8-2-2011

# Novel pseudo random number generation using variant logic framework

Jeffrey Zheng
*Yunnan University, China*



Originally published in the Proceedings of the 2nd International Cyber Resilience Conference, Edith Cowan University, Perth Western Australia, 1st - 2nd August 2011

# NOVEL PSEUDO-RANDOM NUMBER GENERATION USING VARIANT LOGIC FRAMEWORK

Jeffrey Zheng
Department of Information Security, School of Software,
Yunnan University, China

conjugatesys@gmail.com

## Abstract

*Cyber Security requires cryptology for the basic protection. Among different ECRYPT technologies, stream cipher plays a central role in advanced network security applications; in addition, pseudo-random number generators are placed in the core position of the mechanism. In this paper, a novel method of pseudo-random number generation is proposed to take advantage of the large functional space described using variant logic, a new framework for binary logic. Using permutation and complementary operations on classical truth table to form relevant variant table, numbers can be selected from table entries having pseudo-random properties. A simple generation mechanism is described and shown and pseudo-random sequences are analyzed for their cycle property and complexity. Applying this novel method, it can play a useful role in future applications for higher performance of cyber security environments.*

**Keywords**

Pseudo Random Number Generation, Variant Logic, Cryptology

## INTRODUCTION

In advanced cyber environment, cyber security mechanism plays a guider role to protect secure information communicated and stored in network facilities (Robshaw, 1995 & Xiao, Li, Choi, 2004). To achieve adequate network security effects, cryptology has to be placed in the essential position (Robshaw, 1995). Different from block ciphers operate with a fixed transformation on a large blocks of plaintext; stream ciphers operate with a time-varying transformation on individual plaintext digits. Under the stream cipher methodology, Pseudo-Random Number Generator PRNG is placed in the central part of the mechanism.

From 2000-2003, New European Schemes for Signatures, Integrity and Encryption NESSIE were started (Nessie). During 2004-2008, another European stream cipher project: eSTREAM selected four software and three hardware schemes for ECRYPT Stream Ciphers (The eStream Project). Such extensive international activities on ECRYPT methodologies are showing the ultra-importance of Stream Cipher technologies in cyber environments for wider security applications.

From a cyber resilience viewpoint (Standaert, Malkin, Yung 2009 & Standaert, Pereira, Yung, 2010), a set of researchers are focussing attention on leakage resilient pseudorandom generator. This direction has shown interesting results to protect valuable information from side-channel attack aspects.

Since PRNG plays a key role in stream cipher applications and is the heart of cryptology (Angew, 1988, Atkinson, 1979, NIST, 2010). Many mathematical methodologies are applied to this field such as linear automata, cellular automata, Galois fields and other algebraic constructions (Atkinson, 1979, Matsumoto and Nishimura, 2000, Robshaw, 1995). In cryptology, Boolean logic operations are essential to create highly effective cryptology systems (Atkinson, 1979, Park and Miller, 1988, Santha and Vazirani, 1986) Robshaw, 1995) as binary logic generates the greatest efficiency through manipulation of only 1's and 0's. Therefore, it is advantageous to investigate potential mechanisms in binary logic due to the follow-on effect it has in cryptology.

## CLASSICAL LOGIC FUNCTION TABLE

A classic logic function in n variables can be represented as a truth table (Agnew, 1988, Atkinson, 1979). For a classic sequence in an ordinary number sequence, each table contains $2^n$ columns and $2^{2^n}$ rows with a total of $2^n \cdot 2^{2^n}$ bits respectively. An example of the standard truth table can be seen in Figure 1a.

## VARIANT LOGIC FUNCTION TABLE

Variant Logic construction is a new proposed theoretical structure (Zheng, Zheng, Kunii, 2011) to extend classical logic from the three basic operators: $\{\cap,\cup,\neg\}$. Two additional vector-operators: Permutation P and Complementary $\Delta$ are included with the original three to form the five basic operators within the novel framework. Let $S(N)$ denote a permutation group with *N* elements, then $S(N)$ contains a total of $N!$ permutation operators. Let $B_2^N = \{0,1\}^N$ denote a binary group with *N* elements, then $B_2^N$ contains a total of $2^N$ complementary operators.

The Permutation operator (P) and Complementary ($\Delta$) are two vector operators performed on each column vector of $2^{2^n}$ bits. For a given P and $\Delta$, two operators transforms the truth table into a variant table. Permutation operators changes positions of relevant columns but do not change their values. Complementary operators $\Delta$ do not change the position for each column, but may change entire values of the column. Two given operators can be performed together to generate a variant table for further usages. There are $2^n$ columns in the table as permutation elements, so this permutation group $S(2^n)$ contains a total of $2^n!$ permutation operators; and its complementary group $B_2^{2^n}$ includes a total of $2^{2^n}$ complementary operators. An example of the Variant Table can be seen in Figure 1b.

