Hans Journal of Data Mining 数据挖掘, 2014, 4, 9-17
Published Online April 2014 in Hans. http://www.hanspub.org/journal/hjdm
http://dx.doi.org/10.12677/hjdm.2014.42002

Hans汉斯

Distinguish HC256, RC4 Sequence Using Systematic Measurement Mechanism

Ruoyu Shen[1*], Jeffrey Zheng[2]

[1]Department of Information Security, School of Software, Yunnan University, Kunming
[2]Key Lab of Yunnan University Software Engineering, Kunming
Email: *lansry@sina.cn

Received: Apr. 2nd, 2014; revised: Apr. 10th, 2014; accepted: Apr. 16th, 2014

Abstract

Stream cipher was widely used and the random number tests of the key stream play a key role in information security. Unlike usual NIST random number test for the random sequence with selected length, this article was focused on splitting the random sequence to do the NIST random number test. Then, the frequency distribution was measured through using the test results. Finally, two-dimensional feature map was generated. The selected test data show that such feature maps distinguish the key streams generated by HC256 and RC4 through the feature maps explicitly.

Keywords

Stream Cipher, NIST Random Number Test, Visualization, Feature Distribution, Distinction

利用系统化测量机制区分HC256，RC4序列

沈若愚[1*]，郑智捷[2]

[1]云南大学软件学院信息安全系，昆明
[2]云南大学软件工程重点实验室，昆明
Email: *lansry@sina.cn

收稿日期：2014年4月2日；修回日期：2014年4月10日；录用日期：2014年4月16日

*通讯作者。

摘　要

流密码应用广泛，对密钥流进行随机性测试在信息安全领域有重要作用。有别于对选定长度的随机序列利用NIST统计包进行随机性检测，本文针对HC256，RC4密钥流多重分段利用NIST提供的测试方法进行系统随机性检测，并对测量结果进行频率分布统计，形成二维可视化特征分布。选定的测试数据显示，该类特征分布图对HC256和RC4生成的密钥流区分明确。

关键词

流密码，NIST随机数测试，可视化，特征分布，区分

1. 引言

在现代密码学前沿应用中，流密码的设计与分析占据重要的位置。RC4[1]是一个应用广泛的流密码算法，从 eSTREAM 项目[2](ECRYPT Stream Cipher Project)中选出的 HC256[3]显示出潜在应用价值。

在目前的应用环境中，对不同流密码算法生成的 01 序列进行 NIST(National Institute of Standards and Technology)随机性测试[4]在密码学、安全领域起着重要的作用。当前的 NIST 检测包直接提供针对选定序列的测量功能，如何利用该类模型和方法做系统推广，是复杂应用环境需要重视的一类问题。

本文致力于比较不同产生机制下生成的二维可视化特征分布，对流密码算法 HC256 和 RC4 生成的 01 序列进行区分，并对 HC256 和 RC4 内部结构相似这一特征进行进一步探究，为后续的分析提供参考。首先对流密码算法生成的 01 序列分段进行 NIST 随机性检测，然后对测得的 *PValue* 值进行频率分布统计得到 P_i，最后对 P_i 进行模幂运算投影，生成可视化特征分布图。

1.1. 流密码算法

RC4 是 Ron Rivest 为 RSA 公司在 1987 年设计的一种流密码[1]。RC4 算法以随机置换作为基础，是一个密钥长度可变，面向字节流的流密码。已被应用于多种数据传输和网络协议中。

HC-256[3]是欧洲流密码计划(eSTREAM)征集到的面向软件实现的快速同步流密码。HC-256 借鉴了 RC4 的思想，同时引入了面向字节的非线性函数来更新系统的内部状态[2]。由于 HC256 的内部结构和 RC4 结构类似，本文选取这两种算法进行分析比较。

1.2. NIST 随机数检测标准

序列的随机性测试一直是信息安全领域重要的研究方向。美国 NIST SP800-22 测试标准[4]包括 15 种测试手段。有别于常用的对选定长度的随机序列进行 NIST 随机性检测，本文对密钥流分段进行 NIST 随机性检测以寻求随机序列段与段之间的内在联系与区别，关注于流密码算法在同一条件下生成的 01 序列内部分段之间的关联。

