《强智算法》算法分析报告

第 2 组

在报告开始前我们需要指出，第 3 小组给出的算法实现有一定的漏洞，在处理某些输入时程序会产生异常退出，这在某种程度上影响了我们的分析过程，详见附录中的环境说明。

1 随机性检测

我们使用 NIST 随机性检测工具，对该哈希函数的输出进行随机性检测。检测结果表明，该算法的随机性总体十分良好，对高密、低密和随机的输入数据，都能产生随机性较强的输出。

1.1 测试方法

1.1.1 数据输入

输入数据分为三种类型

类型	特征
高密	每位取 1 的概率为 95%
低密	每位取 1 的概率为 5%
随机	每位取 1 的概率为 50%

对于每种类型的输入数据，我们生成长度为 800 bit 的数据流计算其哈希值，然后将哈希值拼接进行检测。

1.1.2 检测规格

- 128 B
- 16 KB

NIST的部分检测对序列长度要求较高

检测	要求
Binary Matrix Rank Test	$n \geq 38912$
Overlapping Template Matching Test	$n > 10^6$
Maurer's "Universal Statistical" Test	$n \geq 387840$
Linear Complexity Test	$n \geq 10^6$
Random Excursions Test	$n \geq 10^6$
Random Excursions Variant Test	$n \geq 10^6$

为了能尽可能的利用 NIST 提供的多种测试，我们增加一组检测规格。

- 128 KB

共进行 $3 \times 3 = 9$ 组实验。对于每种检测规格，我们都从生成的哈希中取 300 个输入流进行检测。

对于不符合测试参数的测试点，在后续分析的过程中予以剔除，其余测试点成称为有效测试点。

1.2 测试结果

以下测试的有效测试点全部通过。

- BlockFrequency
- CumulativeSums
- Frequency
- LinearComplexity
- LongestRun
- OverlappingTemplate
- RandomExcursions
- RandomExcursionsVariant
- Rank
- Runs
- Serial
- Universal

以下测试的部分测试点存在异常。

1.2.1 ApproximateEntropy

在数据规模较大时存在异常。

数据规模	输入类型	通过率	P值
16KB	高密	291/300	0.000082*
128KB	随机	289/300*	0.000012*

1.2.2 FFT

在 $n = 128B$ 时分布 P 值异常。

数据规模	输入类型	通过率	P值
128B	高密	297/300	0.000000*
128B	低密	293/300	0.000000*
128B	随机	297/300	0.000000*

1.2.3 NonOverlappingTemplate

Non-Overlapping Template Test 有很多子测试模板，对每种 (数据规模，输入类型) 组合，统计不同模板的测试通过比例和P值均一性测试通过比例。

在 $n = 128B, n = 128KB$ 时通过率较低，在 $n = 16KB$ 通过率较好。

数据规模	输入类型	检测通过比例	P值通过比例
128B	高密	42/148	0/148
128B	低密	42/148	0/148
128B	随机	41/148	0/148
16KB	高密	148/148	148/148
16KB	低密	148/148	148/148
16KB	随机	147/148	148/148
128KB	高密	113/148	139/148
128KB	低密	123/148	134/148
128KB	随机	119/148	129/148

