Visual Maps of Variant Combinations on Random Sequences

Jeffrey Zheng and Jie Wan

Abstract Random sequences play the key role in network security applications. Randomness testing schemes are very important to ensure the randomness qualities for relevant sequences. This chapter proposes a visual scheme based on variant construction to measure sequences to intuitively show some combinatorial properties of key stream generated by stream ciphers. Basic models are described. This scheme provides a flexible framework for the variant measure method on the key stream of stream ciphers to describe randomness in various combinatorial maps.

Keywords Visual scheme · Variant measure · Combinatorial projection
Random sequence

1 Introduction

Random numbers play an important role in many network protocols and encryption schemas on various network security applications [1], for example, visual crypto, digital signatures, authentication protocols and stream ciphers. To determine whether a random sequence is suitable for a cryptographic application, the NIST has published a series of statistical tests as standards.

J. Zheng (✉)
Key Laboratory of Quantum Information of Yunnan, Yunnan University, Kunming, China
e-mail: conjugatelogic@yahoo.com

J. Zheng
Key Laboratory of Software Engineering of Yunnan, Yunnan University, Kunming, China

J. Wan
The People's Bank of China, Kunming, China
e-mail: wanjiech@163.com

J. Zheng (ed.), *Variant Construction from Theoretical Foundation to Applications*,
https://doi.org/10.1007/978-981-13-2282-2_22

In network security applications, the stream ciphers play a key role that have faster throughput and easier to implement compared to block ciphers [2]. RC4, the famous stream cipher, is suitable for large packets in Wireless LANs [3]. It has been used for encrypting the internet traffic in network protocols such as Sockets Layer (SSL), Transport Layer Security (TLS), Wi-Fi Protected Access (WPA), etc. [2].

eSTREAM project collected stream ciphers from international cryptology society [4] to promote the design of efficient and compact stream ciphers suitable for widespread adoptions. After a series of tests, algorithms submitted to eSTREAM are selected into two profiles. One is more suitable for software and another one is more suitable for hardware. Non-linear structures and recursive are playing the essential roles in new development.

Different visual schemes are required to test randomness of random sequences on different stream ciphers. Along this direction, this chapter proposes a flexible framework to handle a set of mete measurements on different combinatorial projections.

2 Variant Combinatorial Visualization

Architecture of variant visualization is shown in Fig. 1.

The variant visualization architecture is separated into four core components: EAC, SCC CC and VC.

- RGC Randomness Generate Component generate a random sequence;
- VSC Variant Statistic Component handles the statistic process using the variant measure method [5];
- CC Combinatorial Component chooses combinations;
- VC Visualization Component makes visualization based on SCC measures and CC vectors.

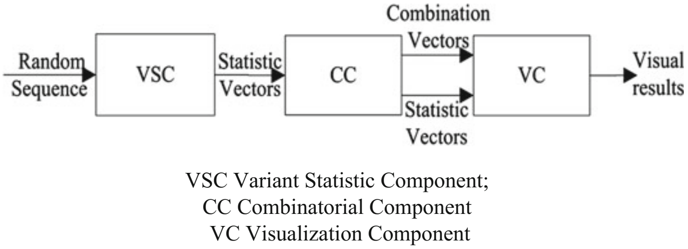

Random
Sequence
VSC
Statistic
Vectors
CC
Combination
Vectors
Statistic
Vectors
VC
Visual
results
VSC Variant Statistic Component;
CC Combinatorial Component
VC Visualization Component

Fig. 1 Visualization architecture

The input n is the length of the binary sequence. The stream ciphers could be changed to any stream cipher that can generate binary sequence. This section focuses on the variant measure module and the visual method module.

A visual example of RC4 will be described in Sect. 2.5.

2.1 Variant Logic Framework

The variant logic framework has been proposed in [6]. Li [7] used the variant measure method to generate different symmetry results [5] based on cellular automata schemes [8]. Under such construction, even some random sequences show symmetry properties in distributions.

Under variant construction, the variant conversion operator can be defined as follows:

$$C(x, y) = \begin{cases} \perp, x = 0, y = 0 \\ +, x = 0, y = 1 \\ -, x = 1, y = 0 \\ \top, x = 1, y = 1 \end{cases} \tag{1}$$

It is convenient to list relevant variant logic variables shown in Table 1.

In the variant measure method, each sequence is converting from binary sequence to probability which generated by counting the number of each variable in $\{\perp, +, -, \top\}$ and computes the probability of each variable. The measurement method is shown in Table 1.

Table 1 The variant measure method

(a) Counting method			(b) Probability computing	
Variant variable	Number of type	Total number	Measure parameters	Number of type
$\perp$	$N_\perp$	$N = N_\perp + N_\top + N_+ + N_-$	$P_\perp$	$N_\perp/N$
$\top$	$N_\top$		$P_\top$	$N_\top/N$
$+$	N_+		P_+	N_+/N
$-$	N_-		P_-	N_-/N

The variant measure method provides a set of results in measures of different 0–1 sequences. The following mechanism can transfer stream cipher sequences as relevant measures.