1.3. 聚类分析

聚类分析是数据挖掘的一项重要的方法。聚类问题实际上是将一组数据分成若干个组，在这些组之间寻找数据之间内在的联系[5]。

随着流密码在信息安全领域的深入应用，对于流密码的分析方法也在不断发展。本文没有采用典型的流密码分析方法[6]，尝试对生成的密钥流进行多元统计分析，对数据进行分组，为聚类分析提供前期

分类图形，为探索流密码内部机制之间的联系与差异，以及对密码序列之间的区分提供参考。

1.4. 区分攻击

区分攻击是一种灵活有效的密码分析方法，其基本思想是通过观察某些输入与输出比特之间的关系来判别这些比特是来自真随机源还是来自密码[7]。

区分攻击的关键是寻找适当的区分器，不同于已有的区分攻击算法，本文致力于分析不同流密码算法生成的比特流，利用输出的二维可视化特征分布图对 HC256 和 RC4 流密码算法生成的 01 序列进行区分。该方法从另一个角度为密码分析研究提供一定的参考。

2. 方法

2.1. 系统总体架构

该系统可分为 4 个子模块，如图 1 所示：

1) 密钥流生成：流密码算法 (如：HC256，RC4 等) 生成 1、0 序列；

2) NIST 随机性检测：用 NIST 方法(如：块内频数检验，频率检测，二元矩阵秩检验，离散傅立叶变换检验等)分段(4000 bit/组，1000 bit/组…)计算 *PValue* 值；

3) 频率分布直方图：对 *PValue* 值分组(25 个/组，50 个/组…)，画直方图(直方图区间数：2, 4, 8, 10, 16…)统计概率 P_i 值；

4) 可视化：对 P_i 进行模幂运算或者对数运算 + 模幂运算，每张直方图得到一个二维坐标对 $\{X_i, Y_i\}$，然后投影画图。

打点画图时 X 轴 Y 轴的坐标公式：(n：模幂运算的指数 $n = 2,3,4,5,\cdots$, j：概率分布直方图的区间数目 $j = 2,4,8,\cdots$)

模幂运算：

X 轴：$\sqrt[n]{\sum_{i=0}^{i=j} P_i^n}$

Y 轴：$\left(\sum_{i=0}^{i=j} \sqrt[n]{P_i}\right)^n$

对数运算 + 模幂运算：

X 轴：$\sqrt[n]{\sum_{i=0}^{i=j} \left(-P_i \ln P_i\right)^n}$

Y 轴：$\left(\sum_{i=0}^{i=j} \sqrt[n]{-P_i \ln P_i}\right)^n$

输入：

Key 表示流密码算法生成 01 序列时，需要输入的初始密钥。 $Key \in \{\text{十进制数}|_{Mode1=H}, \text{字母}|_{Mode1=R}\}$

IV 表示流密码算法生成 01 序列时，需要输入的初始向量。 $IV \in \{\text{十进制数}|_{Mode1=H}, \text{不输入}|_{Mode1=R}\}$

*Mode*1 表示不同的流密码算法。 $Mode1 \in \{H, R\}$，*Mode*1 = *H* 表示选用的算法是 HC256，*Mode*1 = *R* 表示选用的算法是 RC4。

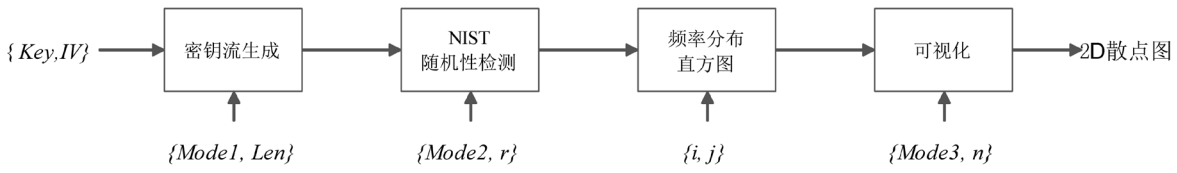


Figure 1. Architecture of the system
图 1. 系统总体架构

LEN 表示生成的 01 序列的长度，$LEN > 0$。

*Mode*2 表示不同的 NIST 随机性检测算法。$Mode2 \in \{1,2,3,4\}$，*Mode*2 = 1 表示选用的算法是块内频数检验，*Mode*2 = 2 表示选用的算法是频率检测，*Mode*2 = 3 表示选用的算法是二元矩阵秩检验，*Mode*2 = 4 表示选用的算法是离散傅立叶变换检验。