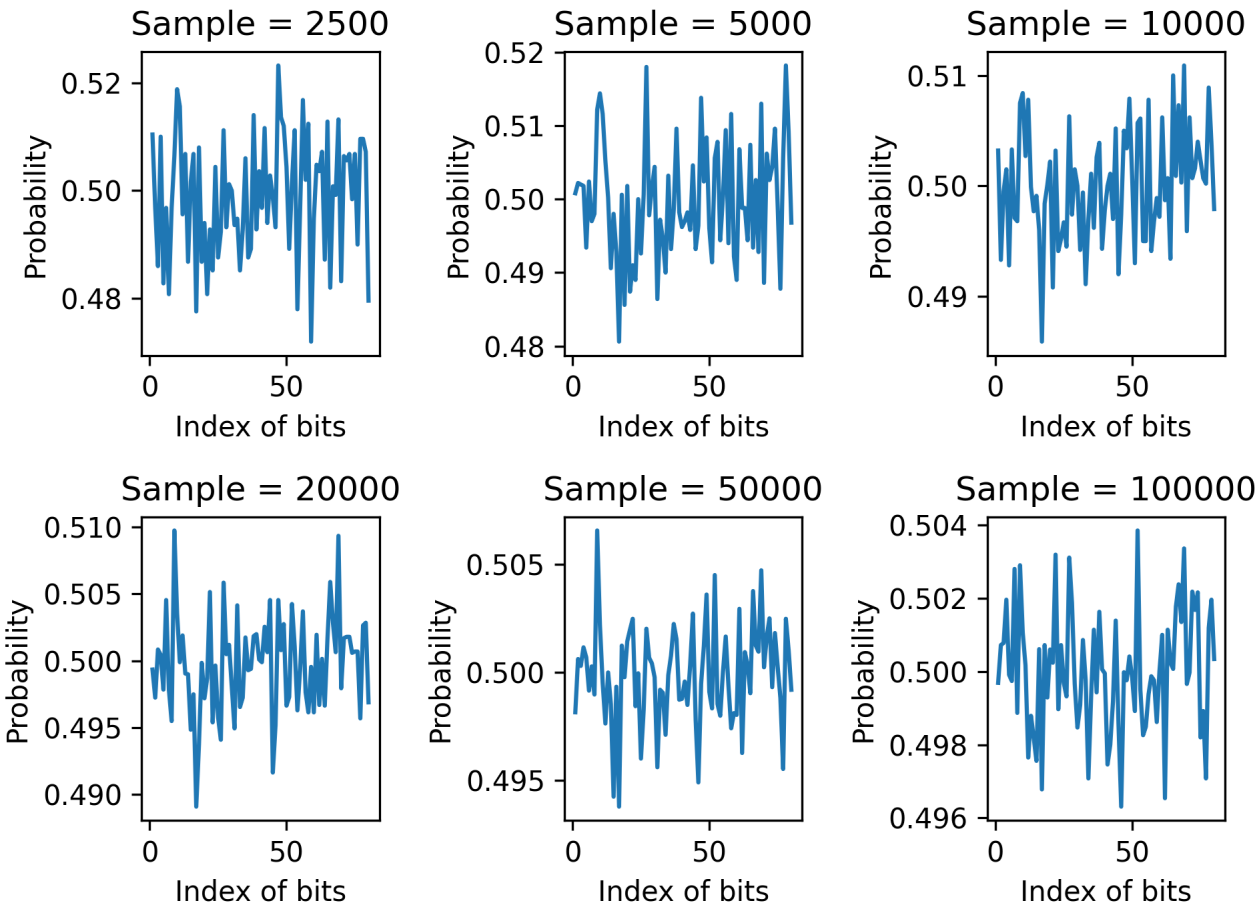
全部测试数据据详见附录。

2 安全性分析

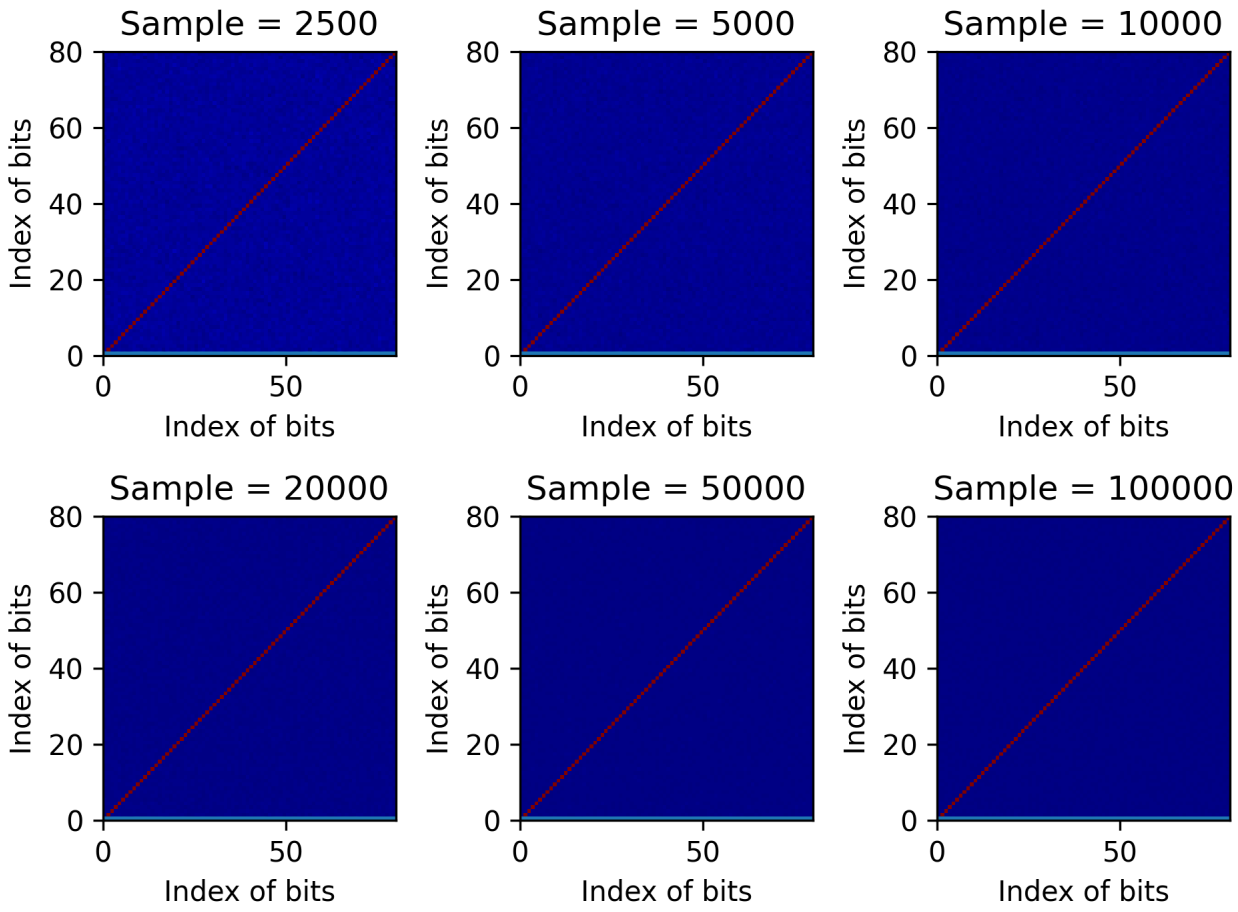
2.1 区分攻击

该算法对于区分攻击具有较强的抗性。我们使用该算法随机生成了 {2500, 5000, 10000, 20000, 50000, 100000} 个哈希值，然后对这些哈希值进行统计分析。

首先是逐 Bit 分析，我们统计每个 Bit 的 “1” 的出现次数。如果是随机分布，每个 Bit 应服从 $p = 0.5$ 的伯努利分布。绘出下图，证明其确实具有较好的随机性。测试脚本见 `./distinguishing_attack` 目录。



然后是检验不同 Bit 之间的关联性。我们绘制出下图，图中每个点的横纵坐标确定哈希值两个比特位，其数值代表两个比特位相同的概率。可以看出不同比特之间没有明显的关联性。



2.2 碰撞攻击 (Redo?)