The essential models of variant scheme are described as follows.

2.2 VSC Variant Statistic Component

The VSC component converts the binary sequence to variant sequence in VCM module, and to compute probabilities and entropies in PECM module, respectively. The component is shown in Fig. 2.

VCM Variant Conversion Module

VCM module transfers input binary sequences by following steps:

Step 1. Generate an n bit binary sequence $S = S_1 S_2 S_3 \ldots S_n$ by a stream cipher.

Step 2. Shift X to left by M bit (M is the length of shifting) and generate a new binary sequence $S' = S'_1 S'_2 S'_3 \ldots S'_{n-M} = S_{1+M} S_{2+M} \ldots S_n$.

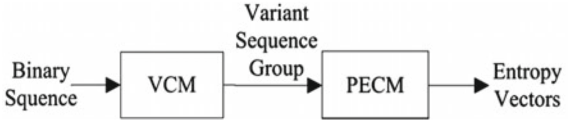
Step 3. Convert two sequences: S and S' to a variant sequence $V = V_i = C\left(S_{i,} S'_i\right), \quad i = 1, 2, 3 \ldots (n - M)$.

Step 4. Separate V into n/N parts. N is the length of each part and $M \leq N \leq n$ to form a set of variant sequence groups

$$\begin{aligned} G &= \left\{G_1, G_2, \ldots, G_{n/N}\right\} \\ &= \{\{V_1, V_2, \ldots, V_N\}, \ldots, \{V_{n-N}, V_{n-N+1}, \ldots, V_n\}\} \end{aligned}$$

Step 5. Separate each item in G into N/M parts to establish a sequence group

$$\begin{aligned} G = \{\{\{V_1, \ldots, V_M\}, \ldots, \{V_{N-M+1}, \ldots, V_N\}\}, \ldots, \\ \{\{V_{n-N}, \ldots, V_{n-N+M}\}, \ldots, \{V_{n-M}, \ldots, V_n\}\}\} \end{aligned}$$



VCM Variant Conversion Module
PECM Probability and Entropy Computing Module

Fig. 2 Variant statistic component

PECM Probability and Entropy Computing Module

PECM converts a variant sequences group to separate it into several parts to compute probability and entropies. The equations computing the parameters have been described in Table 1. The main steps are performed as follows:

Step 6. Compute the probability vector $P = \{P_{\perp}, P_{+}, P_{-}, P_{\top}\}$ of each part in G';

Step 7. Calculate the distribute probability vector $D = \{D_{\perp}, D_{+}, D_{-}, D_{\top}\}$ of each part in G based on P vector;

Step 8. Evaluate the entropy vector $\{E_{\perp}, E_{+}, E_{-}, E_{\top}\}$ from the D vector.

2.3 CC Combinatorial Component

IIn the CC component, it can be separated into two modules. One is SM module to form the vector selecting and another one is VDM module to perform the visualization.

Visual data is a set of E vectors as input for VC. For E vector, choose a projection as a visual vector to compute the visual result from E vectors. So there will be 16 visual results.

Base on the same number of variables in a combination, the combination set can be integrated into 5 parts. i.e. The selected number of variables in the combination is in 0-4.

Let the classification be $EC = \{EC_0, EC_1, EC_2, EC_3, EC_4\}$. Since the EC_0 is empty, it can be ignored. Only four distributions are of concern in Sect. 2.4.

2.4 Visualization Component

According to the variant measure method, in the rectangular axis, let $E_{\perp}$ be the positive axis of X, $E_{\top}$ be the negative axis of X, E_{+} the positive axis of Y, E_{-} be the negative axis of Y. The axis is shown in Fig. 3.

For $EC_1 = \{\{E_{\perp}\}, \{E_{+}\}, \{E_{-}\}, \{E_{\top}\}\}$, points are distributed to the axis.

For $EC_2 = \{\{E_{\perp}, E_{+}\}, \{E_{\perp}, E_{-}\}, \{E_{\perp}, E_{\top}\}, \{E_{+}, E_{-}\}, \{E_{+}, E_{\top}\}, \{E_{-}, E_{\top}\}\}$, points are distributed in the shadow area in Fig. 4.

For $EC_3 = \{\{E_{\perp}, E_{+}, E_{-}\}, \{E_{\perp}, E_{+}, E_{\top}\}, \{E_{\perp}, E_{-}, E_{\top}\}, \{E_{+}, E_{-}, E_{\top}\}\}$, points are distributed in the area of EC_1 and the area of EC_2.

For $EC_4 = \{\{E_{\perp}, E_{+}, E_{-}, E_{\top}\}\}$, points are distributed in Fig. 5.

