A statistical test for IPES

Citation for published version (APA):
A STATISTICAL TEST FOR IPES

Marcel van Asperdt

September 1992
A statistical test for IPES

by Marcel van Asperdt

Feb. '92 - April '92, ETH Zürich
A statistical test for IPES

Table of contents

1. Introduction .................................................................................................. 1
2. Test description ............................................................................................ 1
3. Random mappings ........................................................................................ 1
4. Expectation for the test .............................................................................. 4
5. Proposal for test improvement ..................................................................... 5
6. Test of IPES ............................................................................................... 6
7. Conclusions ................................................................................................. 8
References ......................................................................................................... 8
Appendix A ...................................................................................................... 9
Appendix B ...................................................................................................... 16
Appendix C ...................................................................................................... 24
1. Introduction

IPES stands for Improved Proposed Encryption Standard, it is an encryption algorithm that takes 64 bits as input, producing 64 bits of output using a key of 128 bits. For the encryption process (and matching decryption process), see (1). The aim of this report is to describe one possible statistical test for encryption algorithms, in particular IPES. First of all the test will be described in "Test description". Then a section on random mappings is included. Next the expectations for the test are derived in "Expectation for the test". In Section 5, an improvement for the test is proposed. The results of the test and the improved version can be found in Section 6. Finally we state the conclusions.

In Appendix A the used C-program can be found. In Appendix B all test results for IPES are included. In Appendix C the table for the statistical test is shown.

2. Test description

A theoretical description of the test can be found in (2). We will restrict ourselves to a more formal description. Choose a random input of 64 bits and a random key of 128 bits. Select L (1 ≤ L ≤ 64) bits of the output, say F_o. Choose also L bits of the input, say F_i. Pick a permutation of L symbols, say p.

Now we can start the test. Encrypt the random chosen input. The bits F_o of the resulting output are permuted according to p and fed in in the input at positions F_i. Repeat this process until we find a certain bitpattern in positions F_o for the second time. This will happen; we only have 2^L possibilities available for the output in positions F_o. Look at the sequence of output bits in positions F_o. We will find a cycle in this sequence, the length of this cycle is called the cycle length. The length of the sequence "to come in" this cycle is called the tail length.

We can view this encryption process as a mapping of 2^L elements into itself. We expect this mapping to be a random mapping. Test for all sets F_o and F_i (with #F_o = #F_i = L) and for all permutations whether this mapping is random or not. Do this test for all L with 1 ≤ L ≤ 64.

3. Random mappings

As will become clear in Section 4, we have to distinguish between random mappings from a set of n elements into itself and onto itself.
Let $T$ be a random mapping from a set of $n$ elements into itself. We can represent $T$ as a directed graph as follows; the nodes are the elements of $F_n$, the edges $\{(x, T(x)) \mid x \in F_n\}$, wherein $F_n = \{1, 2, \ldots, n\}$.

Example: $n=4 \quad T_1=3 \quad T_2=1 \quad T_3=1 \quad T_4=4$

![Graph of $T$]

By induction we define the following notation:

$$
\begin{align*}
T^{(0)} x &= T x \\
T^{(k)} x &= T(T^{(k-1)} x) \\& \text{for all } x \in F_n \text{ and } k \in \mathbb{N}.
\end{align*}
$$

Let $x$ be a random element of $F_n$ to start with. What probability do we have to find a cycle of length $\lambda$ and a tail of length $\mu$?

$$
P(\text{cycle length } = \lambda \text{ and tail length } = \mu) =
\begin{align*}
&= P(T^{(0)} x \neq x, T^{(1)} x, \ldots, T^{(r-1)} x \mid 0 < r < \lambda + \mu, \ T^{(\lambda+\mu)} x = T^{(0)} x) \\
&= \binom{n-1}{\lambda + \mu - 1} (\lambda + \mu - 1)! n^{n-\lambda-\mu} / n^* \\
&= \frac{(n-1)!}{(n-\lambda-\mu)! n^{\lambda+\mu}}
\end{align*}
$$

How to understand the formula marked with $*$?

$x$ is given, so we have to choose $T^{(0)} x, T^{(1)} x, \ldots, T^{(\lambda+\mu-1)} x$ different out of $F_n \setminus \{x\}$. This yields $\binom{n-1}{\lambda + \mu - 1}$ possibilities. Placing them in order yields $(\lambda+\mu-1)!$ possibilities. To make this a complete mapping we have to attach to the resulting $n-\lambda-\mu$ elements an image, $n^{n-\lambda-\mu}$ possibilities. Now we have a mapping, the total number of possible mappings is $n^n$. This explains formula $*$. 

2
So we find:

\[
P(\text{cycle length} = \lambda) = \sum_{\mu=0}^{n-\lambda} P(\text{cycle length} = \lambda \text{ and tail length} = \mu)
\]

\[
= \sum_{\mu=0}^{n-\lambda} \frac{(n-1)!}{(n-\mu)! \, n^\mu}
\]

\[
= \sum_{\lambda=\mu+1}^{n} \frac{(n-1)!}{(n-\lambda)! \, n^\lambda}
\]

Of course this result is only valid for \( \lambda > 0 \). If \( \lambda = 0 \) we can't speak of a cycle, so the probability is zero.

\[
P(\text{tail length} = \mu) = \sum_{\lambda=\mu+1}^{n} \frac{(n-1)!}{(n-\lambda-\mu)! \, n^{\lambda+\mu}}
\]

\[
= \sum_{\lambda=\mu+1}^{n} \frac{(n-1)!}{(n-\lambda)! \, n^\lambda}
\]

Note that this result is also valid for \( \mu = 0 \), a tail of length zero means that we start in a cycle.

Now we will consider random mappings from a set of \( n \) elements onto itself. The first observation we make is that the tail length is always zero. Every element \( y \) has an element \( x \) such that \( Tx = y \). So we only have to consider the cycle length.

Let \( x \) be a random element of \( F_n \) to start with. What is the probability that \( x \) is contained in a cycle of length \( \lambda \)?

\[
P(\text{cycle length} = \lambda) = \frac{\binom{n-1}{\lambda-1} (\lambda-1)! (n-\lambda)!}{n!}
\]

\[
= \frac{1}{n} \quad \text{(**)}
\]

How to understand formula **? 