r 表示 01 序列的分段长度。

$$r \in \left\{ r \geq 100 \middle|_{Mode2=1,2}, r \geq 38 \times \text{矩阵行数} \times \text{矩阵列数} \middle|_{Mode2=3}, r \geq 1000 \middle|_{Mode2=4} \right\}$$

i 表示统计频率分布直方图时，*PValue* 序列的分段长度，$i > 0$。

j 表示统计频率分布直方图时，直方图的区间数，$j > 0$。

*Mode*3 表示不同的横纵坐标计算方法。$Mode3 \in \{1,2\}$，*Mode*3 = 1 表示选用的算法是模幂运算，*Mode*3 = 2 表示选用的算法是对数运算 + 模幂运算。

n 横纵坐标计算方法中模幂运算的指数，$n > 0$。

输出：

2D 散点图二维可视化特征分布图。

2.2. 密钥流生成

见图 2。

输入：

*Mode*1 表示不同的流密码算法。$Mode1 \in \{H, R\}$，*Mode*1 = *H* 表示选用的算法是 HC256，*Mode*1 = *R* 表示选用的算法是 RC4。

LEN 表示生成的 01 序列的长度，$LEN > 0$。

Key 表示流密码算法生成 01 序列时，需要输入的初始密钥。$Key \in \left\{ \text{十进制数} \middle|_{Mode1=H}, \text{字母} \middle|_{Mode1=R} \right\}$

IV 表示流密码算法生成 01 序列时，需要输入的初始向量。$IV \in \left\{ \text{十进制数} \middle|_{Mode1=H}, \text{不输入} \middle|_{Mode1=R} \right\}$

输出：

KeyStream 01 序列。

2.3. NIST 随机性检测

见图 3。

输入：

*Key*Stream 01 序列。

*Mode*2 表示不同的 NIST 随机性检测算法。$Mode2 \in \{1,2,3,4\}$，*Mode*2 = 1 表示选用的算法是块内频数检验，*Mode*2 = 2 表示选用的算法是频率检测，*Mode*2 = 3 表示选用的算法是二元矩阵秩检验，*Mode*2 = 4 表示选用的算法是离散傅立叶变换检验。

r 表示 01 序列的分段长度。

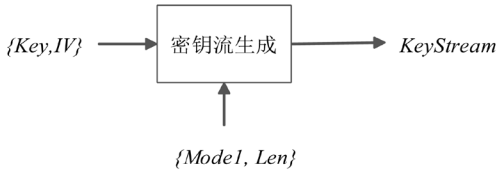


Figure 2. Architecture of key generation module
图 2. 密钥流生成子模块

$$r \in \left\{ r \geq 100 \Big|_{Mode2=1,2}, r \geq 38 \times \text{矩阵行数} \times \text{矩阵列数} \Big|_{Mode2=3}, r \geq 1000 \Big|_{Mode2=4} \right\}$$

输出：

PValuePValue 序列， $0 \leq PValue \leq 1$。

2.4. 频率分布直方图

见图 4。

输入：

PValuePValue 序列， $0 \leq PValue \leq 1$。

i 表示统计频率分布直方图时，*PValue* 序列的分段长度，$i > 0$。

j 表示统计频率分布直方图时，直方图的区间数，$j > 0$。

输出：

P_i 多组 P_i 值， $0 \leq P_i \leq 1$。

2.5. 可视化

见图 5。

输入：

P_i 多组 P_i 值， $0 \leq P_i \leq 1$。

*Mode*3 表示不同的横纵坐标计算方法。$Mode3 \in \{1,2\}$，*Mod*3 = 1 表示选用的算法是模幂运算，*Mode*3

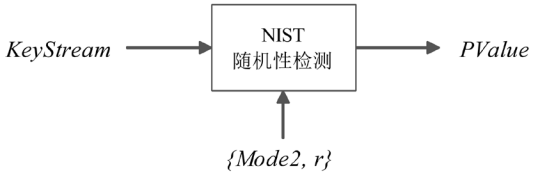


Figure 3. Architecture of NIST random number testing module
图 3. NIST 随机性检测子模块

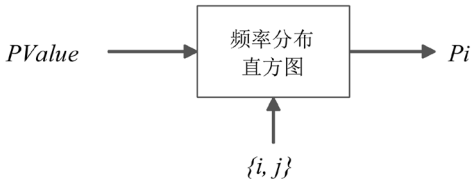


Figure 4. Architecture of Frequency Distribution Histogram Module
图 4. 频率分布直方图子模块