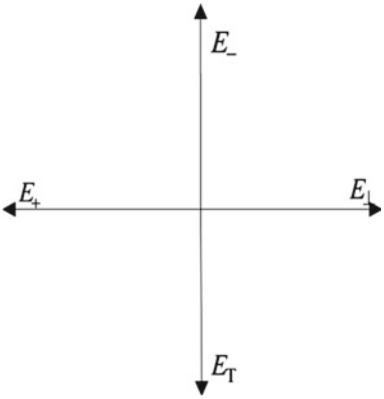


Fig. 3 Visualization axis

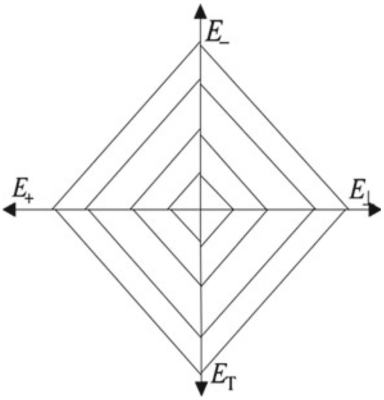


Fig. 4 Distribution areas of EC_2

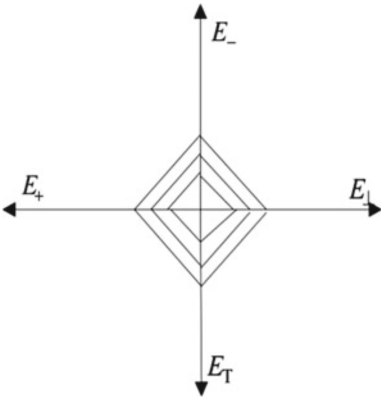


Fig. 5 Distribution areas of EC_4

2.5 Example

An example is given step by step to show how the algorithm runs. In the example, n, N and M are, respectively, assigned to 40, 16 and 8.

Step 1. Input a 35 bit binary sequence, {0101001011101011001011010111101101010101}

Step 2. Generates S', {1110101100101101011101101010101}.

Step 3. Generates V, $\{+\top+-+\bot\top+--\top\bot\top+-\top\bot+\top+\top+-\top\bot\top-\top-+-\top\}$.

Step 4. Separate V into a G vector. The G vector is $\{\{+\top+-+\bot\top+--\top\bot\top+-\top\},\{\bot+\top+\top+-\top\bot\top-\top-+-\top\}\}$.

Step 5. Separate the G into the G' vector. The G' vector in the example is $\{\{+\top+-+\bot\top+,--\top\bot\top+-\top\},\{\bot+\top+\top+-\top,\bot\top-\top-+-\top\}\}$.

Step 6. Generate probability vector P of each sequence in G'. The P vector of $\{+\top+-+\bot\top+\}$ is $\{P_\bot = 0.125,\ P_+ = 0.5,\ P_- = 0.125,\ P_\top = 0.25\}$.

Step 7. Compute the distribute probability vector D of each sequence in G from P. The D vector of $\{+\top+-+\bot\top+,--\top\bot\top+-\top\}$ is shown in Fig. 6.

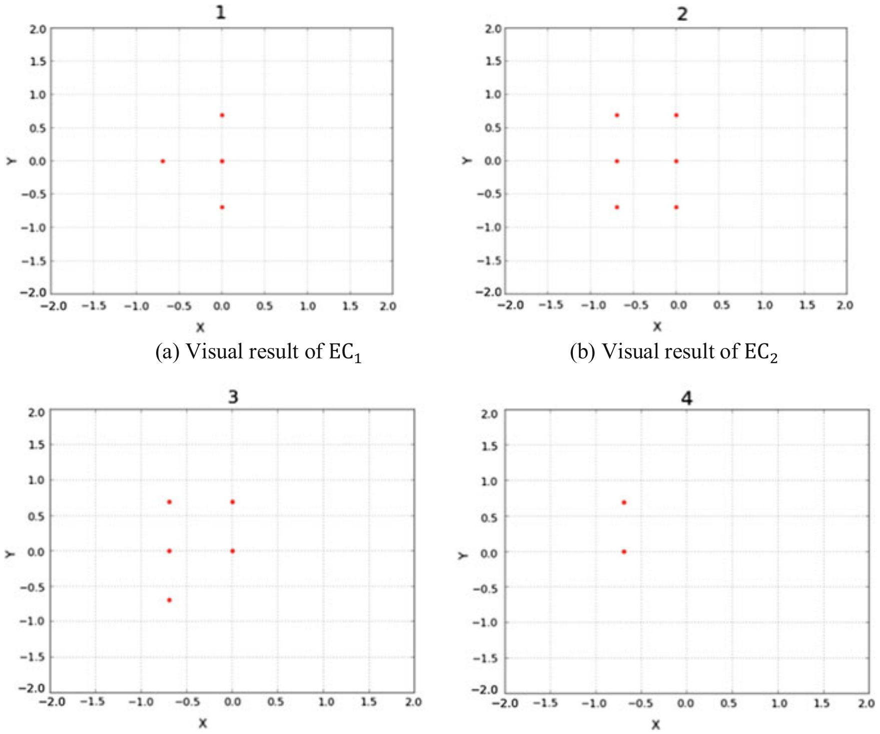
Step 8. Compute the entropy vector E of each sequence in G from D. The E vector of $\{+\top+-+\bot\top+--\top\bot\top+-\top\}$ is shown in Fig. 7.