\( x \) is given, choose \( \lambda-1 \) elements from \( F_n \setminus \{x\} \), this yields \( \binom{n-1}{\lambda-1} \) possibilities. Now we have \( \lambda \) elements. Placing them in order yields \( (\lambda-1)! \) possibilities. The other \( n-\lambda \) elements have to get an image, \( (n-\lambda)! \) possibilities. Now we have a random mapping from \( F_n \) onto
\( F_n \). The total number of possible random mappings from \( F_n \) onto \( F_n \) is \( n! \). This explains formula \(*\).

So we find:

\[ P(\text{cycle length} = \lambda) = \frac{1}{n} \text{ for } 1 \leq \lambda \leq n. \]

4. Expectation for the test

As described in Section 2 we want to test whether the test on the cycle length and tail length yields the same result as a random mapping or not. Therefore we have to postulate what we expect as result, the hypothesis we want to test. So we have to give:

\[ P(\text{cycle}(I, F, F_O, p) = \lambda) \text{ and } P(\text{tail}(I, F, F_O, p) = \mu), \]

wherein \( F, F_O, p \) are as defined in Section 2 and \( I \) is the random chosen input of 64 bits.

Now we have to distinguish between the \( L \)-values, \( 1 \leq L \leq 63 \) and \( L = 64 \). In the first case the encryption process is a mapping of \( F_{2L} \) into \( F_{2L} \). For \( L = 64 \) (thus the whole output is taken as input), the encryption process is onto because it is invertible.

Before stating the expectations using the results of Section 3 we have to make a remark on the tail lengths. In our test we choose \( x \) at random, we perform the encryption process to derive our first (!) output. So we don't start counting our tail length at \( x \) but at \( T_x \), see figure.

![Figure: An input with tail length 3 and cycle length 4.](image)

A consequence of this feature is that we have two different situations in which we will find a tail length zero. Namely if the tail length counted from \( x \) is zero or one. From now on we will call the tail length counted from \( x \) the "modified tail length".
So we find the following probability distributions according to Section 3:

For $1 \leq L \leq 63$:

$$
P(\text{cycle}(I, F_I, F_O, p) = \lambda) = \begin{cases} 
0 & \text{for } \lambda = 0 \\
\sum_{k=0}^{2^L-1} \frac{(2^L-1)!}{(2^L-k)! \ 2^{Lk}} & \text{for } \lambda > 0
\end{cases}
$$

$$
P(\text{tail}(I, F_I, F_O, p) = \mu) = \begin{cases} 
\sum_{k=0}^{2^L} \frac{(2^L-1)!}{(2^L-k)! \ 2^{Lk}} \cdot \frac{1}{2^L} & \text{for } \mu = 0 \\
\sum_{k=\lambda+2}^{2^L} \frac{(2^L-1)!}{(2^L-k)! \ 2^{Lk}} & \text{for } \mu > 0
\end{cases}
$$

Remark: $P(\text{cycle}(I, F_I, F_O, p) = \mu+2) = P(\text{tail}(I, F_I, F_O, p) = \mu)$ for $\mu > 0$

For $L=64$:

$$
P(\text{cycle}(I, F_I, F_O, p) = \lambda) = \frac{1}{2^L}, \text{ for } 1 \leq \lambda \leq 2^L
$$

$$
P(\text{tail}(I, F_I, F_O, p) = 0) = 1
$$

Now we know what to expect as result for the test. But before we can start carrying out the test we will introduce an improvement for the test.

### 5. Proposal for test improvement

As we already saw in Section 3, it is not so difficult to derive a probability distribution for the tail length and cycle length together. So we can test on the simultaneous probability distribution of the tail length and cycle length together. This distribution is:

$$
P(\text{tail}(I, F_I, F_O, p) = \mu \text{ and } \text{cycle}(I, F_I, F_O, p) = \lambda) =
$$

$$
= \begin{cases} 
\frac{(2^L-1)!}{(2^L-\lambda)! \ 2^{L\lambda}} + \frac{(2^L-1)!}{(2^L-\lambda-1)! \ 2^{L(\lambda+1)}} & \text{for } \mu = 0 \\
\frac{(2^L-1)!}{(2^L-\lambda-\mu-1)! \ 2^{L(\lambda+\mu+1)}} & \text{for } \mu > 0
\end{cases}
$$

This distribution is valid for $1 \leq L \leq 63$. For $L=64$ nothing changes, because the tail length is always zero. Therefore we won't mention this case from now on.

The disadvantage of not starting a tail in $x$ but in $T$ (see Section 4) is that we don't test the probability to find a tail of length 0 or 1, but we test on the sum of these probabilities. Therefore we propose a modification of the test, namely to test on the modified tail...
length. The exact test specification of this modified test can easily be derived from the original test by defining $\Omega[0] := \text{Proj}(I, F_O)$, see (2).

The expected probability distribution is:

$$P(\text{mod. tail}(I, F_I, F_O, p) = \mu \text{ and cycle}(I, F_I, F_O, p) = \lambda ) = \frac{(2^L - 1)!}{(2^L - \lambda - \mu)! 2^{L(\lambda + \mu)}}$$

In the next section both the original test and the modified test will be carried out.

6. Test of IPES

A computer program to carry out the tests was written. The results of the tests can be found in Appendix B. For example the results for one case (see table) will be discussed below.

<table>
<thead>
<tr>
<th>cycle length</th>
<th>mod. tail length</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>2210627</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1447402</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>692021</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>164547</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1619391</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>695310</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>167569</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>777095</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>166164</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>188386</td>
</tr>
</tbody>
</table>

*Table: Results for $L=2$ as given in Appendix B*

First we will show how to translate the results to the results for the original test. To get a table for the cycle lengths one only has to sum the values for each value of the cycle length. For the tail lengths one also has to sum the values for the tail lengths, for the tail length zero, one has to sum the values of the tail lengths zero and one. Doing this we get:

<table>
<thead>
<tr>
<th>cycle length</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4514597</td>
</tr>
<tr>
<td>2</td>
<td>2482270</td>
</tr>
<tr>
<td>3</td>
<td>943259</td>
</tr>
<tr>
<td>4</td>
<td>188386</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>tail length</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>7104375</td>
</tr>
<tr>
<td>1</td>
<td>859590</td>
</tr>
<tr>
<td>2</td>
<td>164547</td>
</tr>
</tbody>
</table>

The expected values for the original test can be calculated with the formulas from Section 4 and the knowledge that the total number of observations is 8128512. See the next table.
To test the results we will use a Pearson's $\chi^2$-test for goodness of fit, see (3). First we will explain this test.

