A Generalised Testbed for Analysing Block and Stream Ciphers

Lawrence BROWN Josef PIEPRZYK Reihaneh SAFAVI-NAINI Jennifer SEBERRY

Centre for Computer Security Research,
Department of Computer Science,
University College, UNSW, Australian Defence Force Academy,
Canberra ACT 2600. Australia.

Abstract

With the recent development of a number of new ciphers, especially block ciphers, there is a need for a set of tools to help analyse them, in order to obtain some comparative measure of their relative security, and to assist in identifying any shortcomings in their design. This project uses a number of tests to provide a better determination of a cipher's capabilities than previous attempts, and incorporates them into a framework to aid extension of the testbed, through both the addition of new ciphers, and new tests. The testbed will be used for a comparative analysis of some of the new families of block ciphers, including LOKI, FEAL, Khufu, Khafre, and KSX against the older generation which includes Lucifer and DES. Some preliminary results from this analysis on DES, FEAL, and LOKI are presented here.

1 Introduction

With the increasing need for new ciphers for use in new communications systems, a number of new ciphers, particularly block ciphers, have been developed. There is a need for a set of tools to help analyse these ciphers, in order to obtain some comparative measure of their relative security, and to assist in identifying any shortcomings in their design. Previous attempts at building such a set of tools have generally concentrated on using simple statistical tests on a bit stream produced by the cipher (cf. [1]). This project uses a number of additional tests to provide a better determination of a cipher's capabilities and incorporates them into a framework to aid extension of the testbed, both through the addition of new ciphers, and new tests. The tests used fall into two broad classes: statistical tests; and structural tests. Statistical tests analyse the statistical and complexity properties of a stream of symbols (each symbol being a fixed number of bits), independent of knowledge of their origin. Tests suitable for use include the well known tests for randomness, the Berlekamp-Massey algorithm to measure linear complexity and linear complexity profile measures, and the Ziv complexity test. Structural tests analyse aspects of a particular cipher's structure. Structural tests in use for Feistel style block ciphers include a generalised form of Meyer's analysis of ciphertext dependence on plaintext and key, and an analysis of avalanche propagation in the cipher. There are also several tests which evaluate any substitution boxes used in these ciphers against some known design criteria, and which evaluate their non-linearity.

In addition to developing these tools, a pair of frameworks has been designed to provide an overall framework in which to integrate the tools, and to assist in incorporating new ciphers or new tools into the system. The framework for the statistical tools is shown in Fig. 1, whilst the framework for the structural tools is shown in Fig. 2. These frameworks assume that the tools are separate programs or interpreted scripts, with a standard set of interfaces between them. This interface is either a bit stream represented as a series of ASCII 0 and 1s, or a tagged line to represent parameters or test results.

The testbed will be used for a comparative analysis of some of the new families of block ciphers, including LOKI, FEAL, Khufu and Khafre, and KSX against the older generation which includes Lucifer and DES.

2 Testbed Framework

The testbed framework provides a means by which a number of tools, each concentrating on a different aspect, or a different cipher, may be integrated into a common system. It defines the interfaces between the various tools at various stages in the framework. There are two broad catagories of tools used in analysing ciphers: statistical tests, which analyse the statistical properties of a stream of bits; and structural tests; which analyse some aspects of a ciphers structure.

2.1 Statistical Tests

Statistical tests may be applied to any stream of bits, whether generated by a stream cipher, or by a block cipher used in a stream mode. Such tests

Tag	Usage
K:	a 64-bit key for the cipher used in the stream generator
IV:	a 64-bit inital (or working) value for the stream generator
K128:	a 128-bit key for the cipher used in the stream generator
IV128:	a 128-bit inital (or working) value for the stream generator
ss:	chi square values from the streamstat program
Lk:	a linear complexity value from the berle proogram
LCP:	a linear complexity profile from the berle program

Table 1: Tags Used By the Statistical Test

are used to analyse the properties of the stream in terms of how random, and how cryptographically secure, the stream is (ignoring for the present the arguments over the definitions of these words). The testbed framework for the statistical tools is shown in Fig. 1.