$$\left\{\begin{array}{l} D_\bot = \{P_{0.125} = 1 \qquad\qquad\qquad \} \\ \qquad\qquad\qquad \vdots \\ \qquad\qquad\qquad \vdots \\ D_\top = \{P_{0.25} = 0.5, P_{0.725} = 0.5\} \end{array}\right.$$

Fig. 6 *D* vectors of $\{+\top+-+\bot\top+,--\top\bot\top+-\top\}$

$$\left\{\begin{array}{l} E_\bot = \quad (P_{0.125}\log P_{0.125} \qquad\qquad\qquad\qquad) = 0.0 \\ \qquad\qquad\qquad\qquad\qquad \vdots \\ \qquad\qquad\qquad\qquad\qquad \vdots \\ E_\top = -(P_{0.25}\log P_{0.25} + P_{0.725}\log P_{0.725}) = 0.693147 \end{array}\right.$$

Fig. 7 *E* vectors of $\{+\top+-+\bot\top+--\top\bot\top+-\top\}$



(a) Visual result of EC_1

(b) Visual result of EC_2

(c) Visual result of EC_3

(d) Visual result of EC_4

Fig. 8 Visual result of the example

Step 9. Compute visual results from E vectors. In the E vectors of $\{+\top+-+\bot\top+--\top\bot\top+-\top\}$. If the selection is $\{E_\bot\}$, points will be (0.0, 0.0). If the selection is $\{E_\bot, E_\top\}$, points will be (0.0, −0.693147) and (0.0, 0.0). If the selection is $\{E_\top, E_-\}$, points will be $\{E_- - |E_\top|\} =$ (0.0, 0.0) and (0.0, 0.693147). If the selection is $\{E_\bot, E_\top, E_-\}$, points will be $\{E_\bot, E_- - |E_\top|\} =$ (0.0, 0.0) and (0.0, 0.693147).

Step 10. Separate visual results to EC classification. Visual results of the G in the example are shown in Fig. 8.

3 Result

3.1 Visual Result of RC4

The initial: {**n** : 128,000, **N** : 128, **M** : 16}

The visual result (Fig. 9).

The initial: {**n** : 128,000, **N** : 128, **M** : 24}

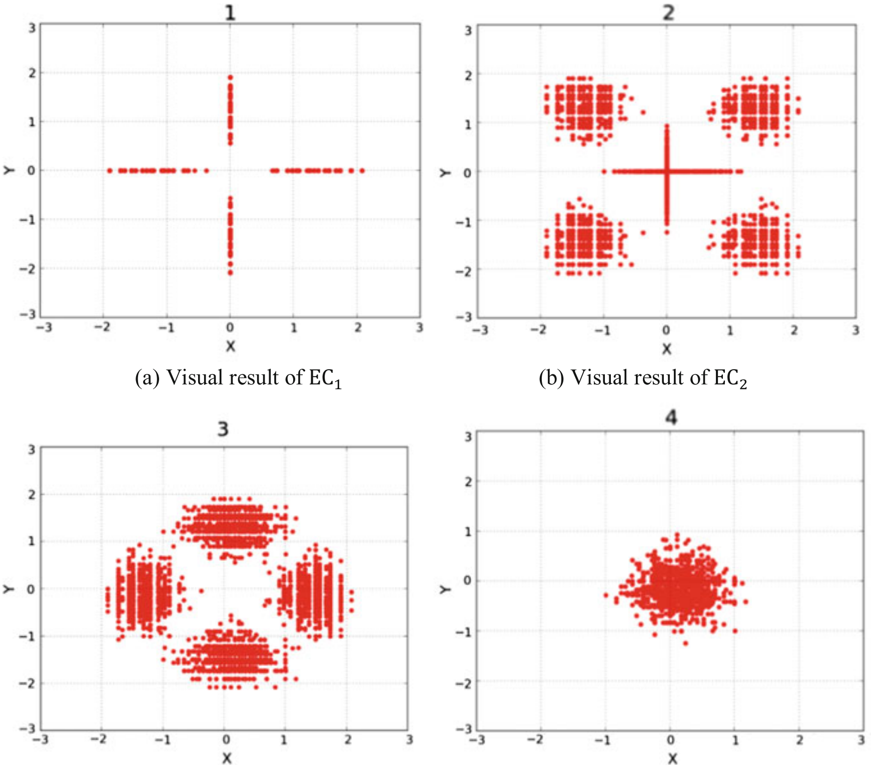
The visual result (Fig. 10).

The initial: {**n** : 128,000, **N** : 1000, **M** : 8}

The visual result (Fig. 11).

The initial: {**n** : 100,000, **N** : 100, **M** : 24}

The visual result (Fig. 12).