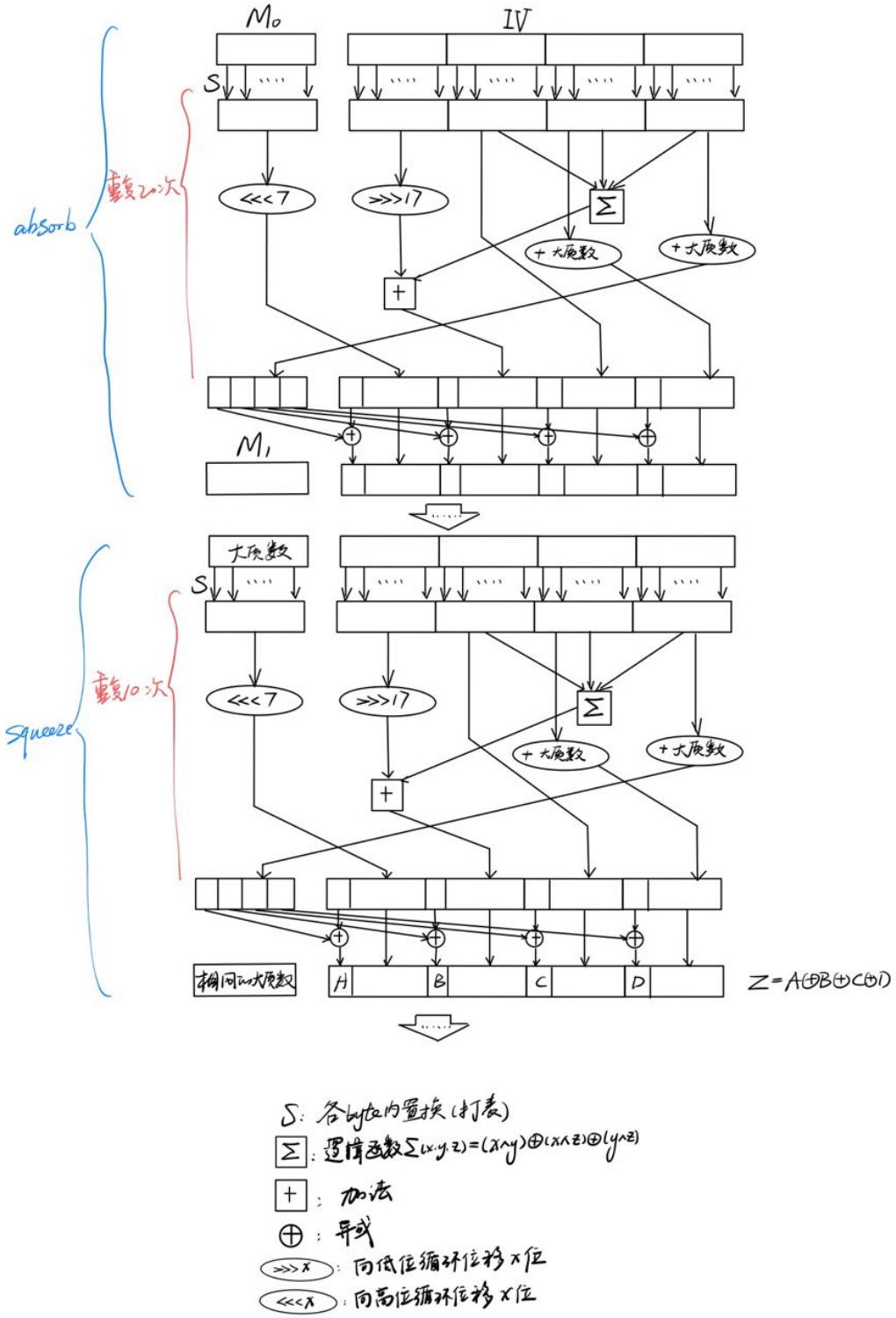
碰撞攻击是说，是否存在 M_1, M_2，使得 $M_1 \neq M_2$ 但 $H(M_1) = H(M_2)$。为寻找碰撞攻击我们使用 Floyd 的随机路径算法。我们给定初始串 M_0，记 $M_i = H(M_{i-1})$，且每次迭代步我们都将 M_i 计入集合 K 中。如果找到某次迭代步的迭代结果 M_j 已经在集合 K 中，那么说明我们这条试探链构成了一个“环”，而这个“环”中必然有一个结点存在两个入度，这两个入边所对应的邻点便是哈希值产生碰撞的原象。

由生日攻击的相关理论我们可以知道，在 2^{80} 种随机数的排列组合中，任从这个分布中取样 2^{40} 个元素，有与 0.5 数量级相同的概率产生重复。而如果哈希算法的随机性足够高，那么我们只有当计算量达到这个量级时，才能够找到相应的碰撞。