| N | $2^n-1$ | | i | | 0 |
|---|---|---|---|---|---|
| 0 | 0 | … | 0 | … | 0 |
| … | … | | … | | … |
| J | $J_{2n-1}$ | … | $J_i$ | … | $J_0$ |
| … | … | | … | | … |
| $2^{2n}-1$ | 1 | … | 1 | … | 1 |

(a) Truth Table Example

| $\Delta P(2^n-1)$ | | $\Delta P(i)$ | | $\Delta P(0)$ | K |
|---|---|---|---|---|---|
| $\Delta P(0_{2n-1})$ | .. | $\Delta P(0_i)$ | .. | $\Delta P(0_0)$ | $K_0$ |
| … | | … | | … | |
| $\Delta P(J_{2n-1})$ | .. | $\Delta P(J_i)$ | .. | $\Delta P(J_0$ | $K_J$ |
| … | | … | | … | |
| $\Delta P((2^{2n}-1)_{2n-1})$ | .. | $\Delta P((2^{2n}-1)_i)$ | .. | $\Delta P((2^{2n}-1)_0)$ | $K_{2^{2n}-1}$ |

(b) Variant Table Example

*Fig 1. n variable Truth Table and Variant Table under P and $\Delta$ operators*

## VARIANT METHOD OF PSEUDO-RANDOM NUMBER GENERATION

**Input:** $n, P, \Delta, m, L$ variables, $n \in N$, $P \in S(2^n)$, $\Delta, L, m \in B_2^{2^n}$

**Output:** $\{K_m, K_{m+1}, ...., K_{m+L-1}\}$ $L \cdot 2^n$ bits sequences

**Method:** The process for pseudo-random number generation can be seen in Figure 2:

n is the input variable number. Using n variables, a standard truth table can be constructed in $2^n$ columns and $2^{2^n}$ rows. *P* is a given permutation operator $P = (P_{2^n-1}...P_I...P_0)$, $P \in S(2^n)$, where $P_I$ corresponds to the *I*-th column. A given complementary operator $\Delta \in B_2^{2^n}$, $\Delta = (\Delta_{2^n-1}...\Delta_I...\Delta_0)$, $\Delta_I \in B_2$ that the operator is performed on the *I*-th column, where $\Delta_I = 0$, all values of the column are reversed and $\Delta_I = 1$, all values are invariant. $0 \le m < 2^{2^n}$ is an initial position for output sequences, from $K_m$, *L* conditions $\{K_{m+i}\}_{i=0}^{L-1}$ are output generated 0-1 bit sequences.

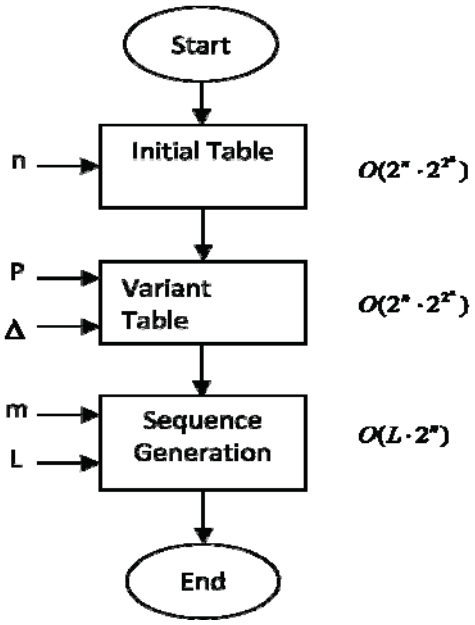


*Fig 2. Variant Method of Random Number Generation*

## SEQUENCE GENERATION EXAMPLE

For convenient understanding procedure, an example is selected to show in the n=2 case shown in Figure 3. Parameters are initialized to arbitrary values: n=2, P=(1203), $\Delta$=(0110)

After the table is generated, the pseudo-random sequence can read off the table. For m=4, L=6 conditions, a random number starting at position 4 of the variant table containing 6elements can be found:

*Fig 3. Example for Generation of Pseudo-Random Sequence*

## COMPLEXITY ANALYSIS

From an application viewpoint, it is important to have the exact complexity evaluation for the method. In the initial stage, it is necessary to manipulate $2^n$ columns and each column with $2^{2^n}$ rows; the total numbers of $2^n \cdot 2^{2^n}$ bits are required. The total complexity is of order $O(2^n \cdot 2^{2^n})$.

To generate Variant Table values, P operations need at least to manipulate bits once and $\Delta$ operations to manipulate the same number of bits. i.e. $O(2^n \cdot 2^{2^n})$.

Selecting $L \cdot 2^n$ bits from the variant table, it is necessary to perform $O(L \cdot 2^n)$ operations.

If a full table needs to be generated and keep the full table as a random resource, $O(2^n \cdot 2^{2^n})$ computational complexity is required. In general, their computational complexity is $O(L \cdot 2^n)$ - $O(2^n \cdot 2^{2^n})$ $0 < L < 2^{2^n}$.