(a) Visual result of EC_1

(b) Visual result of EC_2

(c) Visual result of EC_3

(d) Visual result of EC_4

Fig. 9 Visual result of RC4 {**n** : 128000, **N** : 128, **M** : 16}

3.2 *Visual Result of HC256*

The initial: {**n** : 128,000, **N** : 128, **M** : 16}

The visual result (Fig. 13).

The initial: {**n** : 128,000, **N** : 128, **M** : 24}

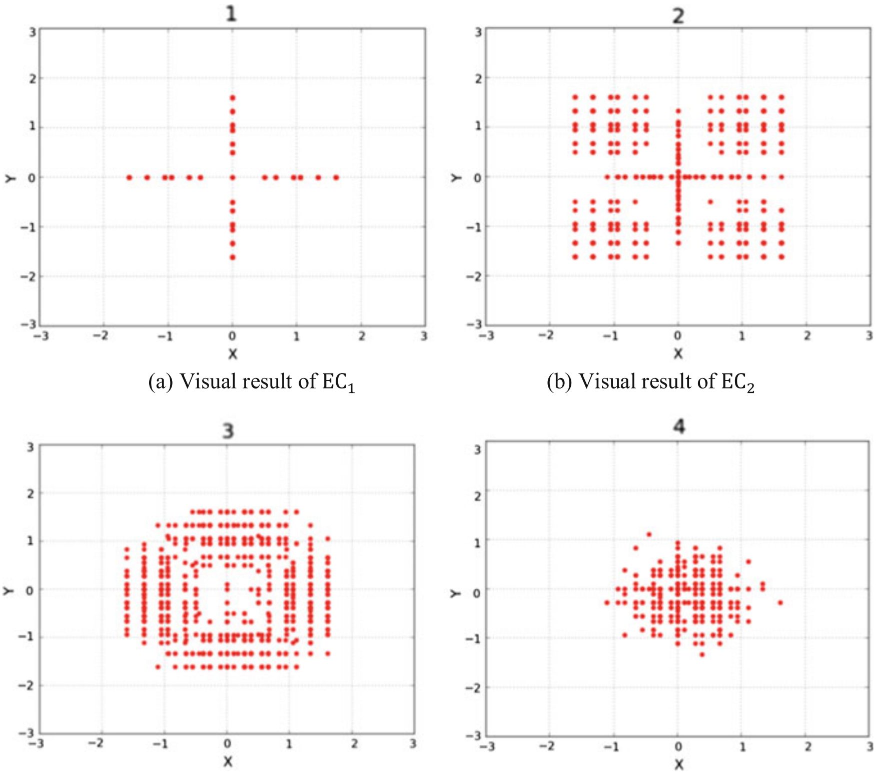
The visual result (Fig. 14).

The initial: {**n** : 100,000, **N** : 100, **M** : 8}

The visual result (Fig. 15).

The initial: {**n** : 100,000, **N** : 100, **M** : 16}

The visual result: (Fig. 16).