限于计算性能所限，我们这里取最长的试探链长为 $10,000$，即从某个 M_0 出发，我们迭代 $10,000$ 步，查看其是否产生碰撞。我们随机选取不同的初值 M_0，重复了 25 次上述实验，该算法均未发生碰撞现象。测试脚本见 `./collision_attack` 目录。

3 Ablation Study

在这一部分我们尝试分析算法中每一个部件的作用，以尝试解释其产生的结果。同时，根据我们对于部件作用的分析，我们会给出为了保证算法整体的安全性，每个部件至少所应选取的参数范围。下图为“强智算法”报告中的实现流程图，我们按照从上到下的顺序依次分析。



3.1 S-Box

首先是 S-Box，这个置换起到的作用是，将输入字符中“邻近”的特性均匀地混淆。也就是说，即使 S-Box 的输入 I_1, I_2 仅仅有 1 Bit 的不同，经过 S-Box 的置换之后所产生的输出结果 $S_1 = S(I_1), S_2 = S(I_2)$ 是完全不同的。这里置换表在算法给出时已经固定。

3.2 Absorbing

为了方便我们分析 Absorbing 的轮函数的效果，我们选取了特殊的输入 I_1, I_2，使得 $S_1 = S(I_1), S_2 = S(I_2)$ 的海明距离只有 1 Bit。也就是说，我们从 S 盒中逆向寻找了输入 I_1, I_2。为了探究 Absorbing 轮函数的作用轮数对 Absorbing 阶段输出结果的影响，我们做了以下探究实验：

选取输入 $I_1 = \{0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x00\}$，输入 $I_2 = \{0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x21\}$，这里 S_1, S_2 只有 1 Bit 的差距。我们使用该串作为 Hash 函数的初始输入，输出在做长度填充之前的隐藏状态 state 的字节表示。

Absorb 轮数	I_1 的输出	I_2 的输出
1	C988397AC302B838 21**B40**DE2D516AFCB 48967973C0450B40 167FCC7F0903FADD	C988397AC302B838 2**2008**DE2D516AFCB 48967973C0450B40 167FCC7F0903FADD
2		
3		

4 附录

4.1 测试环境

以上的分析我们使用的编译命令为 `gcc *.c -O3 main`，测试运行环境为：

- TODO
- TODO
- TODO

4.2 随机性检测详细结果

4.2.1 测试框架说明

本项测试的框架见 `./random_analysis` 目录

- 每项测试的原始报告及测试脚本见 `128B`，`16KB`，`128KB` 三个目录。
- 测试时使用的随机数生成程序见 `main.c`
- 将测试报告转化成下面表格的工具为 `report_gen.ipynb`

下面对每类测试的通过情况进行列举，通过率不达标或 P 值过低的，用 * 进行标注。

4.2.2 ApproximateEntropy

以下测试点存在异常：

数据规模	输入类型	通过率	P值
16KB	高密	291/300	0.000082*
128KB	随机	289/300*	0.000012*

其余测试点全部通过：

数据规模	输入类型	通过率	P值
128B	高密	295/300	0.487885
128B	低密	300/300	0.798139
128B	随机	297/300	0.699313
16KB	低密	293/300	0.000555
16KB	随机	295/300	0.001943
128KB	高密	293/300	0.000427
128KB	低密	294/300	0.000111

4.2.3 BlockFrequency

有效测试点全部通过。

数据规模	输入类型	通过率	P值
128B	高密	295/300	0.487885
128B	低密	297/300	0.561227
128B	随机	299/300	0.090936
16KB	高密	299/300	0.383827
16KB	低密	295/300	0.075719
16KB	随机	297/300	0.401199
128KB	高密	299/300	0.616305
128KB	低密	295/300	0.013889
128KB	随机	297/300	0.726503

4.2.4 CumulativeSums

有效测试点全部通过。

数据规模	输入类型	通过率	P值
128B	高密	296/300	0.746572
128B	高密	297/300	0.003046
128B	低密	296/300	0.150906
128B	低密	298/300	0.047682
128B	随机	296/300	0.000682
128B	随机	295/300	0.006048
16KB	高密	299/300	0.366918
16KB	高密	298/300	0.087338
16KB	低密	297/300	0.637119
16KB	低密	298/300	0.798139
16KB	随机	300/300	0.872947
16KB	随机	299/300	0.990369
128KB	高密	295/300	0.345115
128KB	高密	294/300	0.678686
128KB	低密	299/300	0.228764
128KB	低密	299/300	0.443451
128KB	随机	298/300	0.168733
128KB	随机	297/300	0.791880