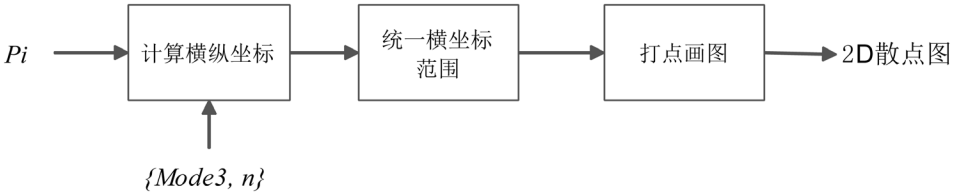


Figure 5. Architecture of visualization module
图 5. 可视化子模块

= 2 表示选用的算法是对数运算 + 模幂运算。

n 横纵坐标计算方法中模幂运算的指数，$n > 0$。

输出：

2D 散点图二维可视化特征分布图。

2.6. 各模块数据说明

从系统的架构来看，得到最终的 2D 散点图以前会生成三种中间数据，包括 01 序列、*PValue* 序列、P_i 序列，下面对这三种中间数据以及系统变量的取值进行说明。

2.6.1. 密钥流生成

为了使二维散点图的点数足够多，所以生成的 01 序列要足够长。本文 01 序列的长度(*LEN*)取值 102,400,000，最终可得到 10,240 个点。为了便于比较分析，通过改变初始密钥 *Key* 和初始向量 *IV*，获得不同的 01 序列。

2.6.2. NIST 随机性检测

PValue 值是 NIST 随机数检测方法输出的一个结果，每段 01 序列都可以得到一个 *PValue* 值，用来判定 01 序列是否是随机的。经过多次尝试，同一数据用四个不同的 NIST 随机数检测算法(Mode2)得到的散点图差异很小，所以最终结果展示以 *Mode*2 = 1 块内频数检验为例。当 01 序列的分段长度(*r*)取值 400 时，二维散点图分类比较明显。计算得到的 *PValue* 数据显示 HC256，RC4 生成的 01 序列具有良好的随机性。

2.6.3. 频率分布直方图

每段 *PValue* 序列做一个频率分布直方图，得到一组 *Pi* 值，全部 *PValue* 序列分段处理后得到多组 *Pi* 值。经过多次尝试，当 *PValue* 序列分段长度(*i*)取值 25，直方图区间(*j*)取值 10 时，二维散点图分类比较明显。

2.6.4. 可视化

每组 Pi 值经过横纵坐标计算公式计算后得到一个二维坐标对，多组 Pi 值得到多组二维坐标对，根据得到的所有二维坐标值，确定二维散点图显示的横纵坐标范围，最终打点画图。同一数据用两个不同的横纵坐标计算方法 (*Mode3*)生成的图形大致对称，结果展示以 *Mode*3 = 1 模幂运算为例。当模幂运算指数(*n*)取值 4 时，二维散点图分类明显。

3. 结果

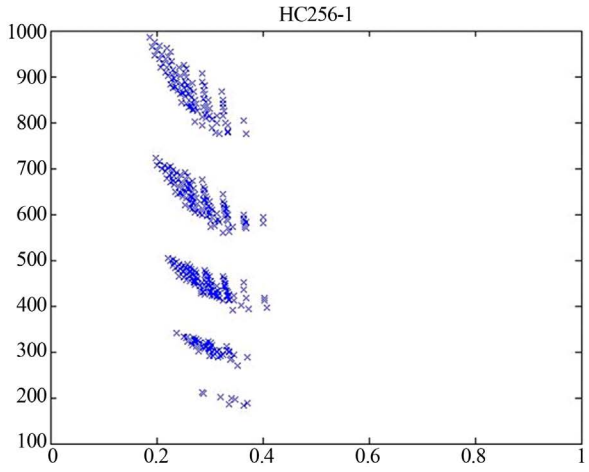
图 6：*LEN* = 102,400,000，*Mode*2 = 1，*r* = 400，*i* = 25，*j* = 10，*Mode*3 = 1，*n* = 4。左边三个图形(a) (c) (e)展示的是 HC256 在不同初始密钥(*Key*)和初始向量(*IV*)下的二维散点图，右边三个图形(b) (d) (f)展示的是 RC4 在不同初始密钥(*Key*)下的二维散点图。通过改变初始密钥(*Key*)和初始向量(*IV*)值，可以比较相同流密码算法在不同初始密钥和初始向量下的异同，以及不同的流密码算法(*Mode1*)之间的异同。