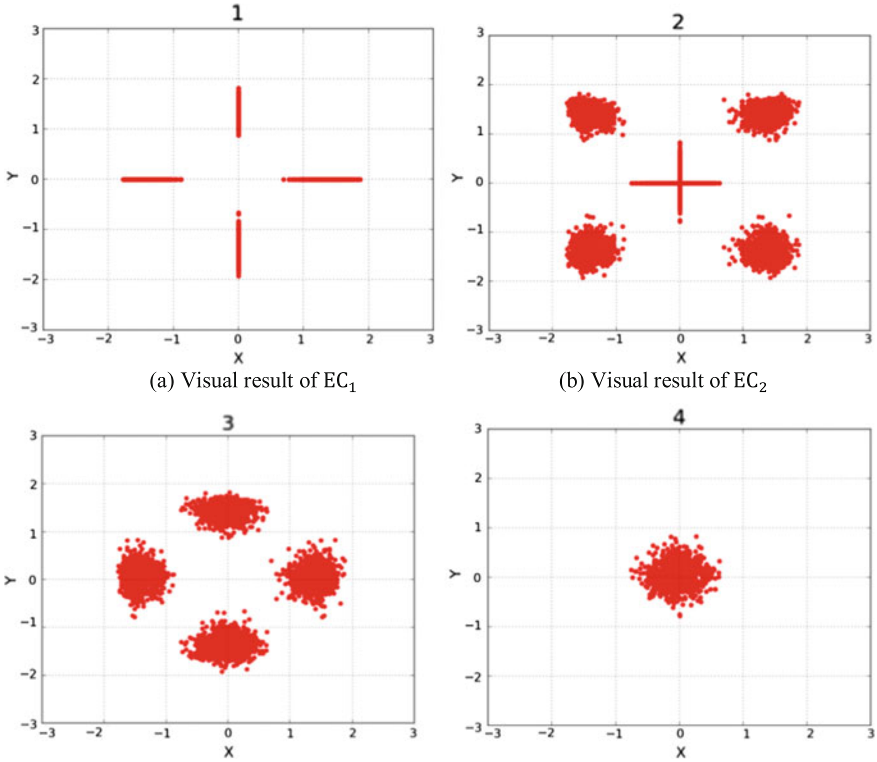
(a) Visual result of EC$_1$

(b) Visual result of EC$_2$

(c) Visual result of EC$_3$

(d) Visual result of EC$_4$

Fig. 10 Visual result of RC4 {**n** : 128000, **N** : 128, **M** : 24}



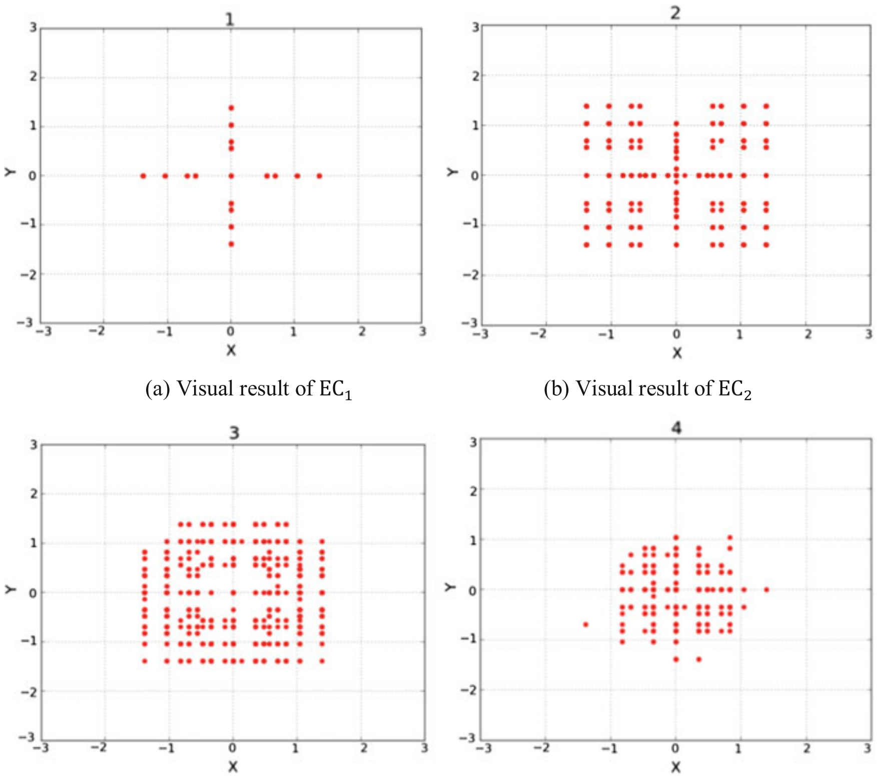
(a) Visual result of EC$_1$

(b) Visual result of EC$_2$

(c) Visual result of EC$_3$

(d) Visual result of EC$_4$

Fig. 11 Visual result of RC4 {**n** : 128000, **N** : 1000, **M** : 8}



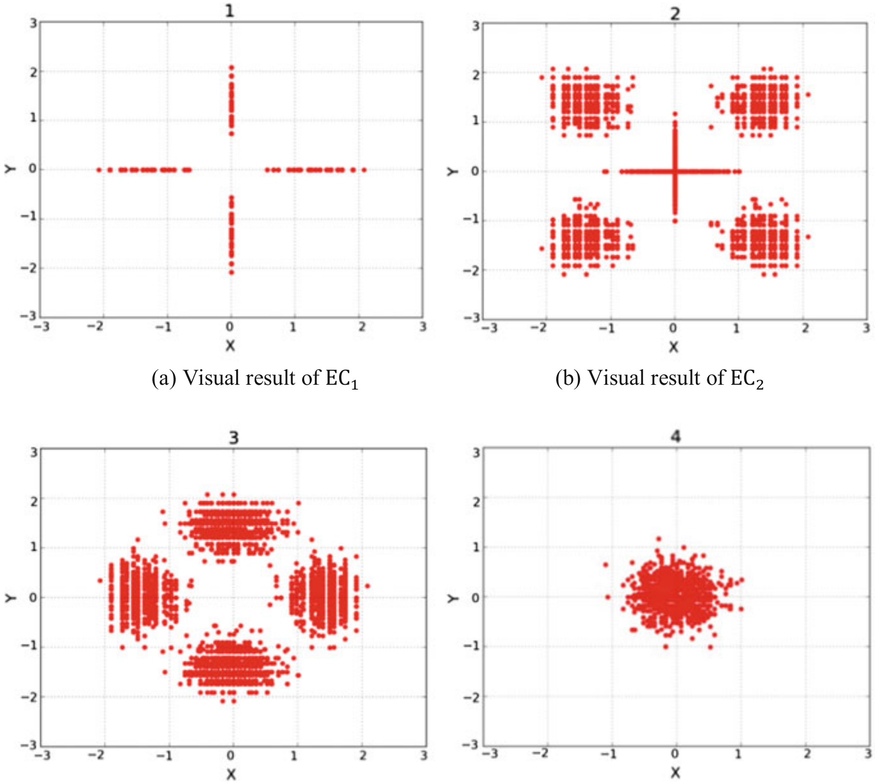
(a) Visual result of EC_1

(b) Visual result of EC_2

(c) Visual result of EC_3

(d) Visual result of EC_4

Fig. 12 Visual result of RC4 {**n** : 100000, **N** : 100, **M** : 24}



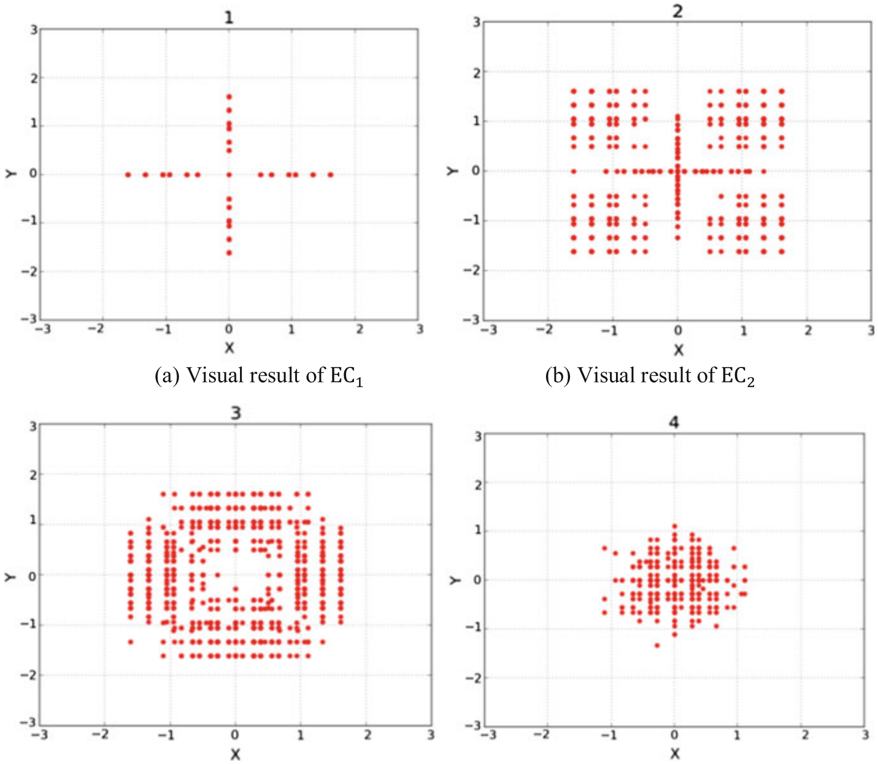
(a) Visual result of EC_1

(b) Visual result of EC_2

(c) Visual result of EC_3

(d) Visual result of EC_4

Fig. 13 Visual result of HC256 {**n** : 128000, **N** : 128, **M** : 16}



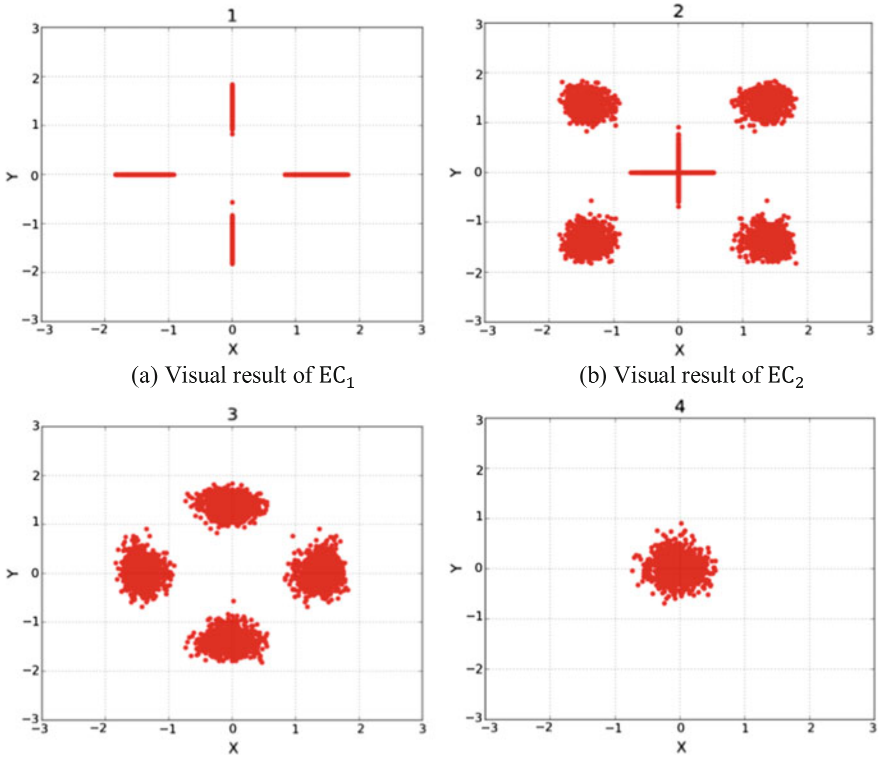
(a) Visual result of EC_1

(b) Visual result of EC_2

(c) Visual result of EC_3

(d) Visual result of EC_4

Fig. 14 Visual result of HC256 {**n** : 128000, **N** : 128, **M** : 24}



