Dynamic Synthesis and Analysis of Maximum Length Linear and Nonlinear Cellular Automata

Abhrajit Sengupta, Shamit Ghosh and **Dipanwita Roy Chowdhury**

Indian Institute of Technology, Kharagpur

June 10, 2015

Introduction

- Random numbers are fundamental to many cryptographic applications
  - used in stream ciphers, key generators and nonces
  - an ideal concept
- Usually approximated by PRNG
- Traditionally LFSR and LHCA were used as PRNG
  - can be readily analyzed compromising security
- As a solution *nonlinearity* was introduced
  - suffers from low periodicity
- Current PRNGs using FSRs or CAs are hard to alter dynamically

- Design of a readily alterable dynamic PRNG
- Proof of nonlinearity and maximum length cycle
- Analysis of nonlinearity and statistical properties

Dynamic Construction of Maximum Length LHCA

Dynamic Construction of Maximum Length LHCA

- In 1996, Cattel and Muzio presented a mapping between *irreducible* polynomial and LHCA
  - irreducible polynomial does not ensure maximum periodicity of LHCA
- There exists one-to-one correspondence between *primitive* polynomial and maximum-length LHCA
  - maps finding maximum-length LHCA to that of finding primitive polynomial
- However, checking of primitivity is too costly
  - requires factorization of $2^n - 1$
  - requires checking primitive root of the defining polynomial, necessitating exponentiation

Dynamic Construction of Maximum Length LHCA

Theorem 1

If $2^n - 1$ is a prime number (Mersenne Prime) then every degree n irreducible polynomial is also primitive

- Finding primitive polynomial changes to finding irreducible polynomial
- Work with polynomials of degree $n : 2^n - 1 \in \mathbb{P}$

Dynamic Construction of Maximum Length LHCA

- A polynomial $f(x) = \Sigma b_i \cdot x^i, 0 \leq i \leq n$ is represented as a bit-string $b_n, b_{n-1}, \cdots, b_0$
- Random patterns of $(n-1)$ bits with 1 added at both ends are produced
  - MSB is 1 as $f(x)$ is monic
  - LSB also should be 1, otherwise x will be a trivial factor of $f(x)$
- Finally, $f(x)$ is subjected to Rabin's irreducibility test

Dynamic Construction of Maximum Length LHCA

Algorithm 1: Generation of maximum length LHCA

Input: An initial seed r
Output: A maximum length LHCA rule Vector

1: $s \leftarrow \textsc{Rand}(r)$
2: $\mathcal{S} \leftarrow 1||s||1$
3: **while** $\textsc{IsIrreducible}(\mathcal{S}) =$ FALSE **do** ▷ Rabin's Test
4: $\quad s \leftarrow \textsc{Rand}(s)$
5: $\quad \mathcal{S} \leftarrow 1||s||1$
6: **end while**
7: $rule \leftarrow \textsc{SythesizeCA}(\mathcal{S})$
8: **return** $rule$

Dynamic Construction of Maximum Length LHCA

Theorem 2

The expected running time of Algorithm 1 is $\mathcal{O}(n)$.

- Number of irreducible polynomials of degree n over $\mathbb{F}_q$
  $I_n = \frac{1}{n} \sum_{k|n} \mu(k) q^{n/k}, \quad \mu \rightarrow$ Möbius function
- Follows that $q^n - 2q^{n/2} \leq nI_n \leq q^n$
- A fraction very close to $\frac{1}{n}$ are irreducible
- On average Algorithm 1 produces an irreducible polynomial with n number of trials

Dynamic Construction of Maximum Length LHCA

Table 1 : Synthesis of CA from a Primitive Polynomial

LHCA bits	Primitive Polynomial	CA rule
7	0xe5	0x12
31	debab241	63c44d9b
61	0x2c3b579be4a2eee1	0x25e9034de86d7fa
89	0x3db838f8e174ed136dd3515	0x1c39fd02bd393870f075167
127	0xd6f6033bd7f334f1a38b09020e145937	0x5320dc7cd47e7581f5e3846d0bd7840a

Table 2 : Computation Time of synthesis of $10,000$ LHCA rules

# of cells	Time Consumed(second)	Throughput (rules/second)
31	8.364	1195.60
61	33.743	296.36
89	80.176	124.73
127	186.137	53.73

Nonlinearity Injection

Nonlinearity Injection

- A maximum length LHCA is converted to a maximum length NHCA
- Nonlinear functions are injected into selected positions of the LHCA
- Maximum length property is ensured with additional boolean functions
- A term *shifting operation* is defined for this purpose

Nonlinearity Injection

Definition 1

The one cell shifting operation, denoted by $f_i \xrightarrow{P} f_{i\pm1}$ moves a set of ANF monomials P from i-th cell of an NHCA to all the cells from $(i-1)$ to $(i+1)$-th cell, according to the dependency of the affected cells upon the i-th cell. Each variables in P is changed by their previous state. Similarly, a k cell shifting is obtained by applying the one cell shifting operation for k times upon the initial NHCA and symbolized as $f_i \xrightarrow{P} f_{i\pm k}$.