The interfaces between the random key generator and the stream generator, and between the stream analysis and summary tools use a common tagged line format. This consists of lines of information, each starting with a known tag identifying the information on the rest of the line. The tags currently in use are listed in Table 1.

The interface between the stream generator and the analysis tools is a stream of bits, represented as ASCII '0' or '1' characters, with one stream per line. This permits a number of streams to be analysed in a single session.

In more detail, the function of each of the stages, and the currently available tools include:

Random Key Generator The random key generator program is used to randomly create a large number of keys which may be used to control a subsequent stream generator program. The current program uses the DES cipher in OFB mode. Given a key and an initial value, it will generate the specified number of random keys (of either 64 or 128-bits in size). This

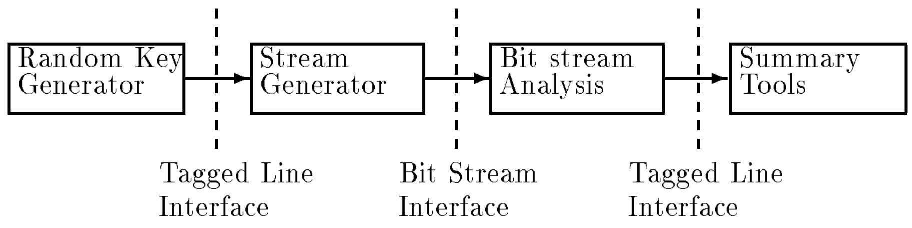

Figure 1: Statistical Testbed Framework

method has the advantage that the same set of keys can be regenerated at any time given the same initial value and key. This permits comparisons of several ciphers to then be conducted, based on the same set of random key values.

Stream Generators The Stream Generators are programs which, given a specific key (and possibly other cipher dependent parameters), and a stream length, generate a stream of bits. These programs use a standard program shell. This supports a standard command-line interface, and also is able to read arguments such as new inital values or keys from the standard input. This permits a series of streams to be generated for analysis in a single session. This shell is customised simply by changing the cipher library routines called. This allows new stream generators to be created very easily, once a new set of cipher library routines are available. The stream generators currently in use run the relevant cipher in OFB mode. The ciphers supported are:

desrnd ANSI standard DES;

lokirnd CCCSR LOKI cipher [2];

fealrnd NTT FEAL cipher [3], [4];

luciferrnd IBM's original Lucifer cipher [5];

Bit Stream Analysis Tools The Bit Stream Analysis tools take as input a stream of bits, and calculate some statistical properties of it. Tools available at present, now adapted for the testbed environment, include:

streamstat This program calculates the well known statistical tests for randomness on a stream from [6], [7], including the frequency, serial, poker (on blocks of 3,4, and 8 bits), run and gap tests. It forms a tagged output for the next stage consisting of the number of bits scanned and the chi-square values for each of these tests.

berle This program performs a linear complexity analysis on a stream [8], [9]. It can produce tagged lines of either the linear complexity, or both the linear complexity and the linear complexity profile for summarising by the next stage. This analysis is computationally expensive, and hence may be used only on fairly short stream lengths.

Summary Tools These tools accumulate the results from a series of streams generated from different keys, and calculate some overall statistics for each of the individual statistics measured by the previous stage.

A number of small scripts, tailored for specific tests, are used to accumulate these results. Alternatively, the tagged output can be read into a statistical package such as S [10] for further analysis and graphing.

The utilities available in the testbed at present are:

sstream summarises the results from the streamstat analysis program. The current script calculates, for each test, the mean and range and the percentage of results that fall under the 5% or over the 95% confidence levels. These provide an indication of how well the results of each test matches the expected distribution.

sberle summarises the results from the berle analysis program. The current script calculates, for each test, the mean and range of the results.

