key space. Tests were conducted to evaluate the robustness and effectiveness of this approach, termed 6D-HMAC.

## REFERENCES


[1] M. Najjar, d-HMAC — An improved HMAC algorithm, International Journal of Computer Science and Information Security (IJCSIS), Vol. 13, No. 4, 2015.

[2] N. Koblitz nk Alfred, Another look at HMAC, MenezesAlfred Menezes, Journal of Mathematical Cryptology, September 2019.

[3] Lingzhi, Y.;Weihong, X.;Wenxin, Y.; Binren,W. Dynamical analysis, circuit implementation and deep belief network control of new six-dimensional hyperchaotic system. J. Algorithms Comput. Technol. 2018, 12, 361–375.

[4] J. Wang, W. Yu, J. Wang, Y. Zhao, J. Zhang, and D. Jiang, “A New Six-dimensional Hyperchaotic System and Its Secure Communication Circuit Implementation”, International Journal of Circuit Theory and Applications, Vol. 47, No. 5, pp. 702–717, 2019.

[5] S. Idris, H.Zorkta, S.Khawatmi, and W. Aiyash, “ENHANCED HMAC BASED UPON 3-D ROSSLER SYSTEM,” IInternational Conference on Future Computer and Communication.