AES Java™ Technology Comparisons

Alan Folmsbee, Sun Microsystems, Inc.

Advanced Encryption Standard candidate algorithm comparisons based on the Java technology implementations.

1.0 Introduction

The Advanced Encryption Standard (AES) selection process is described on the web site of the National Institute of Standards and Technology (NIST) at:

http://csrc.nist.gov/encryption/aes/aes_home.htm

The NIST, a part of the US Department of Commerce, is now in the second round of the selection process, in which five candidate algorithms will be selected out of the fifteen candidates which qualified during the first round. The public was invited to recommend which candidates are the best. This paper makes recommendations to the NIST about which candidates should be considered during the second round of the selection process, according to criteria described in the rest of this paper.

The bit avalanche rates were measured for all 15 candidate algorithms to quantify this basic effect which is needed by any cryptographic algorithm. Also, an evaluation of the general structures of these candidates resulted in the creation of a new way to categorize these symmetric algorithms: the Fractional Feistel Dimension or “fracstel” number.

The Java technology implementations were tested for speed by using the Monte Carlo Tests (MCT) and the Known Answer Tests (KAT) that were supplied by NIST on the CD-2 disk. The RAM memory that the algorithms use was estimated by visually examining the source code files and counting the variable declarations. The RAM estimates did not use any of the improvements published by the cipher designers after the First AES Candidate Conference was held. The ROM memory sizes were measured from the executable file sizes.

2.0 Top Five Candidates

According to the analysis done, the five preferred candidates are:

1 RC6 from RSA Laboratories

2 MARS from IBM

3 Safer+ from Cylink Corporation

4 Serpent from Anderson, Biham, and Knudsen

5 Crypton from Future Systems, Inc.

The ranking was done by measuring five quantities, ranking these results from 1 to 15, and then assigning weights to these ranks as follows:

TABLE 1. Five Measurement Categories and the Relative Weights Given to Each

Measurement	**Weight**
Excess Avalanche	4
RAM Size	3
ROM Size	2
Monte Carlo Test Speed	1
Known Answer Test Speed	1

Excess avalanche was calculated by finding how many rounds it took for avalanche to be near the ideal level (64 bits +/- 16 bits), and then dividing the algorithm’s total rounds by that number. This was measured during 12,800 encryptions using 100 keys for each candidate, for each round count examined. Each key was 128 bits long.

Avalanche, which is similar to diffusion, is discussed in [1]. Avalanche is the effect where a change in a bit of the plaintext will cause more than one bit to change in the output of the round function. As more rounds are executed, an increasing number of bits may be changed as the result of a single plaintext bit changing.

3.0 Summary of Measurements

TABLE 2. AES Candidate Java Technology Measurement Results Summary

Name	Excess Avalanche	RAM size bytes	ROM size bytes	MCT kbit/s	KAT kbit/s	Fracstel # (F)	Total Rounds
RC6	6.6	480	7800	5061	355	1.34	20
MARS	6.4	456	19719	3284	270	1.28	32
SAFER+	4.0	320	13200	811	169	1.00	8
SERPENT	16.0	1248	38900	2544	238	1.00	32
CRYPTON	4.6	800	13979	2710	281	1.00	14
FROG	5.3	576	14100	1029	7	1.00	8
E2	6.0	880	275857	2934	265	1.03	12
MAGENTA	2.0	464	6088	164	106	2.00	6
CAST256	6.0	2260	29000	1213	214	4.00	48
HPC	8.0	15000	44889	2710	185	0.19*	8
DFC	2.6	632	11147	35	16	2.00	8
LOKI97	5.3	10240	15956	420	161	2.00	16
TWOFISH	5.3	8000	19181	1729	156	2.00	16
RIJNDAEL	5.0	20000	18405	513	184	1.00	10
DEAL	3.0	4355	20043	664	176	2.00	6

Notes:

* When one HPC round is broken down into 10 sections, the first 2 sections always produced more than one ciphertext bit change when any plaintext bit was changed. The results from the first HPC section are described later in this report.

# The Fracstel number, F, is the Fractional Feistel Dimension defined on the next page. It is an estimate of the number of rounds needed to ensure that each plaintext bit causes some avalanche.

The speed measurements come from using Sun‘s Ultra Enterprise™ 2 workstation with one UltraSparc™ processor running at 200Mhz. It has 256 Megabytes of RAM. The Solaris™ operating system was used with a Java technology interpreter and a Just-In-Time compiler.

The “fracstel” number, F, is a new term created to express fractional Feistel block splitting observations. The designers of the candidates define what a “round” is, and this definition may be different for various designers. The fracstel is a measure that can be used by independent evaluators to categorize the algorithms’ structures, without relying on the various definitions which are used for the AES candidates. Evaluators can use fractional rounds or multiple rounds to calculate the fracstel number and use the results to produce a way of comparing the structures of the algorithms without needing to understand the logical and mathematical meanings of instructions in computer programs.

F is calculated by the following method: various keys are used (this paper used 100 keys for avalanche testing). Single bit changes are made in the plaintext compared to a reference plaintext. After all bit positions in the plaintext are controlled in this way, when r rounds are used, if more than one ciphertext bit is changed for all 128 plaintext cases, for all keys, the fracstel # is r or less. Some algorithms, when using r rounds, result in some of the ciphertexts being changed in only one bit position when the plaintext is changed in one bit position. This is viewed as a fractional Feistel structure, and it is calculated as follows:

r=number of rounds in the test. A “round” is defined by the cipher designer.

k=number of keys used during avalanche tests

p= number of plaintexts changed by one bit compared to a reference plaintext, for one key

c= number of ciphertexts produced with only one ciphertext bit changed, when compared to a reference ciphertext, for tests with k keys

F=rpk/(pk-c)

For the tests on RC6, F=1*128*100/(128*100-3280) = 1.34

For the tests on Safer+, F=1*128*100/(128*100-0) = 1.00

For the tests on Cast256, F=1*128*100/(128*100-9600) = 4.00

For HPC with 0.1 round, F=0.1*128*100/(128*100-6145) = 0.19

For HPC with 0.2 round, F=0.2*128*100/(128*100-0) = 0.20

Notice that F=4 for Cast256, since it is a quad-round algorithm. F is 4 for Cast256 even when the tests have the number of rounds being 1, 2, 3, or 4. (For DES, F=2 because in one round, only half of the plaintext can avalanche to produce a ciphertext). The Frog algorithm has 8 rounds, but each round has 16 identical sub-rounds, so F=1 even if only 1/16 of a round is used to calculate F. The fracstel number is always one for Frog for any integer number of subrounds used to calculate F, when one round or less is measured.

