Proceeding of the 1st ICEENG conference, 24-26 March, 1998. AC1- 456

MILITARY TECHNICAL COLLEGE
CAIRO-EGYPT

ICEENG 98

FIRST INTERNATIONAL CONF.ON
ELECTRICAL ENGINEERING

NEW MESSAGE AUTHENTICATION CODE USING UNIMODAL CHAOTIC MAPPING

Alaa Fahmy*

ABSTRACT

Message authentication is a procedure established between two ends, (sender and receiver), allows each end to verify the integrity of the received message. The contents of a message are authenticated by verifying the correctness of message authentication code (MAC), which is computed by the sender and is appended to the message. In this paper, we present a new method to generate the MAC, using unimodal chaotic mapping.

1. INTRODUCTION

The unpredictability of chaotic dynamical systems arises from the fact that small error in the initial conditions corresponds to different change in behavior, i.e., chaotic dynamical systems are sensitive dependence on initial conditions [1]. When the system is allowed to evolve, it loses all the information about the initial condition therefore, we used this property to generate the message authentication code.

2. MESSAGE AUTHENTICATION CODE GENERATION

The unimodal map given by eq.(1) has been used to generate the MAC.

$$X_{n+1} = \mu . X_n . (1 - X_n) \quad (1)$$

*Ph.D., Military Technical College.

The Message authentication can be constructed as follows:

(i) Divide the encoded plaintext, Z, into N blocks, $\{Z_i\}_{i=1:N}$, of fixed size b (56 bits). Pad Z if necessary.

(ii) Use the transformation F: $Z_i \rightarrow R_i$ to transform the binary block, Z_i, into

decimal number $R_i \in (0,1)$, and assign R_i to an initial condition x_{oi}.
(iii) Iterate eq.(1) with an initial condition x_{oi} and with $\mu \geq 3.9$ (chaotic region). The number of iterations k is set to 300 iterations. Only the last b iterations are taken and therefore, we get the sequence $x_{k-b+1}, x_{k-b+2}, \ldots, x_k$. If we assign "1" to the points, which are greater than 1/2 and "0" to the points, which are less than 1/2, the evolution of the logistic map will generate, from every initial condition (x_{oi}), a sequence (Y_i) of 1s and 0s, which resemble the tossing of a coin.

(iv) Repeat the steps (ii-iii) N times and exclusive-OR the blocks of the output sequence Y_i, we get the function $ß(M) = Y_1 \oplus Y_2 \oplus \ldots \oplus Y_N$. The function ß transforms a plaintext M of arbitrary length into a relatively short, fixed length (56 bits), pattern of bits ß(M), with a high probability that ß (M) will differ from ß(M`) whenever the recovered plaintext (M`) different from the original plaintext (M).

(v) Encipher the function ß(M), using the cryptographic algorithm presented in [2]. The enciphered value of ß(M) represents the MAC.

At the other end (receiver) the same procedures are applied to reproduce the message authentication code. The receiver compares the calculated message authentication code with the received one. If they are equal, the message is accepted (genuine message) otherwise, it is rejected (forge message).

3. RESULTS AND DISCUSSIONS

The unimodal map, given by eq.(1), can be regarded as a one-way function. That is, given an initial condition x_o, it is feasible to calculate x_{n+1}, but it is computationally infeasible to calculate x_o given x_{n+1}. Since the construction of the

MAC is based on such function, it is computationally infeasible to generate two messages having the same MAC, or to produce any message having a prescribed MAC.

The number of operations required for such procedures is of the order of 2^{56} operations. Whilst the MAC derived from the data encryption standard (DES) algorithm, using either cipher block chaining mode or cipher feedback mode, has 2^{24} and 2^{32} operations for telecommunication and financial applications respectively [3]. The constructed MAC is simple and it is convenient whenever the cryptographic algorithm is available either in software or in hardware. In contrast, the MAC generated from the DES is convenient only if a hardware DES function is available because the algorithm is slow and inconvenient in software. In addition, a single error in the ciphertext causes a single error in the recovered plaintext, and a drastic change in the message authentication code.

The cryptosystem presented in [2], which is based on stream cipher idea can be used to illustrate the error propagation and key-ciphertext avalanche effect. Figure 1 illustrates the error propagation property. Changing one bit in the ciphertext causes a drastic change in the message authentication code. It is also clear from Fig.2 that small change in the initial conditions, with fixed plaintext, will result to radical change in both the ciphertext and the MAC.

Therefore, we can use this idea of MAC generation with the conventional cryptographic systems, based on stream cipher idea, for both data transmission and authentication applications. Figure 3 illustrates error propagation and self synchronization (single bit change in the ciphertext causes a single bit change in the recovered plaintext) property when enciphering both the plaintext and the generated MAC with the cryptographic algorithm presented in [2].

4. CONCLUSION

A new method to generate the message authentication code is introduced. It is based on the sensitive dependence on initial conditions of the chaotic systems. In this way, any single bit change in the ciphertext results in a drastic change in the message authentication code. It can be used with cryptographic algorithms, based on stream cipher idea for both data transmission and authentication applications.

5. REFERENCES

[1] Hao Bai-Lin, Chaos II.

[2] Sobhy, M.I. and Fahmy, A., "Cryptographic Algorithm Based on Chaotic Behavior", Tenth National Radio Science Conference, Cairo, Egypt, Feb. 16-18, 1993.

[3] National Bureau of Standards, "DES Modes of Operation," FIPS Publication 81, Dec. 1980.

Ciphertext C

A 3 4 B 2 2 D 0 8 C F 0 0 F
B 7 B 4 8 7 8 0 9 D D 0 C B

Ciphertext C'

A 2 4 B 2 2 D 0 8 C F 0 0 F
B 7 B 4 8 7 8 0 9 D D 0 C B

MAC

D 3 D F A 6

MAC'

6 4 D E F 6

Fig.1 MAC For Single Bit Change in Ciphertext

Message (Plaintext) M

HELLO HELLO

Plaintext in Hexadecimal Form

4 8 4 5 4 C 4 C 4 F 2 0 4 8
4 5 4 C 4 C 4 F D 3 D F A 6

Ciphertext C For X_0 =0.1268570439

A 3 4 B 2 2 D 0 8 C F 0 0 F
B 7 B 4 8 7 8 0 9 D D 0 C B

MAC

6 4 D E F 6

Ciphertext C' For X'_0 =0.1268570434

A C 9 3 2 F C A 2 C B E 6 8
5 3 8 B 2 F 6 8 A E 9 4 1 A

MAC'

E 0 9 B 7 7

Fig.2 Key-Ciphetext Avalanche Effect

Message M

HELLO HELLO

Ciphertext C

A 3 4 B 2 2 D 0 8 C F 0 0 F
B 7 B 4 8 7 8 0 9 D D 0 C B

MAC

D 3 D F A 6

Ciphertext C'

A 2 4 B 2 2 D 0 8 C F 0 0 F
B 7 B 4 8 7 8 0 9 D D 0 C B

Recovered Plaintext M'

IELLO HELLO

MAC'

6 4 D E F 6

Fig.3 Self Synchronization and Error Propagation Property