图 7：*LEN* = 102,400,000, *r* = 400, *i* = 25, *j* = 10, *Mode*3 = 2, *n* = 4。通过改变流密码算法(*Mode1*)和 NIST 随机数检测算法(*Mode*2)，比较不同流密码算法在不同 NIST 随机数检测算法下生成的二维可视化特征分布图的异同。

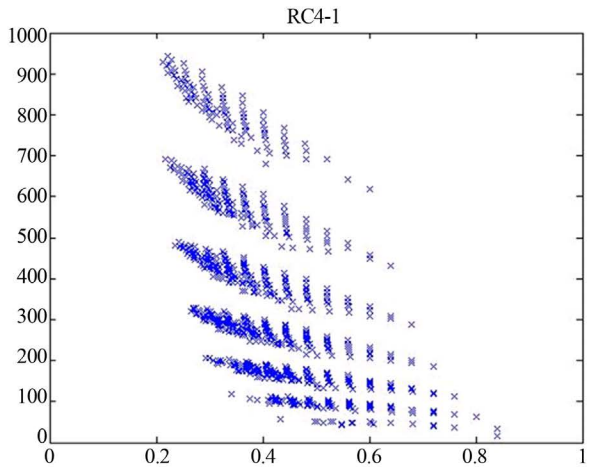
4. 讨论

由图 6(a) (c) (e)和图(b) (d) (f)可看出，同一个流密码算法(*Mode*1)取不同的初始密钥(*Key*)和初始向量

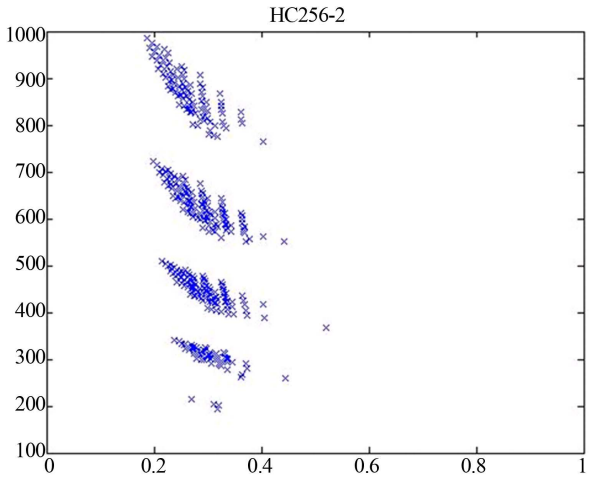
(*IV*)时，所得特征分布图相似度高，差别不大。



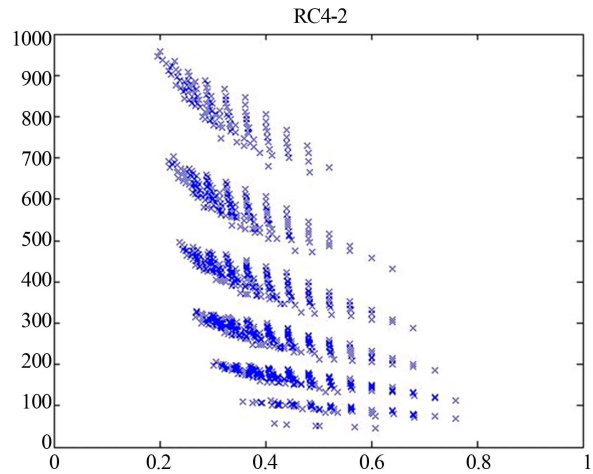
(a) *Mode*1 = *H* (10,240 个点) *Key*: 00000000; *IV*: 00000000