It is interesting to note that all five of the candidates which the author recommends have F less than two. This indicates that the preferred candidates have improved upon the dual-block Feistel structure which DES had, and have moved toward a uni-block structure in which the entire 128 bit block is involved in avalanche in one round. The author conjectures that this trend is one which results in higher security. Large blocks are more secure than small blocks when the rest of the algorithm has the same style. Since this is only a conjecture, it was not used to rank the candidates in this report. This fracstel number can be used for non-Feistel as well as Feistel ciphers to give an estimated number of rounds needed to ensure that each plaintext bit can create some avalanche.

TABLE 3. Candidate Ranking Calculations

Name	Excess Avalanche rank	RAM size rank	ROM size rank	MCT rank	KAT rank	Total Weighted Score	Final Rank
RC6	3	4	2	1	1	30	1
MARS	4	2	10	2	3	47	2
SAFER+	9	1	4	10	10	67	3
SERPENT	1	9	13	6	5	68	4
CRYPTON	8	7	5	4	2	69	5
FROG	6	5	6	9	15	75	6
E2	5	8	15	3	4	81	7
MAGENTA	12	3	1	14	13	86	8
CAST256	5	10	12	8	6	88	9
HPC	2	14	14	5	7	90	10
DFC	11	6	3	15	14	97	11
LOKI97	6	12	7	13	11	98	12
TWOFISH	6	13	9	7	12	100	13
RIJNDAEL	7	15	8	12	8	109	14
DEAL	10	11	11	11	9	115	15

4.0 Explanations of Figures

The rest of this paper concerns the figures which illustrate the avalanche rates of all 15 candidate algorithms. Figure 1 compares nine of the candidates together to show the average amount of avalanche versus the number of rounds. The horizontal axis is for rounds one through four. The vertical axis is the number of bits, on the average for one encryption, that change in a ciphertext, compared with a reference ciphertext, when a single bit is changed in the plaintext, compared with a reference plaintext. 16 keys were used while 128 single bits changes were made for each key, for a total of 2k encryptions for each algorithm for each round count.

Figure 2 is for the remaining six algorithms for the same conditions as in Figure 1, except that seven rounds are shown in this Figure. Since the block size is 128 bits, after several rounds, all of the candidates approached the 64 bit level for average avalanche.

Figure 3 is a histogram of avalanche for the RC6 algorithm. On the horizontal axis are the number of bits that changed in the ciphertext after a single bit was changed in the plaintext. The vertical axis is labelled # Occur. That is the number of occurrences of encryptions when a certain number of bits were changed in the ciphertext. The total number of encryptions for this histogram was 12,800 for each round amount. The number of keys used was 100 for each round amount. The vertical scale is limited to 2000, but the peak number of occurrences was 3280 when only one bit changed in the ciphertext in the first round.

Figures 4 through 17 are histograms of the rest of the 15 candidate algorithms with the same horizontal axis of bits up to 88. The vertical scales for the various figures are not always the same because of the variety of avalanche peaks found in the first round. The vertical scale is limited to 2000 occurrences, so when a peak value is higher than that, it is noted on the figure.

Conclusion

The 17 figures in this paper were used to estimate the excess avalanche for each candidate, as reported in Table 2. The excess avalanche was estimated by dividing the total rounds for a candidate by the number of rounds it took for the avalanche measurements to have values which are close to the optimal values. This technique has some vagueness, due to the integer quantization of rounds, and because of variable estimates of closeness to the optimum values. Excess avalanche was given the highest "weight", as listed in Table 1, because this affects the security of the algorithm more than speed or code size do. The five candidates which are recommended to be chosen for further analysis are RC6, MARS, Safer+, Serpent, and Crypton.

