A cryptographic hash function based on chaotic network automata

To cite this article: Jeaneth Machicao and Odemir M Bruno 2017 J. Phys.: Conf. Ser. 936 012058

View the article online for updates and enhancements.
A cryptographic hash function based on chaotic network automata

Jeaneth Machicao$^1$ and Odemir M Bruno$^1$

$^1$ São Carlos Institute of Physics, University of São Paulo. Av. Trabalhador São Carlense, 400, São Carlos, SP - Brazil

E-mail: bruno@usp.br

Abstract.
Chaos theory has been used to develop several cryptographic methods relying on the pseudo-random properties extracted from simple nonlinear systems such as cellular automata (CA). Cryptographic hash functions (CHF) are commonly used to check data integrity. CHF “compress” arbitrary long messages (input) into much smaller representations called hash values or message digest (output), designed to prevent the ability to reverse the hash values into the original message. This paper proposes a chaos-based CHF inspired on an encryption method based on chaotic CA rule B1357-S2468. Here, we propose an hybrid model that combines CA and networks, called network automata (CNA), whose chaotic spatio-temporal outputs are used to compute a hash value. Following the Merkle and Damgård model of construction, a portion of the message is entered as the initial condition of the network automata, so that the rest parts of messages are iteratively entered to perturb the system. The chaotic network automata shuffles the message using flexible control parameters, so that the generated hash value is highly sensitive to the message. As demonstrated in our experiments, the proposed model has excellent pseudo-randomness and sensitivity properties with acceptable performance when compared to conventional hash functions.

1. Introduction
Cellular automata has been widely applied in many domains of science including physics, biological systems modelings, image processing, and Cryptography. Some CAs present chaotic properties (such as ergodicity, high sensitivity to initial conditions, deterministic dynamics, structural complexity and long periodicity) that can meet desired confusion/diffusion properties and random-like behavior for cryptographic purposes. This close relation between chaos and cryptography have encouraged many researchers to develop chaos-based cryptographic primitives such as hash functions [1, 2, 3]. A CHF is a one-way function that maps an arbitrary finite length message to a fixed-length output, leading a fundamental role in digital signature, document compression, document authentication, etc [4].

In previous work [5], the chaotic properties of the life-like CA with rule B1357/S2468 have shown to be a good source of pseudo-randomness, being appropriate for the construction of cipher algorithms. These CAs were originally built into a 2D lattice, whose dynamics is governed by a rule and its local neighborhood that defines the states of its cells at each time step. However, in order to prevent weaknesses and cryptanalytic attacks of CHFs (by predicting the chaotic orbits), further enhancement can be introduce by considering a non-spatial neighborhood, i.e. to use a network structure as the tessellation of the CA, called as a network automata [6].
In this paper, we propose a cryptographic hash function scheme based on the chaotic properties of network automata. Thus, the hashing processing relies on the Merkle-Damgård scheme: (i) by mixing a block of message together with the outputs from the evolution of the CNA and (ii) to feedback the previous outputs as entrance for the hashing processing with the next block of messages, in an iterative manner. Thus, through these iterations, the final hash value is obtained by extracting 512 bits from the ultimate state of the network automaton.

2. Chaotic network automata
The classical two-dimensional CA gave the inspiration for the network automata model (see Fig. 1). Instead of a lattice, a network becomes the tessellation of the automaton, where each vertex $v_i$ represent a cell $c_i$ that is connected to its $k_i$ neighbors, and a local rule specifies how the NA evolves in time $t$. According to the nomenclature of the Life-Like rules family, for example rule B1357/S2468 (Eq. 1) implies that a cell $c_i$ “born” when it is surrounding by $1 \lor 3 \lor 5 \lor 7$ alive neighbors. Contrarily, a cell “survives” when there exists $2 \lor 4 \lor 6 \lor 8$, otherwise dies.

$$s(c_i, t + 1) = \begin{cases} 
1, & \text{if } \sum_{i=1}^{k_i} s(c_i, t) = 1 \lor 3 \lor 5 \lor 7 \Rightarrow \text{born (B)} \\
1, & \text{if } \sum_{i=1}^{k_i} s(c_i, t) = 2 \lor 4 \lor 6 \lor 8 \Rightarrow \text{survive (S)} \\
0, & \text{otherwise}
\end{cases}$$  

3. Proposed cryptographic hash function
We propose a cryptographic hash function based on the chaotic spatio-temporal outputs from network automata. In general, the design of CHF consists of two components [8]: (i) a compression function $H : \{0,1\}^m \rightarrow \{0,1\}^n$, that maps a fixed-length message to a fixed-length output and (ii) the domain extender algorithm, which is a generic construction that transforms the former function to a hash function taking a message of arbitrary length $M \in \{0,1\}^\ast$ and returning an output of fixed-length $n$ [8]. Thus, we follow the Merkle-Damgård construction [7], which is one of the simplest and most used domain extender in literature.

The propose CHF is illustrated in Fig. 2. First, a message $M$ is broken into $m$-bit blocks (with padding if necessary), such that $M = \{M_1, M_2, \ldots, M_b\}$. Here we considered $m = 512$, as many conventional algorithms do. The initial conditions of the NA at $t = 0$ is setup by considering the binary elements from the first block $M_1$. Then, iteratively, every input block $M_i$ is xored with the outputs states $H_i$ from the evolving CNA and the resulting output is used to perturb the system for the next iteration, until a final hash value $H(M)$ is composed by the $n$-bits corresponding to the ultimate states of the network automaton, with $n = 512$.