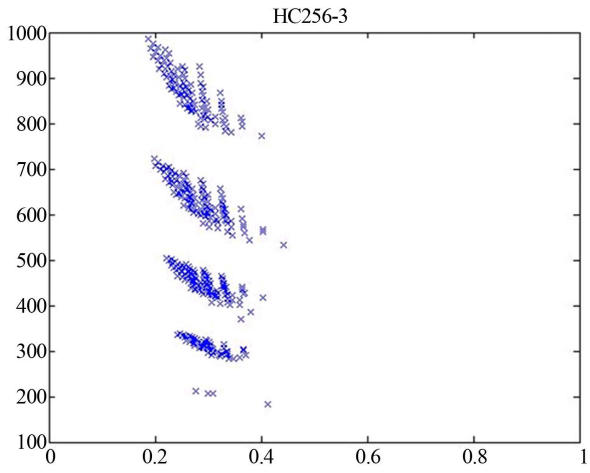
(b) *Mode*1 = *R* (10,240 个点) *Key*: lansry



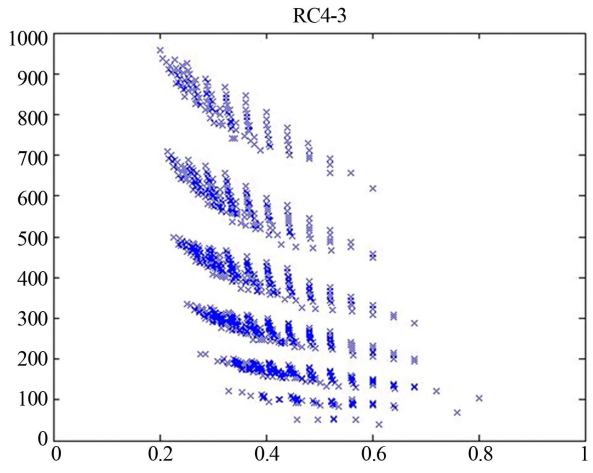
(c) *Mode*1 = *H* (10,240 个点) *Key*: 87654321; *IV*: 23456789



(d) *Mode*1 = *R* (10,240 个点) *Key*: abcdefghijklmnopqrstuvwxyz

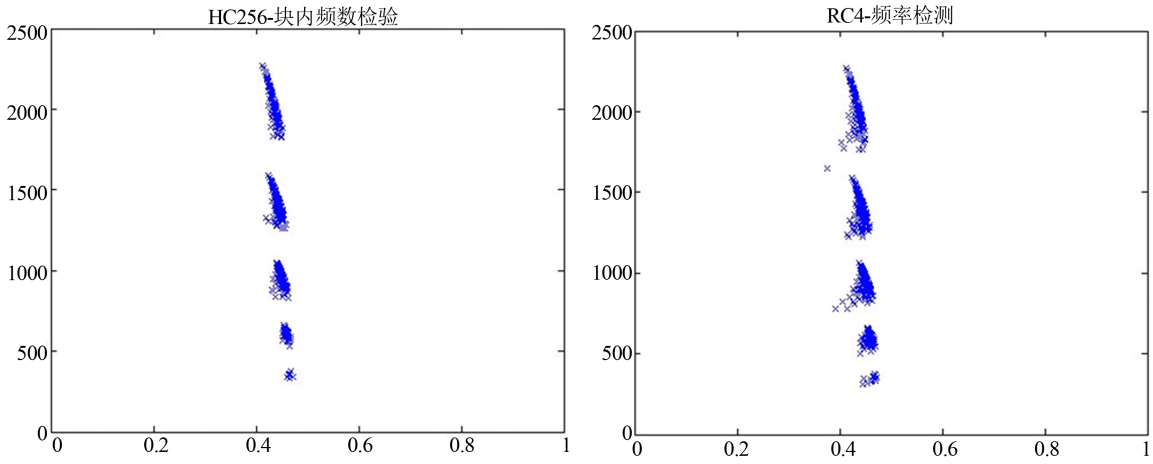


(e) *Mode*1 = *H* (10,240 个点) *Key*: 73298906 *IV*: 75281900



(f) *Mode*1 = *R* (10,240 个点) *Key*: qwertyuiopasdfghjklzxcvbnm

Figure 6. Six 2D scatter diagram in the range of *LEN* = 102,400,000, *Mode*2 = 1, *r* = 400, *i* = 25, *j* = 10, *Mode*3 = 1, *n* = 4
图 6. 6 张二维散点图：*LEN* = 102,400,000, *Mode*2 = 1, *r* = 400, *i* = 25, *j* = 10, *Mode*3 = 1, *n* = 4