5.0 Figures of Candidate Avalanche Measurements

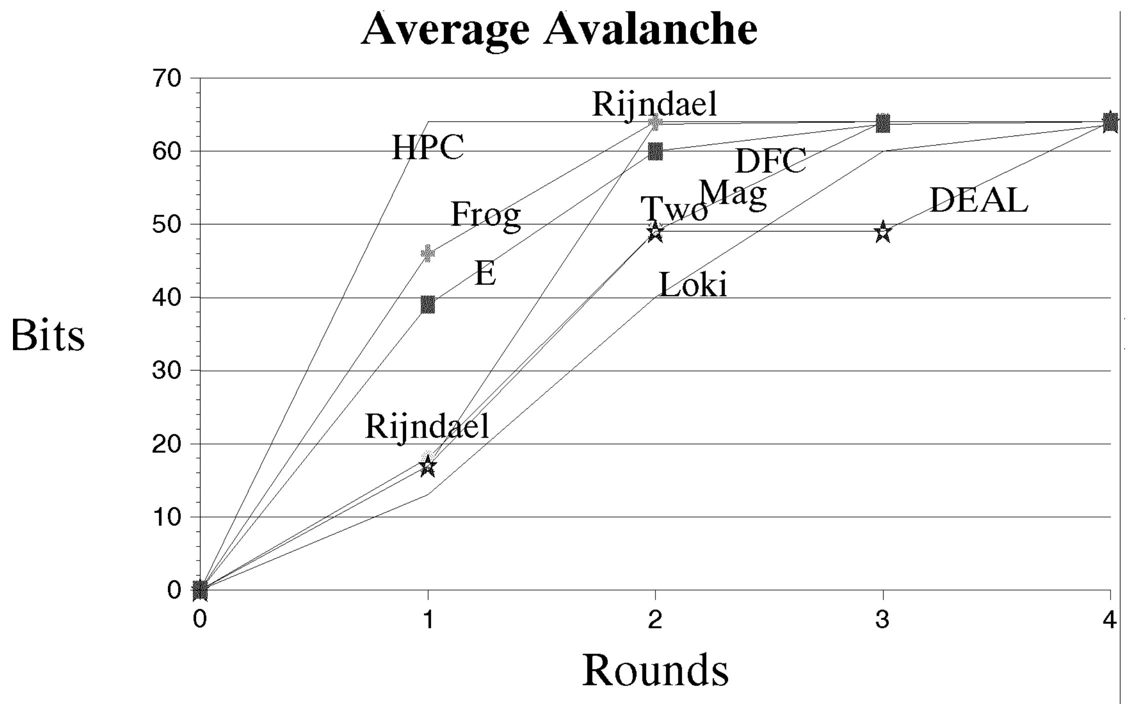

Figure 1

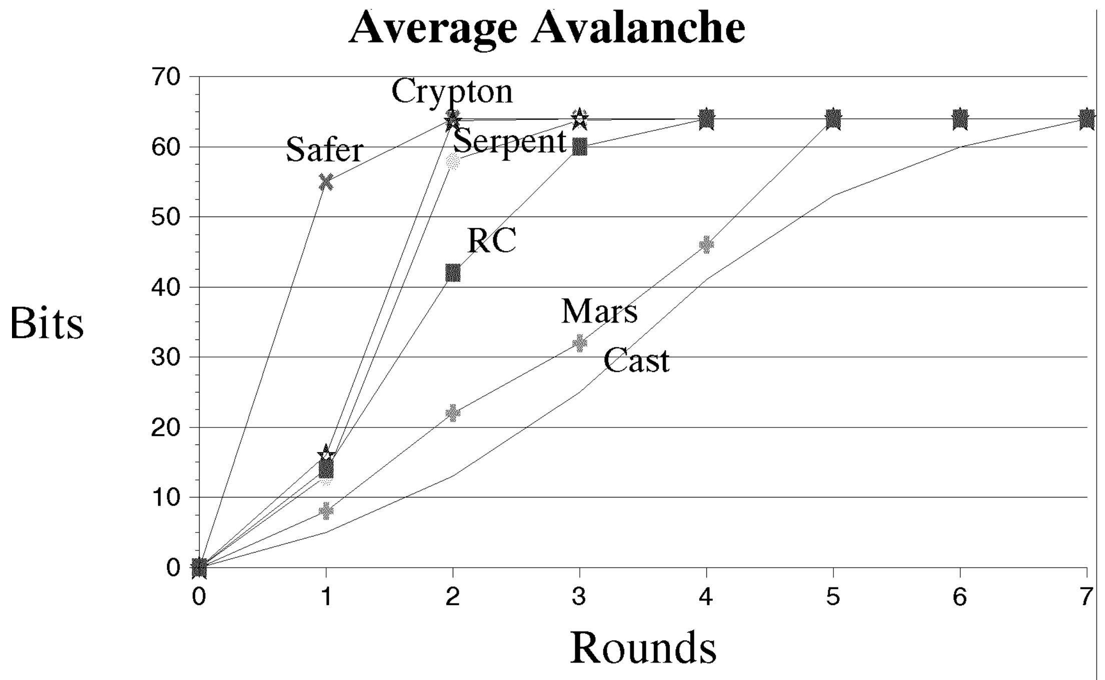

Figure 2

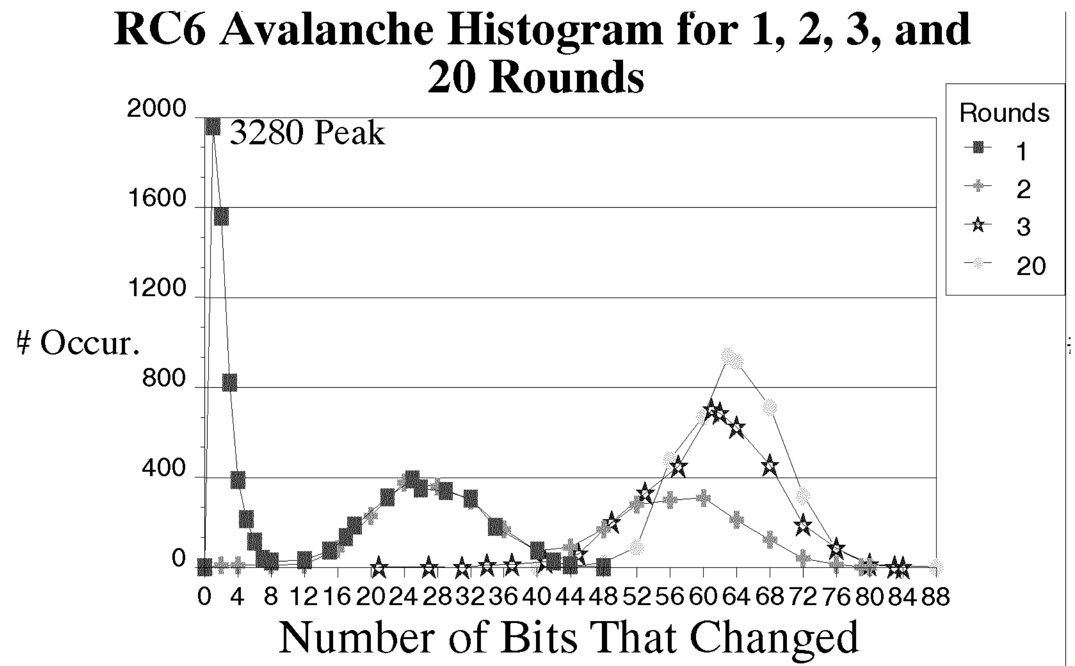