4.2.5 FFT

以下测试点存在异常：

数据规模	输入类型	通过率	P值
128B	高密	297/300	0.000000*
128B	低密	293/300	0.000000*
128B	随机	297/300	0.000000*

其余测试点全部通过：

数据规模	输入类型	通过率	P值
16KB	高密	292/300	0.994250
16KB	低密	294/300	0.425059
16KB	随机	297/300	0.931952
128KB	高密	295/300	0.195163
128KB	低密	296/300	0.009311
128KB	随机	291/300	0.657933

4.2.6 Frequency

有效测试点全部通过。

数据规模	输入类型	通过率	P值
128B	高密	297/300	0.862344
128B	低密	298/300	0.534146
128B	随机	296/300	0.014550
16KB	高密	297/300	0.217094
16KB	低密	297/300	0.609377
16KB	随机	300/300	0.822534
128KB	高密	295/300	0.032203
128KB	低密	300/300	0.289667
128KB	随机	297/300	0.872947

4.2.7 LinearComplexity

有效测试点全部通过。

数据规模	输入类型	通过率	P值
128KB	高密	296/300	0.602458
128KB	低密	296/300	0.554420
128KB	随机	296/300	0.785562

4.2.8 LongestRun

有效测试点全部通过。

数据规模	输入类型	通过率	P值
128B	高密	299/300	0.692455
128B	低密	299/300	0.202268
128B	随机	296/300	0.129620
16KB	高密	298/300	0.872947
16KB	低密	297/300	0.220931
16KB	随机	298/300	0.419021
128KB	高密	294/300	0.195163
128KB	低密	299/300	0.946308
128KB	随机	297/300	0.816537

4.2.9 NonOverlappingTemplate

Non-Overlapping Template Test 有很多子测试模板，在这里不进行逐一列举。

对每种 (数据规模，输入类型) 组合，考察测试通过率和P值均一性测试通过率。

数据规模	输入类型	检测通过比例	P值通过比例
128B	高密	42/148	0/148
128B	低密	42/148	0/148
128B	随机	41/148	0/148
16KB	高密	148/148	148/148
16KB	低密	148/148	148/148
16KB	随机	147/148	148/148
128KB	高密	113/148	139/148
128KB	低密	123/148	134/148
128KB	随机	119/148	129/148

4.2.10 OverlappingTemplate

有效测试点全部通过。

数据规模	输入类型	通过率	P值
128KB	高密	299/300	0.115387
128KB	低密	294/300	0.165646
128KB	随机	297/300	0.893001

4.2.11 RandomExcursions

有效测试点全部通过。

数据规模	输入类型	通过率	P值
128KB	高密	194/200	0.296834
128KB	高密	196/200	0.946308
128KB	高密	200/200	0.834308
128KB	高密	198/200	0.851383
128KB	高密	198/200	0.202268
128KB	高密	198/200	0.224821
128KB	高密	197/200	0.524101
128KB	高密	199/200	0.709558
128KB	低密	188/194	0.196086
128KB	低密	190/194	0.461912
128KB	低密	193/194	0.130014
128KB	低密	193/194	0.095365
128KB	低密	191/194	0.040385
128KB	低密	193/194	0.856619
128KB	低密	192/194	0.598820
128KB	低密	191/194	0.587927
128KB	随机	169/170	0.877806
128KB	随机	170/170	0.018652
128KB	随机	168/170	0.194135
128KB	随机	169/170	0.025698
128KB	随机	168/170	0.679903
128KB	随机	166/170	0.379806
128KB	随机	170/170	0.102885
128KB	随机	168/170	0.903500