Let $x_1, x_2, ..., x_n$ be $n$ independent samples of a random variable $X$ that takes on $J$ possible values, say 1, ..., $J$. To test the hypothesis $H_0 : P(X=j) = p_j$, $j=1, 2, ..., J$ one forms the statistics:

$$X^2 = \sum_{j=1}^{J} \frac{(n_j - np_j)^2}{np_j},$$

where $n_j = \# \{x_i \mid x_i = j \}$.

For $n$ large, $X^2$ has approximately the $\chi^2$-distribution with $J-1$ degrees of freedom if the hypothesis $H_0$ is true. For the table with the $\chi^2$-distribution, see Appendix C.

Applying this Pearson's $\chi^2$-test for goodness of fit in the above case yields the values $X^2(\text{cycle}) = 134.8$ and $X^2(\text{tail}) = 14637.7$, we immediately see that we have to reject our hypothesis in this case, because $P(X^2(3) < 16.3) = 0.999$ and $P(X^2(2) < 13.8) = 0.999$ (the numbers between quotes are the degrees of freedom, resp. for the cycle and tail length).

For the modified test we first calculate the expected numbers using the formulas from Section 5.
The Pearson's $\chi^2$-test for goodness of fit yields a value $X^2 = 47510.9$ in this case. If our hypothesis is true we have $P(\chi^2 < 27.88) = 0.999$ (see Appendix C with 9 degrees of freedom). So also in this case we have to reject our hypothesis that IPES behaves like a random mapping in this cycle and tail length test. In Appendix B the values of $X^2$ for other tests can be found, for $L=1$ and $L=2$.

### 7. Conclusions

The results in Appendix B show that we mathematically have to reject our hypothesis in many of the test cases. The question remains how important we consider this feature. The fact that in some of the cases we don't have to reject our hypothesis shows that we can't be sure that IPES doesn't behave like a random mapping in the cycle and tail length tests. Besides we have the problem that we have to many observations per test, that means a Pearson's $\chi^2$-test for goodness of fit yields the best results if the number of observations is about "$5 / \text{smallest possible probability}$". This is for example in the case $L=2$, but we have 8128512 observations. Maybe a better possible test for goodness of fit is the discrete Kolmogorov-Smirnov test. It seems interesting to perform the test for $L=3$ and maybe for $L=4$ (therefore a super computer will be necessary).

It is also possible to investigate random inputs together with fixed keys instead of random keys. This could, for example, show the existence of a class of weak keys. Besides all this we have the question what consequence does the rejection of our hypothesis imply for the security of IPES?

### References

Appendix A

The computer program used to perform the test for all sets $F_O$, $F_I$ and permutations (of cardinality $L$) is written in the programming language C and runs on a SUN-system with SunOS release 4.1.1.

The code of the program can be found further on in this appendix. The rest of the appendix is dedicated on an explanation of the structure of the program.

In the included file CRYPT.H the used data types are declared, for example the declarations of UserKeyT, KeyT and DataT can be found in this file. Also $u_{\text{int}16}$ is declared in CRYPT.H as a 16 bits integer.

To create a random input and key the functions SRAND48 and MRAND48 are used. First SRAND48 to reset the random number generator. The function MRAND48 returns an integer number, such that the probability of this number to be positive is $\frac{1}{2}$.

The function ExpandUserKey is declared in CRYPT.H and is necessary for the application of $\text{IPES}$ further on in the program. It extends the 128 bits random key to the complete 832 bits key.

The arrays $F_O$ and $F_I$ are used respectively for the representations of the sets $F_O$ and $F_I$. To pass through all possible sets for $F_O$ and $F_I$ the following strategy is used:
(explanation for $F_O$, $F_I$ analogously)

<table>
<thead>
<tr>
<th>Start FO</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>\ldots</th>
<th>L-2</th>
<th>L-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>next FO</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>\ldots</td>
<td>L-2</td>
<td>L</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FO</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>\ldots</td>
<td>L-2</td>
<td>63</td>
</tr>
<tr>
<td>FO</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>\ldots</td>
<td>L-1</td>
<td>L</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>and the last FO</td>
<td>64-L</td>
<td>65-L</td>
<td>66-L</td>
<td>\ldots</td>
<td>62</td>
<td>63</td>
</tr>
</tbody>
</table>
The array \( p \) is used for the representation of the permutations. To test all the permutations the following strategy was implemented:

<table>
<thead>
<tr>
<th>First permutation ( p )</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>L-3</th>
<th>L-2</th>
<th>L-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>next ( p )</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>L-3</td>
<td>L-1</td>
<td>L-2</td>
</tr>
<tr>
<td>( p )</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>L-2</td>
<td>L-3</td>
<td>L-1</td>
</tr>
<tr>
<td>( p )</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>L-2</td>
<td>L-1</td>
<td>L-3</td>
</tr>
<tr>
<td>( p )</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>L-1</td>
<td>L-3</td>
<td>L-2</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>and the last permutation</td>
<td>L-1</td>
<td>L-2</td>
<td>L-3</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

The variable \( \text{stop} \) is used to detect whether all permutations are tested. If \( p \) is the last permutation then \( \text{stop} \) is set to 1 otherwise \( \text{stop} \) has the value 0.

After the line "/* here testing */" the random chosen input stored in data1 is copied in data2 and data3. In the first loop we don't have to encrypt the input, this is for the modified test (for the original test the program has to be slightly modified). In the other loops we encrypt the (new) input data2. This is done in the function Idea (see CRYPT.H). The cipher text is stored in data3. The bits \( F_0 \) of data3 are read as integer in the variable \( \text{res} \). This is done to determine when a cycle has been completed.

**Warning:** This strategy doesn't work for \( L \geq 32 \), because \( \text{res} \) is declared as a long integer.

Also data2 is adjusted in the positions \( F_I \), using the permutation \( p \). The resulting cycle length and tail length is stored in the 2-dimensional array cycle, where we use the convention that the first position is the tail length and the second the cycle length. The variable \( \text{cycle}[\mu][\lambda] \) represents the number of times we found a tail length \( \mu \) and cycle length \( \lambda \).