Figure 3

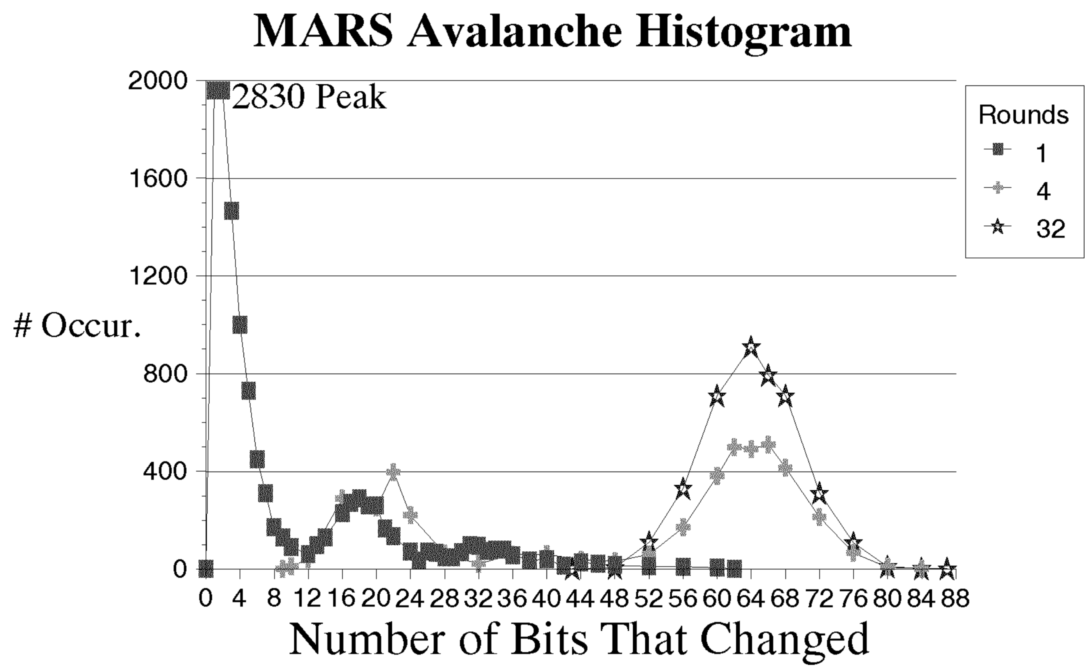

Figure 4

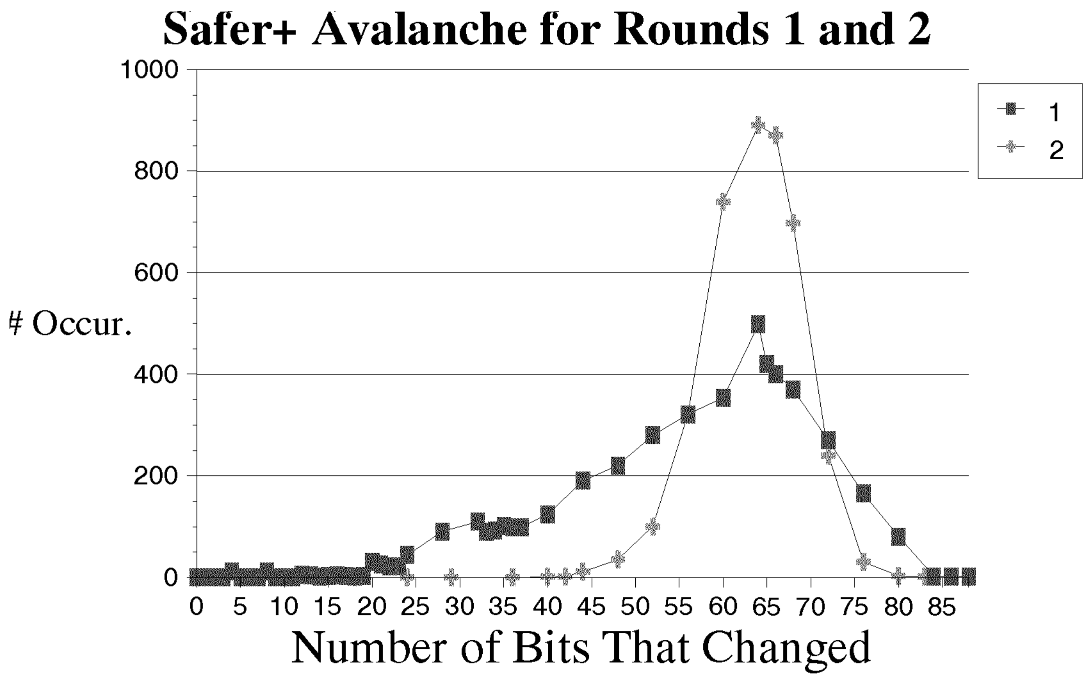

Figure 5

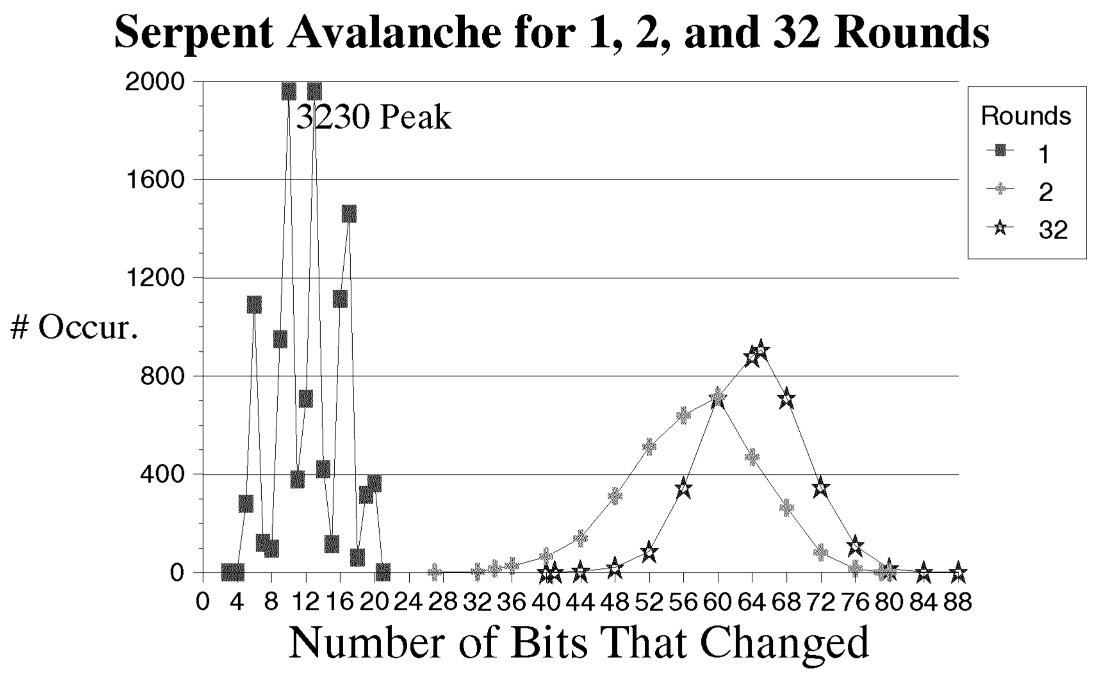

Figure 6

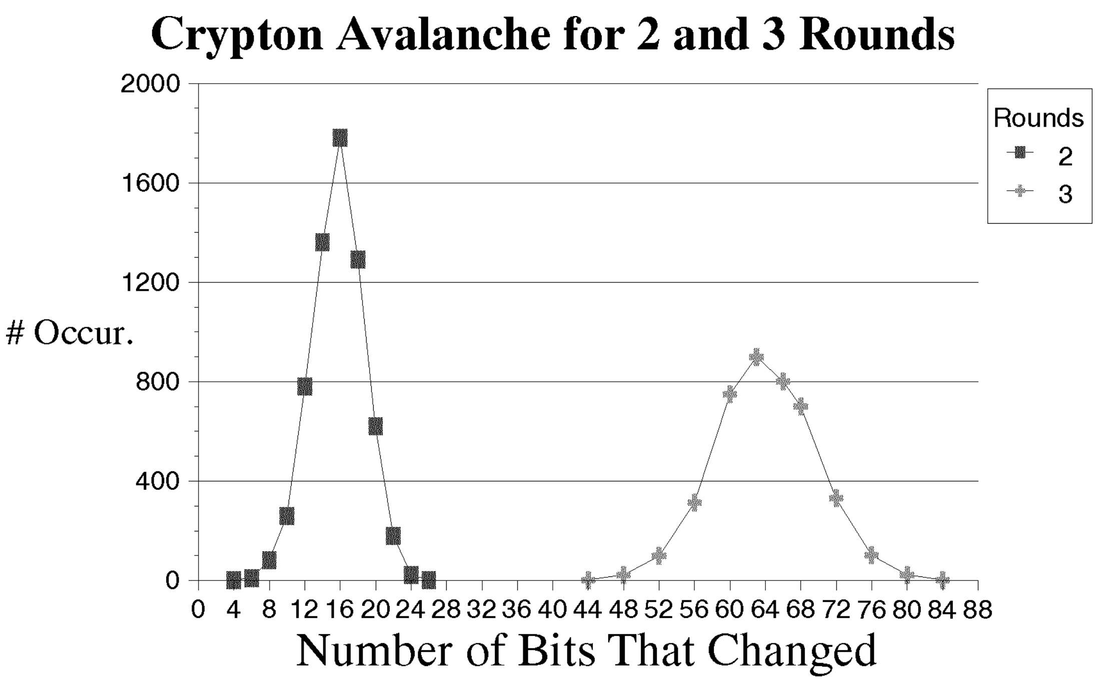