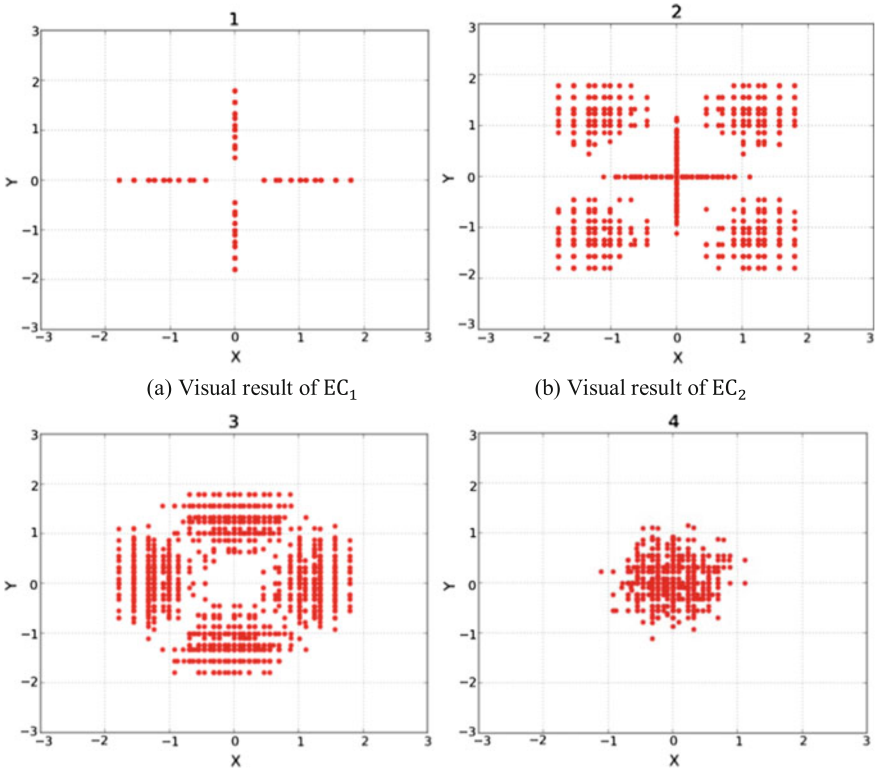
(a) Visual result of EC_1

(b) Visual result of EC_2

(c) Visual result of EC_3

(d) Visual result of EC_4

Fig. 15 Visual result of HC256 {**n** : 100000, **N** : 100, **M** : 8}



(a) Visual result of EC_1

(b) Visual result of EC_2

(c) Visual result of EC_3

(d) Visual result of EC_4

Fig. 16 Visual result of HC256 {**n** : 100000, **N** : 100, **M** : 16}

4 Conclusion

The visual results show the similar symmetry property of sequences generated by RC4 and HC256. They are showing interesting distributions and can be significantly distinguished from their combinatorial maps. From our models and illustrations, various maps can be integrated by their combinatorial projections to show different spatial distributions on random sequences. Under this configuration, the variant measure method provides a new analysis tool for stream cipher applications in further explorations.

Acknowledgements This work was supported by the Key Project on Electric Information and Next Generation IT Technology of Yunnan (2018ZI002), NSF of China (61362014) and Yunnan Advanced Overseas Scholar Project.