Finally the variable \( \text{test} \) shows whether we completed the cycle or not. If the cycle is completed \( \text{test} \) is set to 1, so we can investigate the next permutation. If not, \( \text{test} \) has the value 0.
typedef int int32;
typedef unsigned int u_int32;
typedef unsigned short u_int16;
typedef char u_int8;

#define dataSize 8 /* bytes = 64 bits */
#define dataLen 4
#define keySize 104 /* bytes = 832 bits */
#define keyLen 52
#define userKeySize 16 /* bytes = 128 bits */
#define userKeyLen 8

#define DataT(v) u_int16 v[dataLen]
#define KeyT(v) u_int16 v[keyLen]
#define UserKeyT(v) u_int16 v[userKeyLen]

void Idea( /* DataT(dataIn), DataT(dataout), KeyT(key) */);
void InvertIdeaKey( /* KeyT(key), KeyT(invKey) */ );
void ExpandUserKey( /* UserKeyT(userKey), KeyT(key) */ );
Test program

```c
#include <stdio.h>
#include <crypt.h>
#define L 2
#define TEX 4 /* TEX is 2 to power L */
#define POW65535

UserKeyT(userKey);
KeyT(key);
DataT(data1);
DataT(data2);
DataT(data3);

main(argc, argv)
int argc;
char *argv[];
{
    int i,j,k,p[64],pos1,pos2,pos3,val1,val2,startValue;
    u_int161;
    short stop,ready,test,FO[64],FI[64];
    long res,count,tailar[TEX+1],cycleITEX+1],il;
    FILE *fp;

    if (argc != 2) { printf("Usage: %s initial value\n", argv[0]); } else
    {
        /* initialisation of the key and the plain text */
        sscanf(argv[1], "%d", &startValue);
        srand48(startValue);
        for (i=0; i<8; i++)
            { l=0;
              for (j=0; j<16; j++) l=2*l+(rand48()>0);
              userKey[i]=l; }
        for (i=0; i<4; i++)
            { l=0;
              for (j=0; j<16; j++) l=2*l+(rand48()>0);
              data1[i]=l; }

        /* derive the 832 bits key from the 128 bits representation */
        ExpandUserKey(userKey,key);

        /* initialisation of cycle and tail length */
        for (i=0; i<TEX; i++)
            { for (count=0; count<=TEX; count++) cycle[i][count]=0; }

        /* initialisation of FO */
        for (i=0; i<L; i++) FO[i]=i;
    }
}
```

12
while (FO[L-1] != 64)
{
    /* initialisation of FI */
    for (i=0; i<L; i++) FI[i]=i;

    while (FI[L-1] != 64)
    {
        /* initialisation of permutations */
        stop=0;
        for (i=0; i<L; i++) p[i]=i;

        while (stop==0)
        {
            /* here testing */

            for (i=0; i<4; i++) { data3[i]=data1[i];
            data2[i]=data1[i]; }

            test=0;
            count=0;
            while (test==0)
            {
                /* encrypt data2 result in data3 */

                if (count != 0) { Idea(data2,data3,key); }

                res=0;
                for (i=0; i<L; i++)
                {
                    if (((data3[(FO[i]-FO[i]%16)/16]>>(15-(FO[i]%16)))&1)==1)
                        { res*=2; res+=1;
                            data2[(FI[p[i]]-(FI[p[i]]%16)/16]=1<<(15-(FI[p[i]]%16));
                        }
                    else
                        { res*=2;
                            data2[(FI[p[i]]-(FI[p[i]]%16)/16]&=(POW-(1<<(15-(FI[p[i]]%16))));
                        }
                }

                tailar[count]=res;
                i=count-1;
                while ((tailar[i] != res) && (i>=0)) i--;
                if (i>=0) { cycle[i][count-i]++;
                    test=1; }
                else { count++; }
            }

        }
    }
}
/* determine next permutation */
pos1=L-1;
while ((p[pos1]>p[pos1+1]) && (pos1>0)) pos1--;
if (pos1==0) { stop=1; }
else { val1=p[pos1];
pos2=pos1;
Pos3=pos1;
while (pos2 != L)
{ if ((p[pos2]<val1) && (p[pos2]>p[pos1-1]))
{ 
val1=p[pos2];
pos3=pos2;
};
pos2++; 
};
p[pos3]=p[pos1-1];
p[pos1-1]=val1;
ready=0;
while (ready==0)
{ 
pos2=pos1;
while ((p[pos2]<p[pos2+1]) && (pos2<(L-1))) pos2++;
if (pos2 == (L-1)) { ready=1; }
else { val1=p[pos2];
p[pos2]=p[pos2+1];
p[pos2+1]=val1; }; 
}; /* end of determination next permutation */ 

/* determine next FI */
k=0;
while (((FI[k+1]-FI[k])==1) && (k<(L-1))) k++;
if (k==(L-1)) { FI[L-1]++; 
for (j=0; j<L-1; j++) FI[j]=j; }
else { FI[k]++; 
for (j=0; j<k; j++) FI[j]=j; }; /* end of determination next FI */ 

/* determine next FO */
k=0;
while (((FO[k+1]-FO[k])==1) && (k<(L-1))) k++;
if (k==(L-1)) { FO[L-1]++; 
for (j=0; j<L-1; j++) FO[j]=j; }
else { FO[k]++; 
for (j=0; j<k; j++) FO[j]=j; }; /* end of determination next FO */
fp=fopen("results","w");
fprintf(fp,"The initial value is: %d\n", startvalue);
fprintf(fp,"The used plain text is: \n");
for (i=0; i<4; i++) fprintf(fp, " %7d", data1[i]);
fprintf(fp,"The used key is: \n");
for (i=0; i<8; i++) fprintf(fp, " %7d", key[i]);
fprintf(fp,"\n");
fprintf(fp, "cycle length tail length \n");
for (il=1; il<=TEX; il++)
  { for (count=0; count<=(TEX-il); count++)
     fprintf(fp, "%9d %12d %10d\n", il,count,cycle[count][il]); }
fclose(fp);

};
Appendix B

In this appendix the results of the tests for \( L = 1 \) and \( L = 2 \) are given. The results per test consist of the initial value used to produce the random input and key, see the program in Appendix A. Next we find the random input and key, given respectively as 4 and 8 16-bits integers. Finally the number of cycle length and tail length combinations occurred in the test is given.