Figure 7

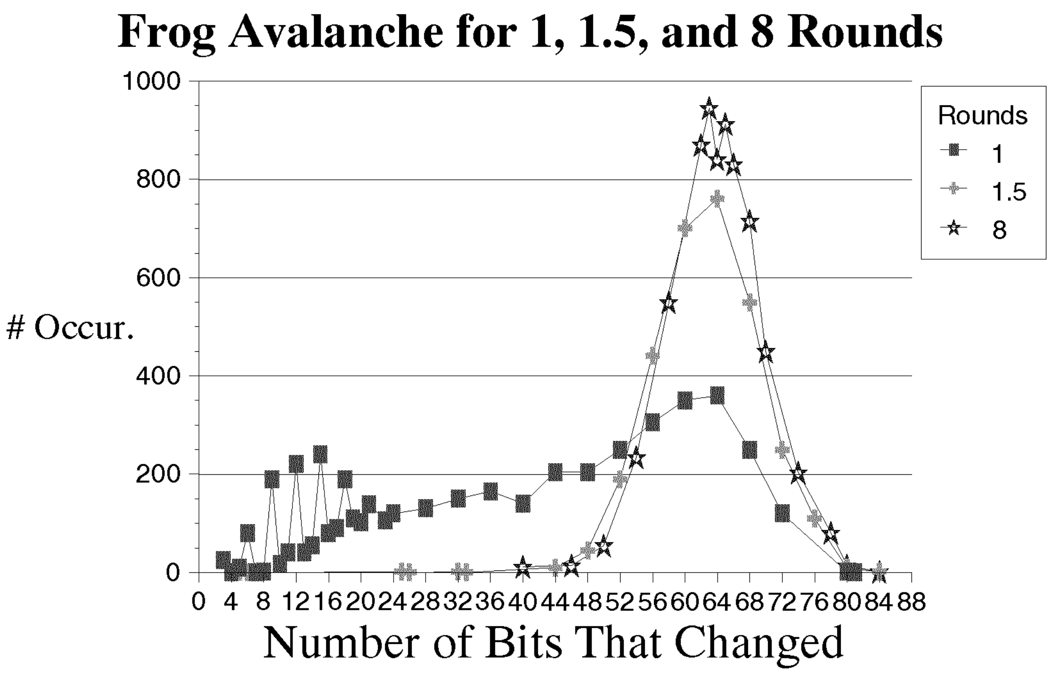

Figure 8

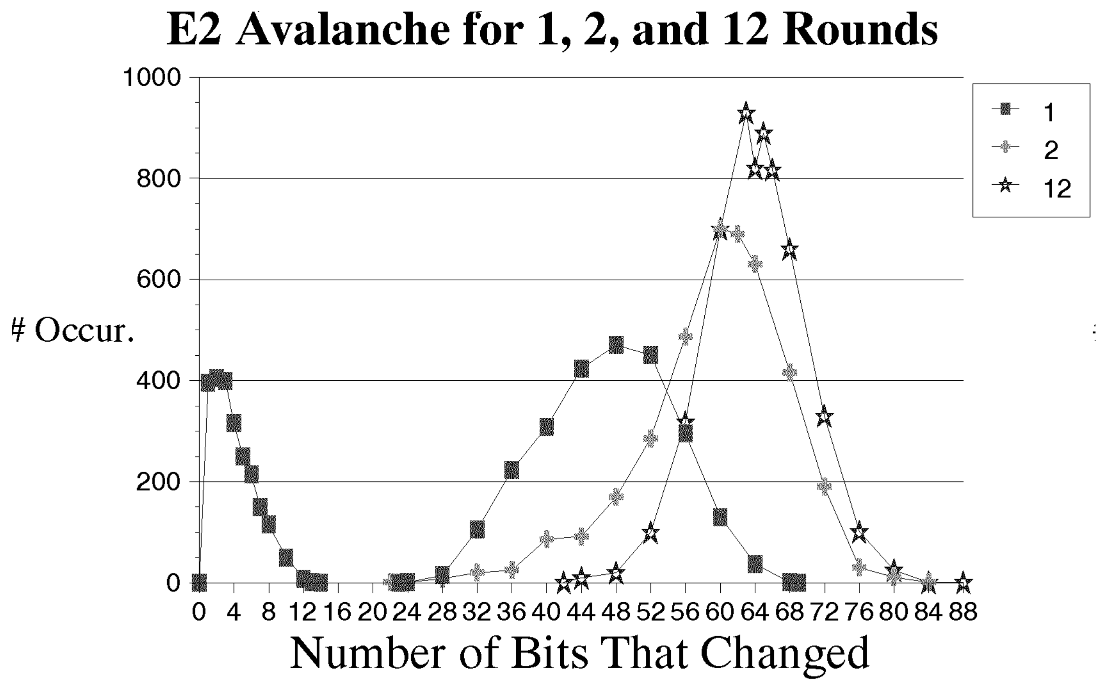

Figure 9

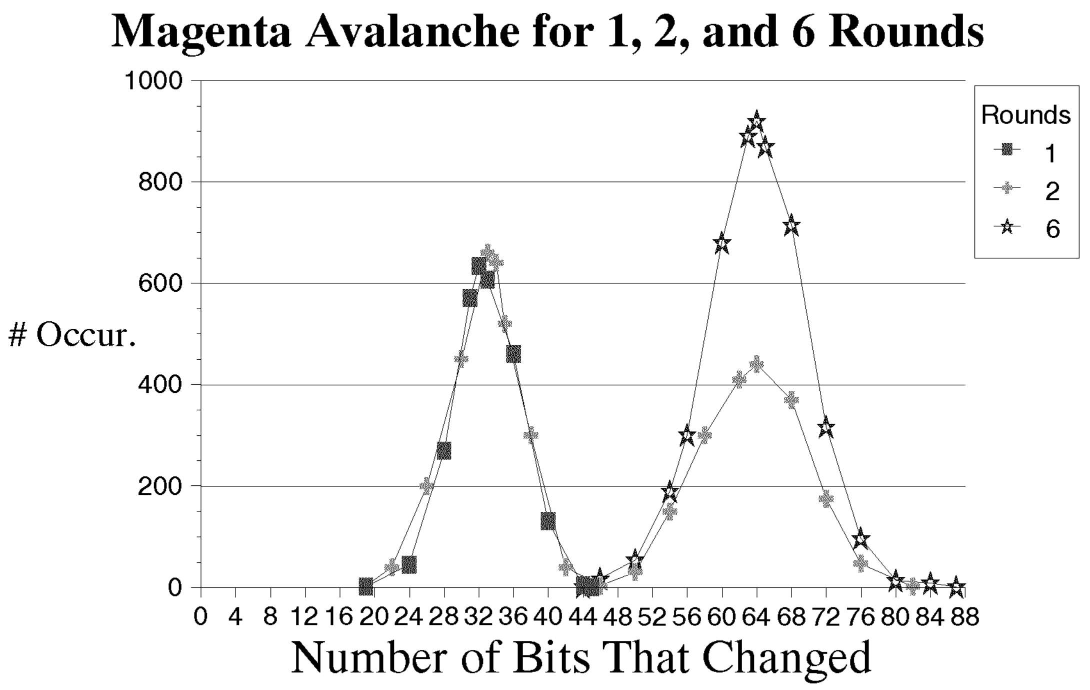

Figure 10

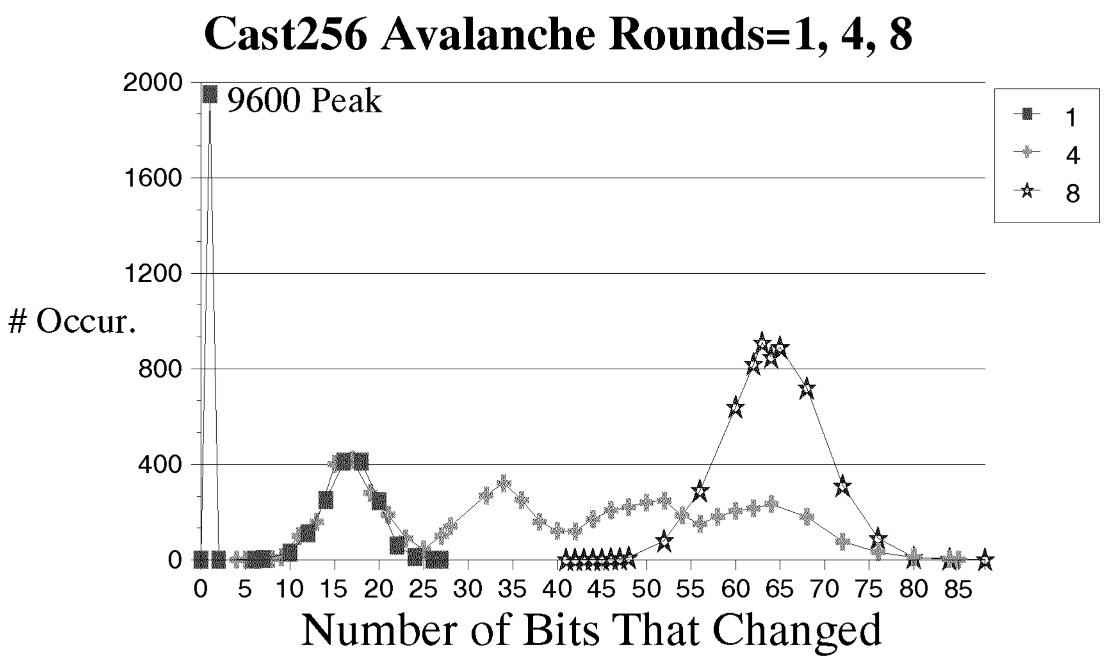

