UNIVERSITY OF PITESTI

Str. Targu din Vale, No. 1, 110040 - Pitesti, ROMANIA

Faculty of Electronics, Communications and Computers

Exploring Hybrid Cellular Automata (HCA) for cryptographic applications

Author: *Petre ANGHELESCU*

petre.anghelescu@upit.ro

Abstract / Overview Summary

The main objective of the project is the design and the analysis of an HCA based pseudo-random number generator (PRNG) with properties that make it suitable for use in cryptography. As the development of HCA based applications is generally an experimental effort, the project implies the exploration, through simulation, of a huge space of CA local rules and global states. In this project it is shown how a series of simple elements called "cells" interact using certain rules and topologies to form a larger system that acts like a PRNG. This is used to encrypt/decrypt data using an XOR logic function.

Results:

I found interesting Hybrid Cellular Automata rules with properties that make them suitable for cryptographic applications.

The work proves that HCA can generate maximal length pseudo-random numbers and can be used as a part of an encryption system.

HCA analyses

Fig. 1 - 2-state, 3-neighborhood CA using Rule 90

Fig. 2 - 2-state, 3-neighborhood CA using Rule 150

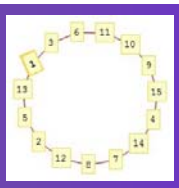
The global transition function of an n-cell HCA with rules 90 & 150 can be represented as a n x n square matrix referred to as the characteristic matrix of the HCA. The characteristic matrix is constructed as follows:

T[i,j] = 1, if the next state of the i^{th} cell depends on the present state of the j^{th} cell.

= 0, otherwise.

$$\begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}^3 \times \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 1 \end{pmatrix}$$

Now, we can **calculate** the **next state** of the HCA by **multiplying** (modulo 2) the **present state vector** (the state of the HCA) by the **characteristic matrix**. For example after three steps, a four cell HCA configured with template {1,1,0,1} and with initial state {0, 0, 0, 1} will be:



In the next figure we show a state transition graph of the 4-bit HCA with template {1,1,0,1} and with initial state {0, 0, 0, 1}. Each node of the transition graph represents one of the possible states of the HCA. Each edge of the graph corresponds to a single time step transition of the automaton.

Testing and Results

message (PlainText) = "NKS Summer School 2011! Exploring Hybrid Cellular Automata (HCA) for cryptographic applications. The main objective of the project is the design and the analysis of a HCA based pseudo-random number generation (PRNG) with properties that make it suitable for use in cryptography. As the development of HCA based applications is generally an experimental effort, the project implies the exploration, through simulation, of the huge space of CA local rules and global states. In this project it is shown how a series of simple elements called cells interact between each other using certain rules and topologies to form a larger system that acts like a PRNG further used to encrypt/decrypt data using XOR logic function.";

encryptedMessage (CipherText) =

"ì¾Dî0B4«=ÊxÄwÚ`@«&ÒxÚÔ+é¦7ÆÖTÍÑLC0«+ÖzÉÉ$Ê9Êý~ÃeÞ7_
%ì¦?Ò~Ë$_=eµ*ËyÛe¾G4Zaµ7IOxE×
@8\fiô1ÃÕcÀØeD>î·*dÉ−5ø½9ÌÕvÆÙeG4Zê°?ÎdÁÎeIq«ãÕuÉÎ
!\tî¡ÍØeÉÓ!Bqþ¹<Ç7ÏØ+D0â»0ÝGúóÃ&ÿ¼~ÒxØØ7F4\t«
6Ã7ÅÜ.8«§+ËvÊÑ
ÊI>«¡−ÇÕ~Æ&V!ä³,ÃÑe«\\qã±~ÆaÍÑ*B4ÿô1ÃÕ_ëüeN\"ïô?Ò8ÁÞ$F>øô
7ÑÕpÍÓ N=ôô?ÌÕrÐÍF<å ?ÎÕrÎÛ*@<Zÿ¼;eÇ×
@qæ¤2ËdÉ−4û¸1ÐcÁÒ+Æ%ù»+Á7ÛÔHC0â»0ÕxÎ1Jqþ³;gÉÞ
Ê@7ZÈ~ÎtÉÑeZ=øôʔÌ7ÏÑ*N=Zø
?Öd\f%âʃ~ÒxÂØ&8«½−ÇÊ+ÊG>\r«µ~ÑeÁØ6Ê@7Zø½3ÒrØL
B4ÿ§~Á8ÅØ!ÊL4ç§~ËcÍÏ$@qî LÇyØ$Gqÿ¼;ÐÕbÛÔ+2ù
?Ë7ÛÈL\\qå°~ÕgÇÑ*F4\t« 1xÛÐe=ù³;ÐÕdÑÍ1Bqaµ*tÜÎeF:«µ~ò§Yï□□
D%î¦~×rÌ14è¦'Ò8ÎØ&V!«°?Ö7ÝÎ,Hq\"Ä~ÎpÁÞeZ?ÿ½1ÌÛ"

The *Mathematica* source code for the encryption process is:

```
ETCAStep[cells_, temp_] := MapThread[IntegerDigits[#2, 2, 8][[-FromDigits[ #1, 2] - 1]] &, {Partition[cells,
3, 1, {2, 2}, 0], temp}]ETCA[init_, temp_, t_] := NestList[ETCAStep[#, temp] &, init, t]
prngEvolution = ETCA[CAInitialState, IntegerDigits[RulesArrangement[[CALength - 3]], 2, CALength] /.{0 →
90, 1 →150}, CAEvolSteps] ;
encryptedMessage = FromCharacterCode[ FromDigits[ #1, 2] & / @ Partition[Flatten[Table[
Fold[BitXor[ #1, prngEvolution[[ #2]]] &, binaryMessage[[i]], Range[CAEvolSteps + 1]],
{i, 1, Length @ binaryMessage}]][[1 ;; -(supplementaryBits + 1)]], 8]];
```

CONCLUSIONS

The project presents the methodology for the development of a cryptosystem with HCA. This implies a huge simulation effort in order to find and select the arrangements of local rules, combined with appropriate initial states and topology, which can be effectively applied in cryptography.

The general conclusion of this work is that it is possible to build evolutionary encryption systems based on a simple mathematical model specified by HCA which use local interactions between cells, local rules, and their robustness to build various models. This implies that there are many opportunities for the implement very efficient encryption systems in software as well as in hardware.

Acknowledgement:

This work was supported by CNCS UEFISCDI, project number PN II-RU PD 369, Contract No. 10/02.08.2010.