4.2.12 RandomExcursionsVariant

有效测试点全部通过。

数据规模	输入类型	通过率	P值
128KB	高密	200/200	0.626709
128KB	高密	199/200	0.105618
128KB	高密	197/200	0.167184
128KB	高密	197/200	0.289667
128KB	高密	197/200	0.428095
128KB	高密	197/200	0.289667
128KB	高密	195/200	0.474986
128KB	高密	197/200	0.242986
128KB	高密	198/200	0.176657
128KB	高密	200/200	0.167184
128KB	高密	198/200	0.595549
128KB	高密	200/200	0.534146
128KB	高密	199/200	0.668321
128KB	高密	199/200	0.935716
128KB	高密	199/200	0.749884
128KB	高密	199/200	0.719747
128KB	高密	199/200	0.358641
128KB	高密	199/200	0.825505
128KB	低密	193/194	0.534146
128KB	低密	192/194	0.642599
128KB	低密	191/194	0.403984
128KB	低密	192/194	0.422829
128KB	低密	193/194	0.534146
128KB	低密	193/194	0.118630
128KB	低密	192/194	0.231703
128KB	低密	193/194	0.073776
128KB	低密	193/194	0.544788
128KB	低密	189/194	0.577070
128KB	低密	187/194	0.359074
128KB	低密	189/194	0.098423
128KB	低密	191/194	0.231703
128KB	低密	192/194	0.653554
128KB	低密	193/194	0.225440
128KB	低密	193/194	0.555494
128KB	低密	194/194	0.781206

数据规模	输入类型	通过率	P值
128KB	低密	194/194	0.598820
128KB	随机	169/170	0.099338
128KB	随机	169/170	0.313747
128KB	随机	169/170	0.818661
128KB	随机	169/170	0.049645
128KB	随机	168/170	0.071961
128KB	随机	169/170	0.570068
128KB	随机	168/170	0.182140
128KB	随机	169/170	0.369867
128KB	随机	168/170	0.360093
128KB	随机	167/170	0.296409
128KB	随机	169/170	0.961917
128KB	随机	169/170	0.582174
128KB	随机	169/170	0.546044
128KB	随机	169/170	0.188060
128KB	随机	169/170	0.379806
128KB	随机	169/170	0.083167
128KB	随机	169/170	0.630995
128KB	随机	169/170	0.453721

注：RandomExcursion及其变种有包含多个测试项目。

4.2.13 Rank

有效测试点全部通过。

数据规模	输入类型	通过率	P值
16KB	高密	294/300	0.999438
16KB	低密	297/300	0.437274
16KB	随机	298/300	0.425059
128KB	高密	298/300	0.224821
128KB	低密	296/300	0.481416
128KB	随机	295/300	0.249284

4.2.14 Runs

有效测试点全部通过。

数据规模	输入类型	通过率	P值
128B	高密	298/300	0.935716
128B	低密	295/300	0.209577
128B	随机	297/300	0.588652
16KB	高密	298/300	0.474986
16KB	低密	296/300	0.581770
16KB	随机	299/300	0.964295
128KB	高密	296/300	0.949602
128KB	低密	298/300	0.753185
128KB	随机	297/300	0.162606

4.2.15 Serial

有效测试点全部通过。

数据规模	输入类型	通过率	P值
128B	高密	298/300	0.961593
128B	高密	293/300	0.153763
128B	低密	298/300	0.906970
128B	低密	299/300	0.228764
128B	随机	296/300	0.872947
128B	随机	299/300	0.425059
16KB	高密	294/300	0.096578
16KB	高密	295/300	0.507512
16KB	低密	299/300	0.245072
16KB	低密	298/300	0.581770
16KB	随机	294/300	0.798139
16KB	随机	297/300	0.014216
128KB	高密	297/300	0.455937
128KB	高密	298/300	0.228764
128KB	低密	298/300	0.010237
128KB	低密	296/300	0.113151
128KB	随机	297/300	0.350485
128KB	随机	298/300	0.657933

4.2.16 Universal

有效测试点全部通过。

数据规模	输入类型	通过率	P值
128KB	高密	297/300	0.766282
128KB	低密	299/300	0.345115
128KB	随机	296/300	0.520767

4.3 实现问题

我们指出算法实现有漏洞的地方在于，在处理某些文件时，其会产生异常退出。