2.2 Structural Tests

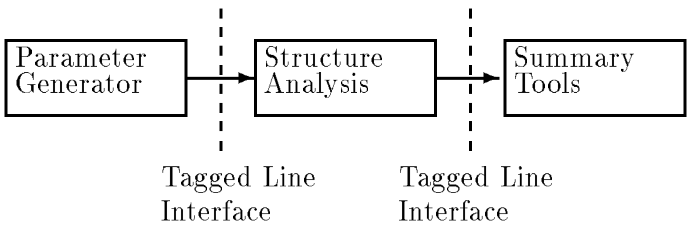

Figure 2: Structural Testbed Framework

Structural tests analyse the particular structure of a cipher. As such, they are specific to each cipher (or perhaps a family of ciphers with an identical structure save for variations in some permutation blocks), and need to be rewritten for each new cipher.

The testbed framework for the structural tools is shown in Fig. 2.

The interfaces between the parameter generator and structure analysis, and between the structure analysis and summary tools use a common tagged line format. This consists of lines of information, each starting with a known tag identifying the information on the rest of the line. The tags currently in use are listed in Table 2.

Parameter Generators These programs generate sets of parameters used to create families of related ciphers. At present the main generators, used in analysing Feistel style ciphers, are for permutations P and PC2, and possibly key schedules. These are used for testing variants of existing schemes,

Tag	Usage
P:	a 32-bit permutation P
PC2:	a 32-bit permutation $PC2$
KS:	a DES style key rotation schedule
CPdep:	a ciphertext dependence on plaintext analysis
CKdep:	a ciphertext dependence on key analysis

Table 2: Tags Used By the Structural Test

especially the DES, and DES like schemes, including LOKI. They build the permutations according to some rules, and then supply them to the structure analysis programs.

A generator can also be used to control different variants of the LOKI S-boxes for analysis of their relative merits.

Programs currently available include:

desp builds DES style permutation P using the rules from [11]

lbgenp builds DES style permutations P using latin-square criteria [12]

regp builds DES style permutations P using a regular structure based on a difference function from [12]

despc2 builds DES style permutations $PC2$ using a regular distribution with excluded bits interspersed by the rules identified in [13]

As well as these generators, a number of files of standard, worst case, or example permutation and key rotation schedules are available. These may be combined to form all combinations using a special tool:

comb form all combinations of permutations in up to three files

Structural Analysis Tools The currently available structural analysis tools fall into two major catagories.

Meyer's Dependency Analysis These programs analyse the overall cipher structure, using Meyer's ciphertext dependency on plaintext and key [14] as a measure of effectiveness. They assume that the S-boxes are well formed, and they examine how changes in the choice of the permutations P and PC2, and the Key Schedule KS, affect the growth of the avalanche effect in the cipher [12], [13]. These programs have to be tailored for a specific cipher framework. At present the testbed includes programs for DES and LOKI. Also available, but not yet integrated into the testbed are programs for the KSX DES extension, and for a 128-bit variant of the DES. The testbed programs are:

des_cp measures the ciphertext dependence on plaintext for DES

des_ck measures the ciphertext dependence on key for DES

loki_cp measures the ciphertext dependence on plaintext for LOKI

loki_ck measures the ciphertext dependence on key for LOKI

S-Box Analysis These programs examine the effectiveness of the S-boxes in fulfilling known design criteria. The main suite of programs measure the degree to which the S-boxes meet the criteria identified in the original NBS report on DES [11]. These use a general program shell, which enables new S-boxes to be tested by supplying suitable routines and building a new test program. The tests have been extended to supply the XOR profile used by Shamir's differential cryptanalysis of ciphers [15].

The existing S-box testing programs are:

des_s analyses the DES S-boxes.

loki_s analyses the LOKI S-boxes (standard and variant forms).

feal_s analyses the FEAL S-boxes.

lucifer_s analyses the Lucifer S-boxes.

Summary Tools These tools accumulate statistics from analysing the results for a number of structural tests on ciphers in a given family. Again, as for the statistical tools, these results can be loaded into a statistical package for further analysis.

scandep summarises dependency results from any of the dependency analysis programs.