Each test contains:
- all possible choices for \( F_I \) (with cardinality \( L \)),
- all possible choices for \( F_O \) (with cardinality \( L \)),
- all permutations of \( L \) symbols.

A simple calculation shows that each test consists of \( \left(\frac{64}{L}\right)^2 L! \) loops. For \( L = 1 \) we have 4096 loops, for \( L = 2 \) 8128512. This number of loops is exactly the number \( n \) used in Section 6 in the Pearson's \( \chi^2 \)-test for goodness of fit.

After each test result the value of \( \chi^2 \) is given for that test. For the tests with \( L = 1 \) the first value is for the original test (only the cycle length is interesting, the tail lengths are always zero), the second for the modified test, denoted respectively by \( \chi^2_o \) and \( \chi^2_m \). In the cases with \( L = 2 \) we give 3 values because then the original test gets an \( X^2 \) value for both the cycle length and the tail length.
$L=1$:

The initial value is: 0

The used plain text is:
\[ 4120 \ 40980 \ 39127 \ 42075 \]

The used key is:
\[ 41270 \ 7453 \ 28474 \ 20618 \ 60946 \ 2466 \ 26460 \ 46393 \]

cycle length tail length
\[
\begin{array}{ccc}
1 & 0 & 2205 \\
1 & 1 & 924 \\
2 & 0 & 967 \\
\end{array}
\]

$\chi^2_o = 4.23 \quad \chi^2_m = 24.97$

The initial value is: 1

The used plain text is:
\[ 28796 \ 48444 \ 45482 \ 45865 \]

The used key is:
\[ 54893 \ 22089 \ 27312 \ 37584 \ 57610 \ 43125 \ 6177 \ 58028 \]

cycle length tail length
\[
\begin{array}{ccc}
1 & 0 & 2233 \\
1 & 1 & 853 \\
2 & 0 & 1010 \\
\end{array}
\]

$\chi^2_o = 0.26 \quad \chi^2_m = 45.46$

The initial value is: 2

The used plain text is:
\[ 20035 \ 30723 \ 58969 \ 33461 \]

The used key is:
\[ 17440 \ 38547 \ 45034 \ 3875 \ 47787 \ 24254 \ 53646 \ 34049 \]

cycle length tail length
\[
\begin{array}{ccc}
1 & 0 & 2050 \\
1 & 1 & 1047 \\
2 & 0 & 999 \\
\end{array}
\]

$\chi^2_o = 0.81 \quad \chi^2_m = 1.129$
The initial value is: 3

The used plain text is:
8001  24747  35708  50414

The used key is:
14130  54358  43690  3842  62758  31048  36551  10766

cycle length  tail length
1    0          2190
1    1          907
2    0          999

\[ \chi_o^2 = 0.81 \quad \chi_m^2 = 23.82 \]

The initial value is: 4

The used plain text is:
29670  9723  4267  63448

The used key is:
8445  31682  43972  62321  59135  55179  34806  3285  7523

cycle length  tail length
1    0          1984
1    1          1054
2    0          1058

\[ \chi_o^2 = 1.51 \quad \chi_m^2 = 4.008 \]

The initial value is: 5

The used plain text is:
41920  64215  27867  59027

The used key is:
21398  14741  11422  28962  43359  34806  3285  7523

cycle length  tail length
1    0          2031
1    1          1007
2    0          1058

\[ \chi_o^2 = 1.51 \quad \chi_m^2 = 1.552 \]
The initial value is: 6

The used plain text is:
36549 56175 11126 36873

The used key is:
34253 64357 24863 43025 63430 8244 21116 750

cycle length  tail length
1       0       1886
1       1       1211
2       0       999

\[ \chi_o^2 = 0.81 \quad \chi_m^2 = 47.57 \]

The initial value is: 7

The used plain text is:
65250 34659 37381 33055

The used key is:
48835 61626 58453 44154 15463 38603 40225 58819

cycle length  tail length
1       0       1758
1       1       1280
2       0       1058

\[ \chi_o^2 = 1.51 \quad \chi_m^2 = 106.2 \]

The initial value is: 8

The used plain text is:
42820 5723 56800 14180

The used key is:
59664 8942 61711 31785 1022 61773 49230 38727

cycle length  tail length
1       0       1867
1       1       1209
2       0       1020

\[ \chi_o^2 = 0.02 \quad \chi_m^2 = 49.44 \]
The initial value is: 9

The used plain text is:
33091 23537 41171 21630

The used key is:
55903 17788 61551 61978 16479 16798 47927 37098

cycle length  tail length
1       0        2121
1       1        979
2       0        996

\( \chi^2_o = 1.02 \) \( \chi^2_M = 5.345 \)

The initial value is: 10

The used plain text is:
63855 36588 41774 16825

The used key is:
10276 36793 13669 875 8151 26211 25754 32739

cycle length  tail length
1       0        1878
1       1        1210
2       0        1008

\( \chi^2_o = 0.33 \) \( \chi^2_M = 48.15 \)

The initial value is: 11

The used plain text is:
44104 34508 24173 29603

The used key is:
12076 3431 15039 36616 5198 45217 10709 18502

cycle length  tail length
1       0        1811
1       1        1259
2       0        1026

\( \chi^2_o = 0.005 \) \( \chi^2_M = 81.36 \)
The initial value is: 12

The used plain text is:
50382 53556 2009 9433

The used key is:
23779 20018 16379 57114 19435 5214 30380 2760

cycle length  tail length
1  0  1995
1  1  1067
2  0  1034

\chi_o^2 = 0.13 \quad \chi_M^2 = 3.275

The initial value is: 13

The used plain text is:
62637 23608 10430 5582

The used key is:
35760 33952 47745 20841 370 12189 43201 32229

cycle length  tail length
1  0  1866
1  1  1154
2  0  1076

\chi_o^2 = 3.52 \quad \chi_M^2 = 35.32

The initial value is: 14

The used plain text is:
43432 11520 32007 788

The used key is:
47611 33396 48011 32786 22226 51563 62910 53800

cycle length  tail length
1  0  2091
1  1  984
2  0  1021

\chi_o^2 = 0.01 \quad \chi_M^2 = 2.474
L=2:

The initial value is: 0

The used plain text is:
4120 40980 39127 42075

The used key is:
41270 7453 28474 20618 60946 2466 26460 46393

cycle length  tail length
1  0  2224472
1  1  1487587
1  2  717344
1  3  170082
2  0  1572979
2  1  702710
2  2  168310
3  0  745606
3  1  162040
4  0  177382

$\chi^2_{o}(\text{cycle}) = 5277.8 \quad \chi^2_{o}(\text{tail}) = 7982.1 \quad \chi^2_{M} = 38184.0$

The initial value is: 1

The used plain text is:
28796 48444 45482 45865

The used key is:
54893 22089 27312 37584 57610 43125 6177 58028

cycle length  tail length
1  0  2210627
1  1  1447402
1  2  692021
1  3  164547
2  0  1619391
2  1  695310
2  2  167569
3  0  777095
3  1  166164
4  0  188386

$\chi^2_{o}(\text{cycle}) = 134.8 \quad \chi^2_{o}(\text{tail}) = 14637.7 \quad \chi^2_{M} = 47510.85$
The initial value is: 2

The used plain text is:
20035 30723 58969 33461

The used key is:
17440 38547 45034 3875 47787 24254 53646 34049

cycle length tail length
1 0 2008215
1 1 1561151
1 2 783878
1 3 193804
2 0 1493989
2 1 774810
2 2 193400
3 0 743593
3 1 189147
4 0 186525

\( \chi^2_{(\text{cycle})} = 905.0 \quad \chi^2_{(\text{tail})} = 810.6 \quad \chi^2_{M} = 3256.94 \)

The initial value is: 3

The used plain text is:
8001 24747 35708 50414

The used key is:
14130 54358 43690 3842 62758 31048 36551 10766

cycle length tail length
1 0 2160582
1 1 1487527
1 2 710092
1 3 170181
2 0 1588086
2 1 716849
2 2 174263
3 0 763421
3 1 169399
4 0 188112

\( \chi^2_{(\text{cycle})} = 936.2 \quad \chi^2_{(\text{tail})} = 8175.4 \quad \chi^2_{M} = 23835.14 \)
### Appendix C

Table of the \( \chi^2 \)-distribution.