Maximal cycle length: under this construction, the maximal length of the pseudo-random number sequence is $2^n \cdot 2^{2^n}$ bits. For any short sequences, the output sequence has a length less than this number. No clear cycle effects can be directly observed.

## CONCLUSION

It is important to design this new PRNG method to use variant logic construction. Since P and $\Delta$ potentially have a huge configuration space $2^n !\times 2^{2^n}$ times larger than classical Logic function spaces. Exploring how difficulties for this mechanism to be decoded will be the main issue for coming cryptologist's theoretical targets. In addition, it is important to understand what type of distribution will be relevant to this generation mechanism. Owing to intrinsic complexity of variant logic construction, this provides potential barriers to protect this type of sequences decoded directly.

Considering PRNG placed in the central part of stream cipher mechanism, and stream cipher technologies are more and more important in advanced network security environment, higher performance methodology and relevant implementation will be useful in this fields. Ongoing approaches will be focus on whether this mechanism to provide better PRNG methods to help different protections on side-channel attacks (Robshaw, 1995, Nessie, the eStream Project, gong, 2002, Xiao, Li, Choi, 2004, Aissa, Nouredine, 2009, Standaert, Malkin, Yung, 2009, Dwivedi, Tebbben, Harshavardhanna, 2010, Yu, Standaert, Pereira, Yunk, 2010) in wider network applications to resolve practical leakage-resilient issues in the future.

## REFERENCES

Agnew, G.B., (1988) "Random Source for Cryptographic Systems," *Advances in Cryptology | EUROCRYPT '87 Proceedings,* Springer-Verlag, pp. 77-81.

Aissa, B., and Nouredine, D., (2009) Designing resilient functions and bent function for stream ciphers. Georgian Electronic Scientific Journal: Computer Science and Telecommunication, No.1(18), 27-33

Atkinson, C., (1979) "A Family of Switching Algorithms for the Computer Generation of Beta Random Variables." Biometrika 66, no. 1: 141-145.

Davies, R., (2000) Hardware random number generators. *Int. 15th Australian Statistical Conference*, Jul..

Dwivedi, A., Tebben, D., and P. Harshavardhanna, P., (2010) Characterizing Cyber-Resiliency. The 2010 Military Communication Conference-Unclassified Program – Cyber Security and Network Management, IEEE press 1847-1852

Eastlake, D., Crocker, S.D. and Schiller, J.I., (1994) Randomness Requirements for Security," RFC 1750, Internet Engineering Task Force, Dec.

Gong, G., (2002) Cryptographic Properties of the Welch-Gong Transformation Sequence Generators, IEEE Trans. On Information Theory, Vol 48, N0.11, 2837-2846

Kachitvichyanukul, V., and Schmeiser, V.W. (1988) "Binomial Random Variate Generation." Communications of the ACM 31, no. 2: 216-223.

Matsumoto, M., and T. Nishimura, T., (2000) "Dynamic Creation of Pseudorandom Number Generators." In Proceedings of the Third International Conference on Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing: Monte Carlo and Quasi-Monte Carlo Methods 1998, 56-69.

NESSIE New European Schemes for Signatures, Integrity and Encryption https://www.cosic.esat.kuleuven.be/nessie/

NIST, (2010) "A statistical test suite for random and pseudorandom number generators for cryptographic applications," NIST Special Publication, 800-22.

Park, S.K., and Miller, K.W., (1988) " Random Number Generators: Good Ones Are Hard To Find", Communications of the ACM, October, pp. 1192-1201.

Robshaw, M., (1995) Stream Ciphers. RSA Laboratories Technical Report TR-701.

Santha, M. and Vazirani, U.V., (1986) "Generating Quasi-Random Sequences from Slightly Random Sources," *Journal of Computer and System Sciences,* v. 33, pp. 75-87.

Standaert, F.X., Malkin, T., and Yung, M., (2009) A unified framework for the analysis of side-channel key recovery attacks. EUROCRYPT, 443-461

The eSTREAM Project http://www.ecrypt.eu.org/stream/index.html

Xiao, Y., Li, H., and Choi, S., (2004) Protection and Guarantee for Voice and Video Traffic in IEEE 802.11e Wireless LANs, 11pages, IEEE INFOCOM

Yu, Y., Standaert, F.X., Pereira, O., and Yung, M., (2010) Practical Leakage-Resilient Pseudorandom Generator. CCS'2010, 141-151, ACM.

Zheng, J., Zheng, C. & Kunii, T.L., (2011) "A Framework Of Variant Logic Construction For Cellular Automata," InTech - Open Access Publisher, ISBN 978-953-307-172-5 http://www.intechopen.com/articles/show/title/a-framework-of-variant-logic-construction-for-cellular-automata

Zheng, J., & Zheng, C., (2010) "A Framework to express variant and invariant functional spaces for binary logic", Frontiers of Electrical and Electronic Engineering in China, Higher Education Press & Springer-Verlag. Vol.5 No.2, 163-173, http://www.springerlink.com/content/91474403127n446u/