2.3 Overall Controlling Programs

As well as the various components of the testbed, some control programs have been written to assist in performing particular analysis tasks with the testbed. These include:

Tbed This is the general testbed control utility. It prompts for a particular set of tests which are wanted, eg; doing a series of stream statistics on LOKI, and then sets the relevant tools running (in this case, for a month).

lokiS This is a specialised tool for running a series of tests on the LOKI S-boxes in a single session. It uses all of the options provided by the loki_s program to supply an overall profile of the LOKI S-box. A similar tool for the DES S-boxes is yet to be integrated into the testbed.

Cipher	DES			FEAL			LOKI		
Range	min	av	max	min	av	max	min	av	max
Dist	$< 5\%$	$5-95\%$	$> 95\%$	$< 5\%$	$5-95\%$	$> 95\%$	$< 5\%$	$5-95\%$	$> 95\%$
Num Trials		1000			1000			8161	
Num bits		640000			640000			6400000	
Freq	0.00	0.98	10.73	0.00	0.94	13.65	0.00	0.98	17.32
	4.5	90.2	5.3	6.1	89.8	4.1	4.8	90.6	4.6
Serial	-0.01	1.03	13.44	-0.01	1.05	13.79	0.00	0.99	16.10
	4.4	91.1	4.5	4.2	89.4	6.4	4.5	90.5	4.9
Poker 3	0.27	7.03	25.99	0.62	6.92	22.73	0.39	6.98	30.22
	6.1	87.9	6.0	5.0	90.0	5.0	4.9	90.0	5.2
Poker 4	3.11	14.94	41.23	3.50	14.94	36.30	1.78	14.99	45.67
	5.0	90.0	5.0	4.8	91.0	4.2	5.0	89.7	5.3
Poker 8	182.54	255.38	337.60	193.96	256.55	325.71	182.99	254.98	357.20
	4.8	90.2	5.0	4.6	90.2	5.2	4.9	90.2	5.0
Gaps	3.85	16.8	44.35	4.00	17.06	45.94	2.64	15.65	49.84
	1.8	88.5	9.7	2.3	86.6	11.1	3.9	89.7	6.4
Runs	2.76	16.69	44.85	4.64	16.75	40.03	3.00	15.71	44.48
	2.8	87.6	9.6	2.1	88.8	9.1	3.7	89.8	6.5

Table 3: Streamstat Chi-square values for DES, FEAL and LOKI

Cipher	DES			FEAL			LOKI		
Test	min	av	max	min	av	max	min	av	max
Num Trials		1000			1000			8161	
Num bits		960			960			960	
Berle	477	480.29	485	476	480.15	484	477	480.16	483

Table 4: Berle results for DES, FEAL and LOKI

3 Initial Results from the Testbed

3.1 Some Comparative Stream Statistics

A number of trials have been run to obtain some initial comparative results for the DES, FEAL, and LOKI ciphers. These results are summarised in Tables 3 and 4 for the streamstat and berle results, respectively.

Note the similarity in profiles between the LOKI and the DES results. The most noticable difference is in the skew of the distributions for the gap and run tests. This was also observed when a similar small set of tests was done on LOKI and is believed to be due to the stream length being too short in these examples (these tests scan for up to 15 bit runs and gaps). As may be seen from the longer length LOKI tests, the results here also appear to match the expected random profile.

These results have also been loaded into S, and QQ plots made to compare the DES and FEAL, and DES and LOKI distributions for each of the statistics calculated (see [10] for details). These plots indicate a fairly good match, suggesting that all of these ciphers produce bits with the same profile. Examples of these plots are given in Fig. 3.

3.2 XOR Profile of DES and LOKI

A key component of Shamir's new differential cryptanalysis technique is the XOR profile of the S-boxes used in a cipher. This profile consists of taking all possible pairs of inputs and plotting a distribution of the XOR of the input values against the XOR of their output values. Any frequency values substantially above the average may then be used in the differential cryptanalysis process.

The following two tables (5, 6) present a Stem plot of the XOR profile for DES S-box 1, and the LOKI S-box. This plot represents the relative frequency of each value that occurs in the profile, with values greater than several standard deviations from the mean listed separately as high values. For more information on this plot see the S manual [10].