Nonlinearity Injection

Algorithm 2: NHCA synthesis

Input: A maximum length LHCA with ruleset $\mathcal{F}_L$, A position j to inject non-linearity and the set of cells of the LHCA $\mathcal{S}$

Output: A maximum length NHCA ruleset $\mathcal{F}_N$

1: $\mathcal{F}_N \leftarrow \mathcal{F}_L$
2: Let $\mathcal{F}_N = \{f_{n-1}, \cdots, f_0\}$
3: $\mathcal{X} \subset \mathcal{S} : \forall x \in \mathcal{X}, x \notin \mathbf{N}(j)$ ▷ select a subset from $\mathcal{S}$
4: $P \leftarrow \mathbf{f_N}(\mathcal{X})$ ▷ $\mathbf{f_N}$ is non-linear function
5: $f_j \leftarrow f_j \oplus P$
6: $(f_j \xrightarrow{P} f_{j+1})$ ▷ Apply shifting operation
7: $f_j \leftarrow f_j \oplus P$
8: **return** $\mathcal{F}_N$

Nonlinearity Injection

The maximum length property of the synthesized NHCA is established via following theorem.

Theorem 3

Algorithm 2 generates a maximum length NHCA.

Nonlinearity Injection

Table 3 : Maximum Length NHCA Synthesis from LHCA

Linear Rule		Nonlinear Rule
$f_0 = s_1$		$f_0 = s_1$
$f_1 = s_0 \oplus s_2$		$f_1 = s_0 \oplus s_2$
$f_2 = s_1 \oplus s_2 \oplus s_3$		$f_2 = (s_0\&s_4) \oplus (s_0\&s_5) \oplus (s_0\&s_6)\oplus$ $(s_2\&s_4) \oplus (s_2\&s_5) \oplus (s_2\&s_6) \oplus s_1 \oplus s_2 \oplus s_3$
$f_3 = s_2 \oplus s_4$	$\Longrightarrow$	$f_3 = (s_1\&s_5) \oplus s_2 \oplus s_4$
$f_4 = s_3 \oplus s_5$		$f_4 = (s_0\&s_4) \oplus (s_0\&s_5) \oplus (s_0\&s_6)\oplus$ $(s_2\&s_4) \oplus (s_2\&s_5) \oplus (s_2\&s_6) \oplus s_3 \oplus s_5$
$f_5 = s_4 \oplus s_5 \oplus s_6$		$f_5 = s_4 \oplus s_5 \oplus s_6$
$f_6 = s_5$		$f_6 = s_5$

Analysis

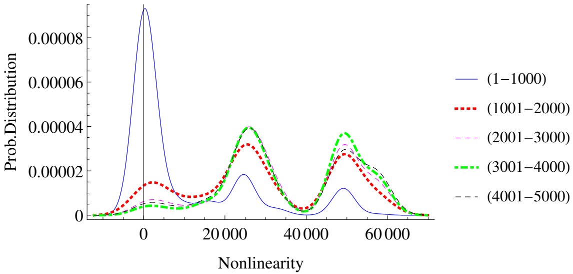
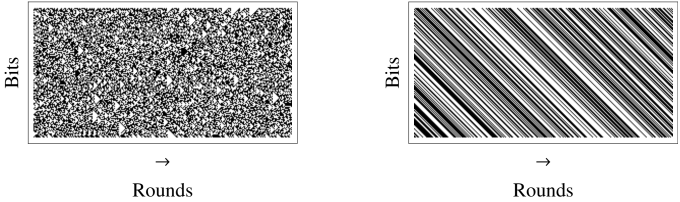


Figure 1 : Probability Density Function of Nonlinearity of a 17 bit NHCA

Table 4 : NIST Test Suite Result

Test Name	P-values	
	NHCA (127 bits)	NFSR (128 bits)
ApproximateEntropy	0.596753	0.000000
BlockFrequency	0.914133	1.000000
CumulativeSums(Forward)	0.135160	0.000000
CumulativeSums(Backward)	0.314951	0.000000
FFT	0.234132	0.000000
Frequency	0.171906	0.000000
LinearComplexity	0.187039	0.000000
LongestRun	0.914449	0.000000
NonOverlappingTemplate	0.441900	0.000000
OverlappingTemplate	0.219981	0.000000
Rank	0.381395	0.039176
Runs	0.194929	0.000000
Serial	0.930595	0.000000



(a) 254 iterations of 127 bit NHCA (b) 256 iterations of 128 bit NFSR

Figure 2 : Pictorial Comparison of NHCA and NFSR

Conclusion

Conclusion

- A scalable method for constructing maximum-length LHCA
- Nonlinearity is injected in LHCA retaining maximum periodicity
- Statistical results show suitability as a cryptographic primitive

Thank You