Figure 11

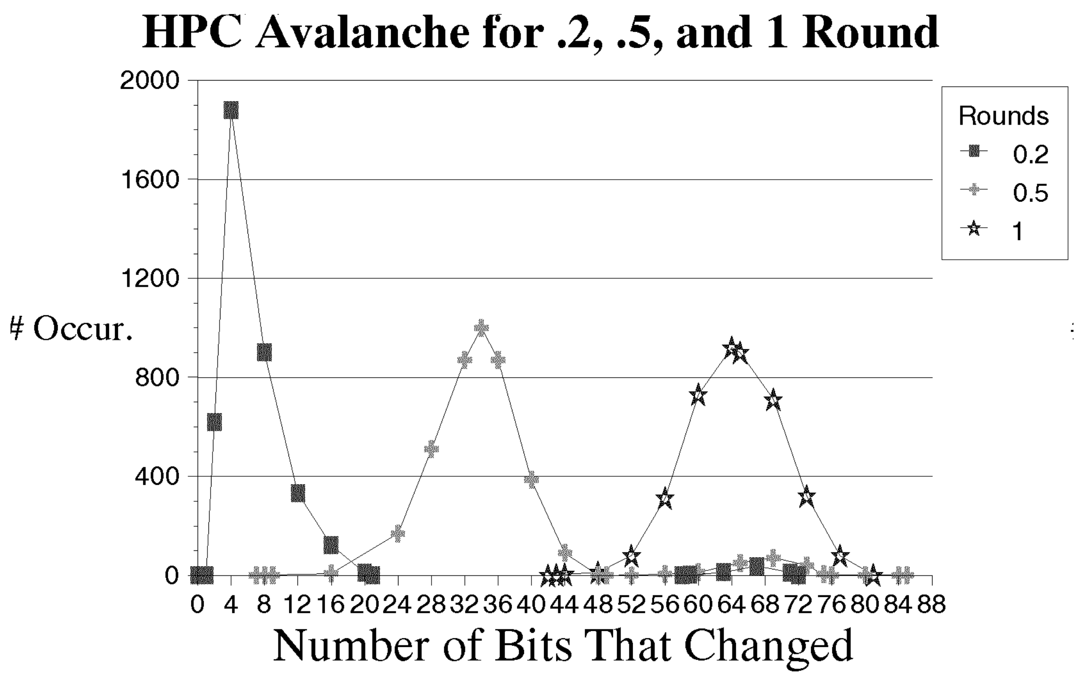

Figure 12

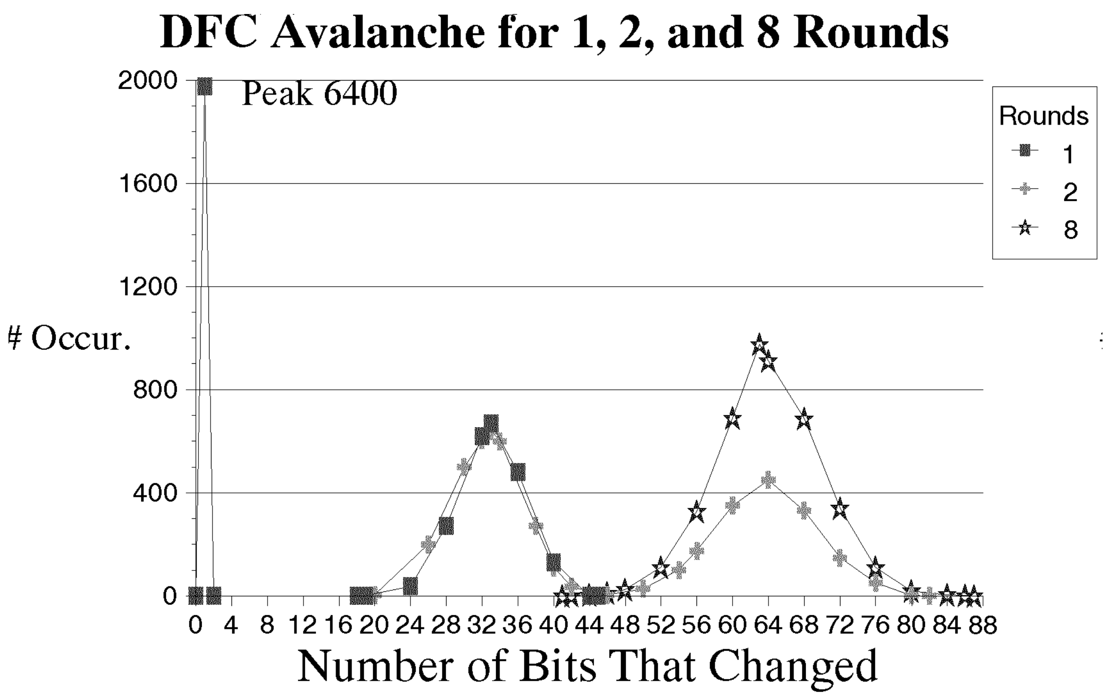

Figure 13

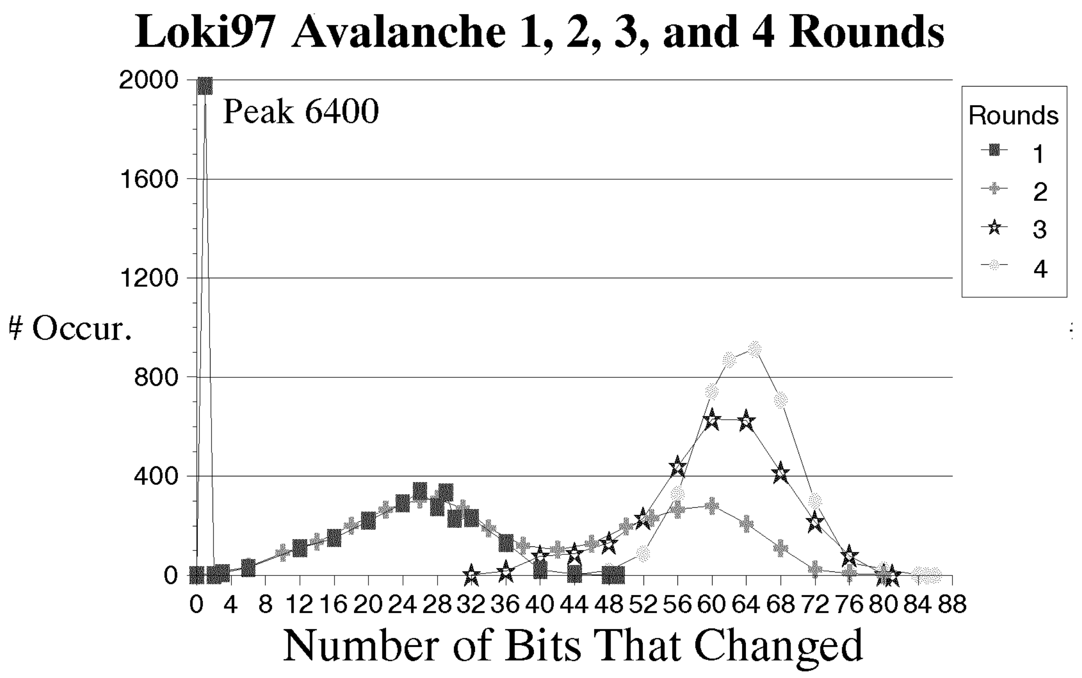

Figure 14