The most obvious difference between the two profiles is the much larger size of the LOKI profile. This makes any use of it computationally more expensive both in time and space. The next critical feature is the relative probablilities of the high values in any column. Ignoring the anomolous value in the [0,0] location for each (being 64 in the DES profile and 4096 in the LOKI one), the highest peak in the DES profile is 14, giving a probability of its occurance in the column of $Pr(14/64)$, ie; just under $Pr(1/4)$. In the case of LOKI there is a single high value of 256 (in the [1,1] location) with $Pr(1/16)$, with most of the other high values being around 70 with approx $Pr(1/58)$. These values are obviously much lower than in the DES case, and hence the differential cryptanalysis technique, which relies on these probabilities to skew the output distribution after a large number of trials, will require many more trials to be tested before an uneven distribution becomes obvious. This early analysis suggests that LOKI may be significantly less susceptible to the differential cryptanalysis method than DES.

Another aspect of the XOR profile worth examining is the profile of the

Figure 3: QQ-plots of DES vs FEAL and DES vs LOKI for the 4-bit Poker test

```
N = 1024   Median = 4
Quartiles = 2, 6
Decimal point is 1 place to the right of the colon
  210  210   0 : zzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzx
  456  246   0 : 22222222222222222222222222222222222222222222222222222222222222x
       232   0 : 44444444444444444444444444444444444444444444444444444444444444x
  336  168   0 : 66666666666666666666666666666666666666666666666666666666666666x
  168   84   0 : 88888888888888888888888888888888888888888888888888888888888888x
   84   46   1 : 0000000000000000000000000000000000000000000000
   38   24   1 : 222222222222222222222222
   14   12   1 : 444444444444
High:  16  64
```

Table 5: Stem plot of the XOR Profile of DES S-box 1

```
N = 1048576   Median = 16
Quartiles = 12, 20
Decimal point is 1 place to the right of the colon
   0 : zzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzx
   0 : 222222222222222222222222222222222222222222222222222222222222222222222222x
   0 : 444444444444444444444444444444444444444444444444444444444444444444444444x
   0 : 666666666666666666666666666666666666666666666666666666666666666666666666x
   0 : 888888888888888888888888888888888888888888888888888888888888888888888888x
   1 : 000000000000000000000000000000000000000000000000000000000000000000000000x
   1 : 222222222222222222222222222222222222222222222222222222222222222222222222x
   1 : 444444444444444444444444444444444444444444444444444444444444444444444444x
   1 : 666666666666666666666666666666666666666666666666666666666666666666666666x
   1 : 888888888888888888888888888888888888888888888888888888888888888888888888x
   2 : 000000000000000000000000000000000000000000000000000000000000000000000000x
   2 : 222222222222222222222222222222222222222222222222222222222222222222222222x
   2 : 444444444444444444444444444444444444444444444444444444444444444444444444x
   2 : 666666666666666666666666666666666666666666666666666666666666666666666666x
   2 : 888888888888888888888888888888888888888888888888888888888888888888888888x
   3 : 000000000000000000000000000000000000000000000000000000000000000000000000x
   3 : 222222222222222222222222222222222222222222222222222222222222222222222222x
   3 : 444444444444444444444444444444444444444444444444444444444444444444444444x
   3 : 666666666666666666666666666666666666666666666666666666666666666666666666x
   3 : 888888888888888888888888888888888888888888888888888888888888888888888888x
   4 : 000000000000000000000000000000000000000000000000000000000000000000000000x
   4 : 222222222222222222222222222222222222222222222222222222222222222222222222x
   4 : 444444444444444444444444444444444444444444444444444444444444444444444444x
   4 : 666666666666666666666666666666666666666666666666666666666666666666666666x
   4 : 888888888888888888888888888888888888888888888888888888888888888888888888x
   5 : 000000000000000000000000000000000000
   5 : 222222222222222222222222222222
   5 : 4444444444444444444
   5 : 666666666666
   5 : 8888
   6 : 0000
High:   62   62   62   62   64   64   64   64   64   66   66   68   68   68   70
High:   72   76   86  256 4096
```

Table 6: Stem plot of the XOR Profile of the LOKI S-box

first column, that is, with output XOR of 0. This column indicates those input pairs which have the same output XOR. Obviously, if the two input values are the same, the outputs will be the same. This leads to the anomalous value of 64 in DES, and 4096 in LOKI profiles at location [0,0]. Apart from this value, if any other of the values in the column are substantially above the average, then these particular values are of use in the cryptanalysis.