Regarding the configuration of the NA, we considered a network with 512 nodes, each of them is connected to 8 non-local neighbors, which are randomly chosen. This connectivity parameter...
brings a large number of possible combinations. For our experiments, we considered a random fixed neighborhood, however many other approaches could be considered, such as introducing a hash key to determine the configuration of this parameter. We use a NA with rule B1357/S2468, as this rule has been proven to be chaotic by means of the positive Lyapunov exponent explored in previous work [5], which is maintained since the proposed CNA also contains 8 neighbors.

4. Experimental results and discussion
To show that our proposed method ensures that even a small change on the message can causes a substantial difference on the hash value. We have compared the number of bits changed among the hash values of two close messages: \( \mathcal{M} = \) “Schrodinger’s cat” and \( \mathcal{M}' = \) “schrodinger’s cat”

\[
H(\mathcal{M}) = \text{3e300fa7c550b453d85b16cb23111899520576b3434a46aa40cd1faaceed7} \quad \text{and} \quad H(\mathcal{M}') = \text{1801cb79c18a9800adbcb4ce5641f42cc830000611424e6792a7a22597540944}
\]

which differs in 257 bits.

In general, when a minimal disturbance on the original message is introduced, an ideal CHF should produce a hash value with 50% of changing probability in each bit (avalanche effect). A well-known security test [1, 2, 3] consists to perform statistical analysis on a set of ensembles denoted by \( \mathcal{E} = \{ R^e | e = 1, \ldots, n \} \), where \( R \) represents a replica of the original message \( \mathcal{M} \) except that the \( e \)-th bit is changed, i.e. \( R^e = \mathcal{M}_e \) (Boolean complement).

Similar to the former experiment, we systematically calculated the distance between two hash values \( \mathcal{E}_e = d( H(\mathcal{M}), H(R^e) ) \). These tests were run for several lengths, using different number of blocks. The following measures were calculated: the mean number of changed bits \( \bar{B} \), the minimum and maximum number of changed bits \( B_{\min} \) and \( B_{\max} \), standard deviation \( \Delta B \) and the mean changed probability: \( P = \frac{\bar{B} \times 100}{n} \). The results for this security test are summarized in Table 1. Here, we can observe very similar results when comparing our proposal with conventional hash algorithms such as Whirlpool algorithm [9]. This result reinforces the potential of the proposed algorithm and its good level of security.

The performance of the algorithm was also measured in terms of the average computing time (in seconds) depending on the size of the input message. For this experiment, we applied the proposed CHF using 10 random messages with different numbers of message blocks \( b = \{1 \times 10^5, \ldots, 10 \times 10^5\} \). Fig. 3 presents a time performance comparison between the proposed CHF and the Whirlpool algorithm [9].

5. Conclusions
A cryptographic hash function based on a network automata was proposed, which takes advantages from rule B1357/S2468 which has a chaotic behavior. The proposed CHF possesses
Table 1. Statistical analysis comparison of the proposed CHF and Whirlpool algorithm.

|        | # blocks | \( \bar{B} \) | \( \Delta B \) | \( B_{\text{min}} \) | \( B_{\text{max}} \) | P       |
|--------|----------|----------------|--------------|----------------|----------------|---------|
| Proposed CHF          |          |                |              |                |                |         |
| \( b = 2 \)       |          | 255.6          | 3.8          | 198            | 323            | 49.9%   |
| \( b = 4 \)       |          | 255.8          | 3.7          | 189            | 315            | 50.0%   |
| \( b = 6 \)       |          | 255.9          | 3.6          | 197            | 313            | 50.0%   |
| \( b = 8 \)       |          | 255.2          | 3.7          | 190            | 319            | 49.9%   |
| \( b = 10 \)      |          | 255.2          | 3.7          | 199            | 318            | 49.8%   |
| Whirlpool [9]      | \( b = 10 \) | 255.9          | 2.9          | 216            | 298            | 50.0%   |

Figure 3. Time performance comparison between the proposed CHF and the Whirlpool algorithm [9] regarding to the number of blocks messages.

good statistical properties, high flexibility (network connectivity and CNA initial conditions), and good collision resistance, since it was shown that given any change in the original message, even minimal, it will eventually generate a significant change in the final hash result. Although, the algorithm has a lower performance when compared to a classical algorithm, it does not underestimate its equivalent security, flexibility and easy construction.

Acknowledgements

J.M. acknowledges the Brazilian agency CAPES (PROEX). O.M.B. acknowledges support from CNPq (Grant #307797/2014-7 and Grant #484312/2013-8) and FAPESP (grant #14/08026-1).

References

[1] Kanso A and Ghebleh M 2013 Communications in Nonlinear Science and Numerical Simulation 18 109–123
[2] Yang H and et al 2009 Chaos, Solitons & Fractals 41 2566–2574
[3] Xiao D, Liao X and Deng S 2005 Chaos, Solitons & Fractals 24 65–71
[4] Bakhtiari S, Safavi-naini R and Pieprzyk J 1995 Cryptographic hash functions: A survey Technical report 95-09 Department of Computer Science, University of Wollongong New South Wales, Australia
[5] Machicao J, Marco A G and Bruno O M 2012 Expert Systems with Applications 39 12626–12635
[6] Miranda G, Machicao J and Bruno O M 2016 Scientific Reports 6
[7] Coron J S and et al 2005 Merkle-damgård revisited: How to construct a hash function Advances in Cryptology: CRYPTO 05 Proceedings, Springer-Verlag pp 430–448
[8] Mironov I 2005 Hash functions: Theory, attacks, and applications Tech. Rep. MSR-TR-2005-187 Microsoft Research Moffett Field, CA, USA
[9] Barreto P and Rijmen V 2000 First open NESSIE Workshop URL http://www.larc.usp.br/ pbarreto/WhirlpoolPage.html