(a) *Mode*1 = H, *Mode*2 = 1 (10,240 个点) *Key*: 00000000, *IV*: 00000000　　(b) *Mode*1 = R, *Mode*2 = 2 (10,240 个点) *Key*: lansry

Figure 7. Two 2D scatter diagram in the range of *LEN* = 102,400,000, *r* = 400, *i* = 25, *j* = 10, *Mode*3 = 2, *n* = 4
图 7. 2 张二维散点图：*LEN* = 102,400,000, *r* = 400, *i* = 25, *j* = 10, *Mode*3 = 2, *n* = 4

由图 6(a) (c) (e)和图(b) (d) (f)对比可看出，不同的流密码算法的特征分布图不同，该系统可用来区分 HC256 和 RC4 生成的 01 序列，同时也表明了 HC256 和 RC4 都具有各自的特征图谱。

由图 7 可看出 HC256 的块内频数检验和 RC4 的频率检测所得图形类似，分析 HC-256 的内部结构可知，HC256 有两个分别为 1024 个字的内部状态 P 和 Q，每运行 2048 步这两个状态全部更新一次，两个状态互相影响，类似于 RC4 中 SWAP 函数。图 7 的两个特征分布图相似也许和 HC256, RC4 算法的内部机制相似有关。

5. 结论

通过本文的系统化测量可视化机制的处理，HC256 和 RC4 这两种流密码算法生成的 01 序列显现出各自的特征图谱。通过该特征图谱可以对 HC256 和 RC4 生成的 01 序列进行区分，效果明显。同一流密码算法生成的不同 01 序列之间的特征图谱区分度不大，也进一步证明了 HC256 和 RC4 这两个流密码算法的安全性。

流密码在信息安全领域发挥的作用值得我们进一步对其进行探究。对 NIST 随机数测试方法做系统推广是复杂应用环境中需要重视的一类问题。对密钥流的随机性测量结果流进行多元统计分析形成可视化特征分布图作为流密码分析一个新的尝试，显示出巨大的潜力，具有一定的探索、研究价值。

项目基金

国家人才培养创新实验区资项目(RJ003)；国家自然科学基金项目(61362014)。

致　谢

感谢云南大学软件学院、云南省软件工程重点实验室信息安全基金及郑智捷博士国家自然科学基金(61362014)的支持，感谢黄源霖提供的 NIST 随机数检测代码。

参考文献 (References)

[1] William, S. (2010) 王丽娜, 傅建明, 等, 译. 密码编码学与网络安全: 原理与实践. 第四版, 电子工业出版社, 北京.

[2] 刘运毅, 覃团发, 倪皖荪, 张淑仪 (2006) 简评 ECRYPT 的候选流密码算法 (上). *信息安全与通信保密*, **7**,

26-28.

[3] Wu, H. (2004) A new stream cipher HC-256. In: *Fast Software Encryption*, Springer Berlin Heidelberg, Berlin, 226-244.

[4] Andrew Rukhin, Juan Soto, James Nechvatal, Miles Smid, Elaine Barker, Stefan Leigh, Mark Levenson, Mark Vangel, David Banks, Alan Heckert, James Dray, San Vo (2010) A statistical test suite for random and pseudorandom number generators for cryptographic applications. NIST Special Publication 800-22 Revision 1a.

[5] 许丽利 (2010) 聚类分析的算法及应用. 硕士学位论文, 吉林大学, 吉林.

[6] 陈一阳, 陈恭亮 (2010) 流密码典型分析方法及示例. *信息安全与通信保密*, **6**, 87-89.

[7] 李顺波, 胡予濮, 王艳 (2012) 针对流密码 HC-256'的区分攻击. *电子与信息学报*, **4**, 807-811.