As in the general profile, the profile for this column shows that the probablility of getting specific high values is much lower in the LOKI profile than in the DES profile. This will also hinder any cryptanalysis method.

4 Conclusions

In this project we have developed the design framework for a testbed of tools needed to evaluate modern block ciphers. A number of tools have been designed within this framework, and these have been used to obtain some initial results on a comparison of the DES, FEAL and LOKI ciphers. These results show that these ciphers have similar statistical and bit propogation properties. The XOR-profile analysis though, shows a significant difference between the DES and LOKI S-boxes. A number of further tools, as well as more ciphers are to be added to the testbed in future.

5 Acknowledgements

Thank you to the members of the crypt group for their support and suggestions, and to Paul Stokes for his assistance with programming.

This work has been supported by ARC grant A48830241, ATERB, and Telecom Australia research contract 7027.

6 References

[1] H. Gustafson, E. Dawson and B. Caelli, “Comparison of Block Ciphers,” in *Advances in Cryptology: Auscrypt ’90* (Lecture Notes in Computer Science), vol. 453. Berlin: Springer Verlag, pp. , 1990.

[2] L. Brown, J. Pieprzyk and J. Seberry, “LOKI - A Cryptographic Primitive for Authentication and Secrecy Applications,” in *Advances in Cryptology: Auscrypt ’90* (Lecture Notes in Computer Science), vol. 453. Berlin: Springer Verlag, pp. , 1990.

[3] A. Shimizu and S. Miyaguchi, “Fast Data Encipherment Algorithm FEAL,” in *Eurocrypt 87 Abstracts*, 1988.

[4] S. Miyaguchi, "The FEAL Cipher Family," *Advances in Cryptology - Crypto'90*, 537, pp. , 1991.

[5] A. Sorkin, "Lucifer, a Cryptographic Algorithm," *Cryptologia*, 8, no. 1, pp. , Jan. 1984.

[6] H. Beker and F. Piper, *Cipher Systems: The Protection of Communications.* London, Northwood Books, 1982.

[7] D. E. Knuth, *Seminumerical Algorithms* (The Art of Computer Programming), vol. 2. London, Addison Wesley, 1981.

[8] H. van Tilborg, *An Introduction to Cryptology.* Mass., Kluwer Academic Press, 1988.

[9] R. A. Rueppel, *Analysis and Design of Stream Ciphers.* Berlin, Springer Verlag, 1986.

[10] R. A. Becker and J. M. Chambers, *S An Interactive Environment for Data Analysis and Graphics* (Wadsworth Statistical/Probability Series). Belmont, CA, Wadsworth, 1984.

[11] L. Brown, "A Proposed Design for an Extended DES," in *Computer Security in the Age of Information*, W. J. Caelli, Ed. Amsterdam: North-Holland, 1989.

[12] L. Brown and J. Seberry, "On the Design of Permutation P in DES Type Cryptosystems," in *Advances in Cryptology - Eurocrypt'89* (Lecture Notes in Computer Science), vol. 434, J. J. Quisquater and J. Vanderwalle, Eds. Berlin: Springer Verlag, pp. , 1990.

[13] L. Brown and J. Seberry, "Key Scheduling in DES Type Cryptosystems," in *Advances in Cryptology: Auscrypt '90* (Lecture Notes in Computer Science), vol. 453. Berlin: Springer Verlag, pp. , 1990.

[14] C. H. Meyer and S. M. Matyas, *Cryptography: A New Dimension in Data Security.* New York, John Wiley Sons, 1982.

[15] E. Biham and A. Shamir, "Differential Cryptanalysis of DES-like Cryptosystems," Weizmann Institute of Science, Rehovot, Israel, Technical Report, 19 July 1990.