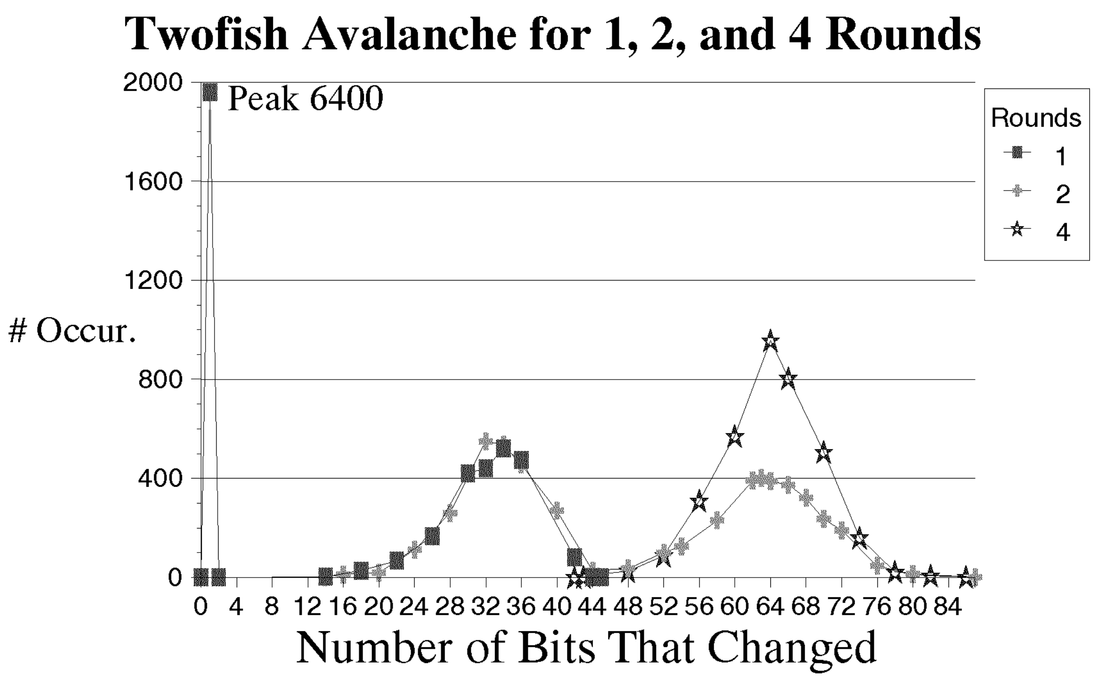

Figure 15

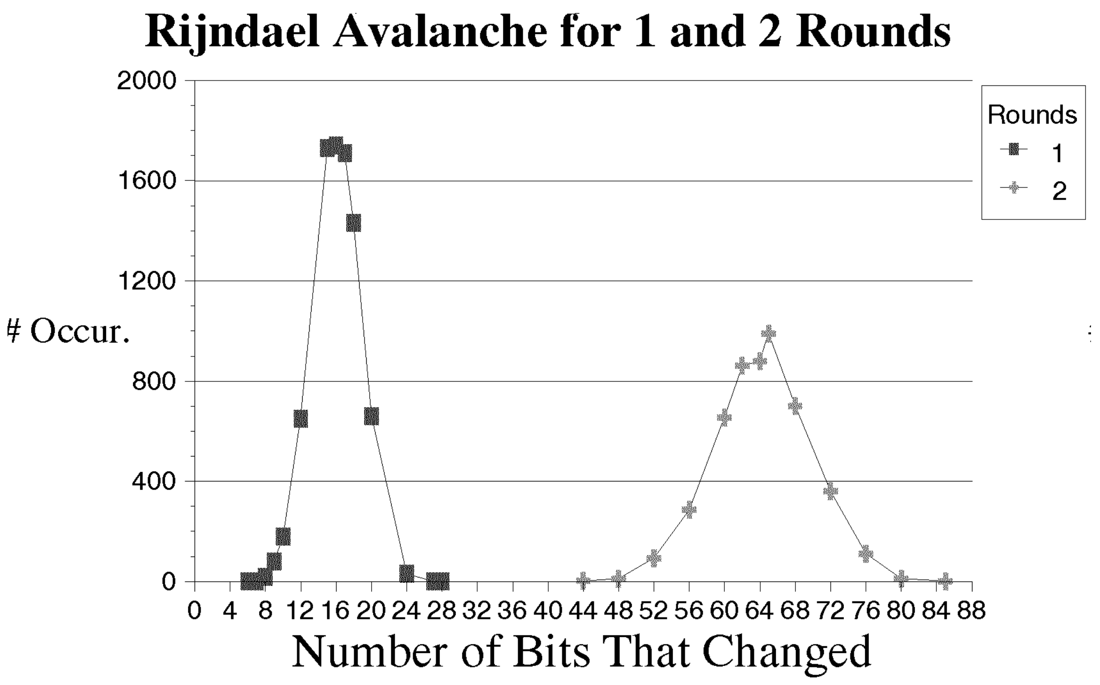

Figure 16

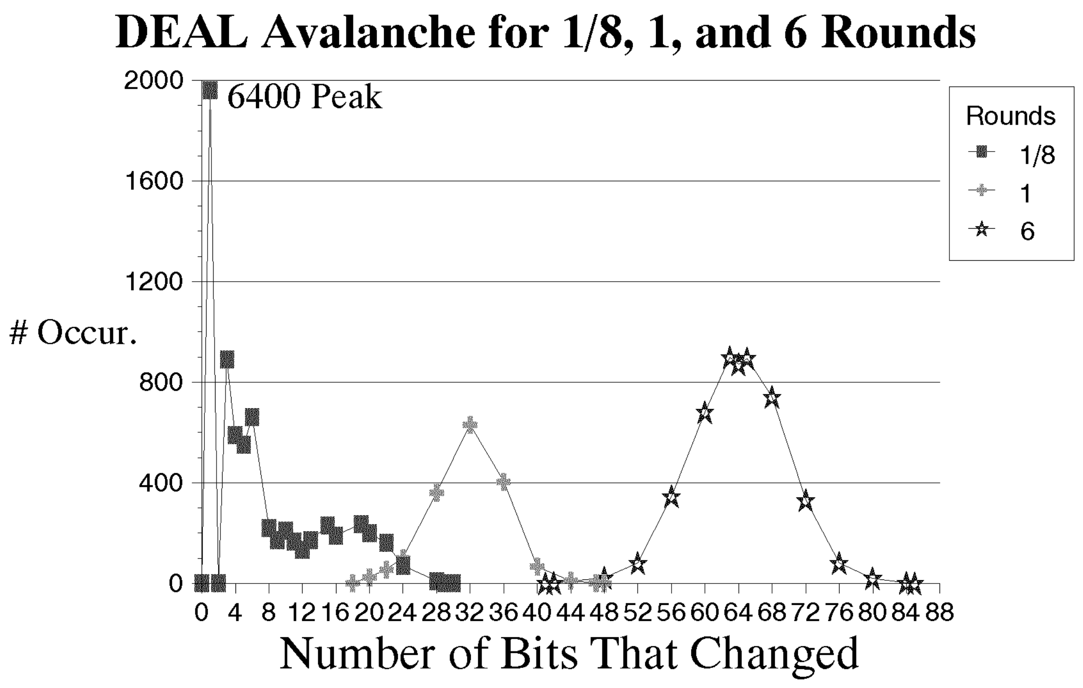

Figure 17

Acknowledgments

The author wishes to thank the engineers and management of Sun Microsystems, Inc. who reviewed this paper and who approved of its publication.

References

1. B. Schneier (1996) *Applied Cryptography,* Second Edition, Wiley, page 273.