<table>
<thead>
<tr>
<th>Degrees of freedom</th>
<th>F(x)</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.02</td>
<td>0.09</td>
<td>0.21</td>
<td>0.38</td>
<td>0.60</td>
<td>0.86</td>
<td>1.15</td>
<td>1.48</td>
</tr>
<tr>
<td>0.005</td>
<td>0.00</td>
<td>0.01</td>
<td>0.07</td>
<td>0.21</td>
<td>0.41</td>
<td>0.68</td>
<td>0.99</td>
<td>1.34</td>
<td>1.73</td>
<td>2.16</td>
<td></td>
</tr>
<tr>
<td>0.01</td>
<td>0.00</td>
<td>0.02</td>
<td>0.11</td>
<td>0.30</td>
<td>0.55</td>
<td>0.87</td>
<td>1.24</td>
<td>1.65</td>
<td>2.09</td>
<td>2.56</td>
<td></td>
</tr>
<tr>
<td>0.025</td>
<td>0.00</td>
<td>0.05</td>
<td>0.22</td>
<td>0.48</td>
<td>0.83</td>
<td>1.24</td>
<td>1.69</td>
<td>2.18</td>
<td>2.70</td>
<td>3.25</td>
<td></td>
</tr>
<tr>
<td>0.05</td>
<td>0.00</td>
<td>0.10</td>
<td>0.35</td>
<td>0.71</td>
<td>1.15</td>
<td>1.64</td>
<td>2.17</td>
<td>2.73</td>
<td>3.33</td>
<td>3.94</td>
<td></td>
</tr>
<tr>
<td>0.1</td>
<td>0.02</td>
<td>0.21</td>
<td>0.58</td>
<td>1.06</td>
<td>1.61</td>
<td>2.20</td>
<td>2.83</td>
<td>3.49</td>
<td>4.17</td>
<td>4.87</td>
<td></td>
</tr>
<tr>
<td>0.25</td>
<td>0.10</td>
<td>0.58</td>
<td>1.21</td>
<td>1.92</td>
<td>2.67</td>
<td>3.45</td>
<td>4.25</td>
<td>5.07</td>
<td>5.90</td>
<td>6.74</td>
<td></td>
</tr>
<tr>
<td>0.5</td>
<td>0.45</td>
<td>1.39</td>
<td>2.37</td>
<td>3.36</td>
<td>4.35</td>
<td>5.35</td>
<td>6.35</td>
<td>7.34</td>
<td>8.34</td>
<td>9.34</td>
<td></td>
</tr>
<tr>
<td>0.75</td>
<td>1.32</td>
<td>2.77</td>
<td>4.11</td>
<td>5.39</td>
<td>6.63</td>
<td>7.84</td>
<td>9.04</td>
<td>10.22</td>
<td>11.39</td>
<td>12.55</td>
<td></td>
</tr>
<tr>
<td>0.9</td>
<td>2.71</td>
<td>4.61</td>
<td>6.25</td>
<td>7.78</td>
<td>9.24</td>
<td>10.64</td>
<td>12.02</td>
<td>13.36</td>
<td>14.68</td>
<td>15.99</td>
<td></td>
</tr>
<tr>
<td>0.95</td>
<td>3.84</td>
<td>5.99</td>
<td>7.81</td>
<td>9.49</td>
<td>11.07</td>
<td>12.59</td>
<td>14.07</td>
<td>15.51</td>
<td>16.92</td>
<td>18.31</td>
<td></td>
</tr>
<tr>
<td>0.975</td>
<td>5.02</td>
<td>7.38</td>
<td>9.35</td>
<td>11.14</td>
<td>12.83</td>
<td>14.45</td>
<td>16.01</td>
<td>17.53</td>
<td>19.02</td>
<td>20.48</td>
<td></td>
</tr>
<tr>
<td>0.99</td>
<td>6.63</td>
<td>9.21</td>
<td>11.34</td>
<td>13.28</td>
<td>15.09</td>
<td>16.81</td>
<td>18.48</td>
<td>20.09</td>
<td>21.67</td>
<td>23.21</td>
<td></td>
</tr>
<tr>
<td>0.995</td>
<td>7.88</td>
<td>10.60</td>
<td>12.84</td>
<td>14.86</td>
<td>16.75</td>
<td>18.55</td>
<td>20.28</td>
<td>21.96</td>
<td>23.59</td>
<td>25.19</td>
<td></td>
</tr>
<tr>
<td>0.999</td>
<td>10.83</td>
<td>13.82</td>
<td>16.27</td>
<td>18.47</td>
<td>20.52</td>
<td>22.46</td>
<td>24.32</td>
<td>26.13</td>
<td>27.88</td>
<td>29.59</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Degrees of freedom</th>
<th>F(x)</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
<th>19</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.001</td>
<td>1.83</td>
<td>2.21</td>
<td>2.62</td>
<td>3.04</td>
<td>3.48</td>
<td>3.94</td>
<td>4.42</td>
<td>4.90</td>
<td>5.41</td>
<td>5.92</td>
<td></td>
</tr>
<tr>
<td>0.005</td>
<td>2.60</td>
<td>3.07</td>
<td>3.57</td>
<td>4.07</td>
<td>4.60</td>
<td>5.14</td>
<td>5.70</td>
<td>6.26</td>
<td>6.84</td>
<td>7.43</td>
<td></td>
</tr>
<tr>
<td>0.01</td>
<td>3.05</td>
<td>3.57</td>
<td>4.11</td>
<td>4.66</td>
<td>5.23</td>
<td>5.81</td>
<td>6.41</td>
<td>7.01</td>
<td>7.63</td>
<td>8.26</td>
<td></td>
</tr>
<tr>
<td>0.025</td>
<td>3.82</td>
<td>4.40</td>
<td>5.01</td>
<td>5.63</td>
<td>6.26</td>
<td>6.91</td>
<td>7.56</td>
<td>8.23</td>
<td>8.91</td>
<td>9.59</td>
<td></td>
</tr>
<tr>
<td>0.05</td>
<td>4.57</td>
<td>5.23</td>
<td>5.89</td>
<td>6.57</td>
<td>7.26</td>
<td>7.96</td>
<td>8.67</td>
<td>9.39</td>
<td>10.12</td>
<td>10.85</td>
<td></td>
</tr>
<tr>
<td>0.1</td>
<td>5.58</td>
<td>6.30</td>
<td>7.04</td>
<td>7.79</td>
<td>8.55</td>
<td>9.31</td>
<td>10.09</td>
<td>10.86</td>
<td>11.65</td>
<td>12.44</td>
<td></td>
</tr>
<tr>
<td>0.25</td>
<td>7.58</td>
<td>8.44</td>
<td>9.30</td>
<td>10.17</td>
<td>11.04</td>
<td>11.91</td>
<td>12.79</td>
<td>13.68</td>
<td>14.56</td>
<td>15.45</td>
<td></td>
</tr>
<tr>
<td>0.5</td>
<td>10.34</td>
<td>11.34</td>
<td>12.34</td>
<td>13.34</td>
<td>14.34</td>
<td>15.34</td>
<td>16.34</td>
<td>17.34</td>
<td>18.34</td>
<td>19.34</td>
<td></td>
</tr>
<tr>
<td>0.75</td>
<td>13.70</td>
<td>14.85</td>
<td>15.98</td>
<td>17.12</td>
<td>18.25</td>
<td>19.37</td>
<td>20.49</td>
<td>21.60</td>
<td>22.72</td>
<td>23.83</td>
<td></td>
</tr>
<tr>
<td>0.9</td>
<td>17.28</td>
<td>18.55</td>
<td>19.81</td>
<td>21.06</td>
<td>22.31</td>
<td>23.54</td>
<td>24.77</td>
<td>25.99</td>
<td>27.20</td>
<td>28.41</td>
<td></td>
</tr>
<tr>
<td>0.95</td>
<td>19.68</td>
<td>21.03</td>
<td>22.36</td>
<td>23.68</td>
<td>25.00</td>
<td>26.30</td>
<td>27.59</td>
<td>28.87</td>
<td>30.14</td>
<td>31.41</td>
<td></td>
</tr>
<tr>
<td>0.975</td>
<td>21.92</td>
<td>23.34</td>
<td>24.74</td>
<td>26.12</td>
<td>27.49</td>
<td>28.85</td>
<td>30.19</td>
<td>31.53</td>
<td>32.85</td>
<td>34.17</td>
<td></td>
</tr>
<tr>
<td>0.99</td>
<td>24.73</td>
<td>26.22</td>
<td>27.69</td>
<td>29.14</td>
<td>30.58</td>
<td>32.00</td>
<td>33.41</td>
<td>34.81</td>
<td>36.19</td>
<td>37.57</td>
<td></td>
</tr>
<tr>
<td>0.995</td>
<td>26.76</td>
<td>28.30</td>
<td>29.82</td>
<td>31.32</td>
<td>32.80</td>
<td>34.27</td>
<td>35.72</td>
<td>37.17</td>
<td>38.58</td>
<td>40.00</td>
<td></td>
</tr>
<tr>
<td>0.999</td>
<td>31.26</td>
<td>32.91</td>
<td>34.53</td>
<td>36.12</td>
<td>37.70</td>
<td>39.25</td>
<td>40.79</td>
<td>42.31</td>
<td>43.82</td>
<td>45.32</td>
<td></td>
</tr>
</tbody>
</table>
### Degrees of freedom