References

1. S. William, *Cryptography and Network Security: Principles and Practice* (Pearson Education, 2006)
2. G. Paul, S. Maitra, *RC4 Stream Cipher and Its Variants* (CRC Press, 2012)
3. P. Prasithsangaree, P. Krishnamurthy, Analysis of energy consumption of RC4 and AES algorithms in wireless LANs, in *Global Telecommunications Conference, 2003*. GLOBECOM ‘03. IEEE, vol. 3, pp. 1445–1449 (2003)
4. eSTREAM project, http://www.ecrypt.eu.org/stream/
5. J. Zheng, C. Zheng, T. Kunii, Interactive maps on variant phase spaces- from measurements—micro ensembles to ensemble matrices on statistical mechanics of particle models, in *Emerging Applications of Cellular Automata*, pp. 113–196, CC BY (2013)
6. J.Z.J. Zheng, C.H. Zheng, A framework to express variant and invariant functional spaces for binary logic. Front. Electr. Election. **5**(2), 163–172 (2010)
7. Q. Li, Z. Zheng. Spacial distributions for measures of random sequences using 2D conjugate maps, in *Proceedings of Asia-Pacific Youth Conference on Communication (APYCC)* (2010)
8. S. Wolfram, Theory and applications of cellular automata. Scientific (1986)