<table>
<thead>
<tr>
<th>F(x)</th>
<th>21</th>
<th>22</th>
<th>23</th>
<th>24</th>
<th>25</th>
<th>26</th>
<th>27</th>
<th>28</th>
<th>29</th>
<th>30</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.001</td>
<td>6.4</td>
<td>7.0</td>
<td>7.5</td>
<td>8.1</td>
<td>8.7</td>
<td>9.2</td>
<td>9.8</td>
<td>10.4</td>
<td>11.0</td>
<td>11.6</td>
</tr>
<tr>
<td>0.005</td>
<td>8.0</td>
<td>8.6</td>
<td>9.3</td>
<td>9.9</td>
<td>10.5</td>
<td>11.2</td>
<td>11.8</td>
<td>12.5</td>
<td>13.1</td>
<td>13.8</td>
</tr>
<tr>
<td>0.01</td>
<td>8.9</td>
<td>9.5</td>
<td>10.2</td>
<td>10.9</td>
<td>11.5</td>
<td>12.2</td>
<td>12.9</td>
<td>13.6</td>
<td>14.3</td>
<td>15.0</td>
</tr>
<tr>
<td>0.025</td>
<td>10.3</td>
<td>11.0</td>
<td>11.7</td>
<td>12.4</td>
<td>13.1</td>
<td>13.8</td>
<td>14.6</td>
<td>15.3</td>
<td>16.0</td>
<td>16.8</td>
</tr>
<tr>
<td>0.05</td>
<td>11.6</td>
<td>12.3</td>
<td>13.1</td>
<td>13.8</td>
<td>14.6</td>
<td>15.4</td>
<td>16.2</td>
<td>16.9</td>
<td>17.7</td>
<td>18.5</td>
</tr>
<tr>
<td>0.1</td>
<td>13.2</td>
<td>14.0</td>
<td>14.8</td>
<td>15.7</td>
<td>16.5</td>
<td>17.3</td>
<td>18.1</td>
<td>18.9</td>
<td>19.8</td>
<td>20.6</td>
</tr>
<tr>
<td>0.25</td>
<td>16.3</td>
<td>17.2</td>
<td>18.1</td>
<td>19.0</td>
<td>19.9</td>
<td>20.8</td>
<td>21.7</td>
<td>22.7</td>
<td>23.6</td>
<td>24.5</td>
</tr>
<tr>
<td>0.5</td>
<td>20.3</td>
<td>21.3</td>
<td>22.3</td>
<td>23.3</td>
<td>24.3</td>
<td>25.3</td>
<td>26.3</td>
<td>27.3</td>
<td>28.3</td>
<td>29.3</td>
</tr>
<tr>
<td>0.75</td>
<td>24.9</td>
<td>26.0</td>
<td>27.1</td>
<td>28.2</td>
<td>29.3</td>
<td>30.4</td>
<td>31.5</td>
<td>32.6</td>
<td>33.7</td>
<td>34.8</td>
</tr>
<tr>
<td>0.9</td>
<td>29.6</td>
<td>30.8</td>
<td>32.0</td>
<td>33.2</td>
<td>34.4</td>
<td>35.6</td>
<td>36.7</td>
<td>37.9</td>
<td>39.1</td>
<td>40.3</td>
</tr>
<tr>
<td>0.95</td>
<td>32.7</td>
<td>33.9</td>
<td>35.2</td>
<td>36.4</td>
<td>37.7</td>
<td>38.9</td>
<td>40.1</td>
<td>41.3</td>
<td>42.6</td>
<td>43.8</td>
</tr>
<tr>
<td>0.975</td>
<td>35.5</td>
<td>36.8</td>
<td>38.1</td>
<td>39.4</td>
<td>40.6</td>
<td>41.9</td>
<td>43.2</td>
<td>44.5</td>
<td>45.7</td>
<td>47.0</td>
</tr>
<tr>
<td>0.99</td>
<td>38.9</td>
<td>40.3</td>
<td>41.6</td>
<td>43.0</td>
<td>44.3</td>
<td>45.6</td>
<td>47.0</td>
<td>48.3</td>
<td>49.6</td>
<td>50.9</td>
</tr>
<tr>
<td>0.995</td>
<td>41.4</td>
<td>42.8</td>
<td>44.2</td>
<td>45.6</td>
<td>46.9</td>
<td>48.3</td>
<td>49.6</td>
<td>51.0</td>
<td>52.3</td>
<td>53.7</td>
</tr>
<tr>
<td>0.999</td>
<td>46.8</td>
<td>48.3</td>
<td>49.7</td>
<td>51.2</td>
<td>52.6</td>
<td>54.1</td>
<td>55.5</td>
<td>56.9</td>
<td>58.3</td>
<td>59.7</td>
</tr>
</tbody>
</table>

### Degrees of freedom

<table>
<thead>
<tr>
<th>F(x)</th>
<th>40</th>
<th>50</th>
<th>60</th>
<th>70</th>
<th>80</th>
<th>90</th>
<th>100</th>
<th>&gt;100 (approximation)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.001</td>
<td>17.9</td>
<td>24.7</td>
<td>31.7</td>
<td>39.0</td>
<td>46.5</td>
<td>54.2</td>
<td>61.9</td>
<td>(h-3.09)²/2</td>
</tr>
<tr>
<td>0.005</td>
<td>20.7</td>
<td>28.0</td>
<td>35.5</td>
<td>43.3</td>
<td>51.2</td>
<td>59.2</td>
<td>67.3</td>
<td>(h-2.58)²/2</td>
</tr>
<tr>
<td>0.01</td>
<td>22.2</td>
<td>29.7</td>
<td>37.5</td>
<td>45.4</td>
<td>53.5</td>
<td>61.8</td>
<td>70.1</td>
<td>(h-2.33)²/2</td>
</tr>
<tr>
<td>0.025</td>
<td>24.4</td>
<td>32.4</td>
<td>40.5</td>
<td>48.8</td>
<td>57.2</td>
<td>65.6</td>
<td>74.2</td>
<td>(h-1.96)²/2</td>
</tr>
<tr>
<td>0.05</td>
<td>26.5</td>
<td>34.8</td>
<td>43.2</td>
<td>51.7</td>
<td>60.4</td>
<td>69.1</td>
<td>77.9</td>
<td>(h-1.64)²/2</td>
</tr>
<tr>
<td>0.1</td>
<td>29.1</td>
<td>37.7</td>
<td>46.5</td>
<td>55.3</td>
<td>64.3</td>
<td>73.3</td>
<td>82.4</td>
<td>(h-1.28)²/2</td>
</tr>
<tr>
<td>0.25</td>
<td>33.7</td>
<td>42.9</td>
<td>52.3</td>
<td>61.7</td>
<td>71.1</td>
<td>80.6</td>
<td>90.1</td>
<td>(h-0.67)²/2</td>
</tr>
<tr>
<td>0.5</td>
<td>39.3</td>
<td>49.3</td>
<td>59.3</td>
<td>69.3</td>
<td>79.3</td>
<td>89.3</td>
<td>99.3</td>
<td>h²/2</td>
</tr>
<tr>
<td>0.75</td>
<td>45.6</td>
<td>56.3</td>
<td>67.0</td>
<td>77.6</td>
<td>88.1</td>
<td>98.6</td>
<td>109.1</td>
<td>(h+0.67)²/2</td>
</tr>
<tr>
<td>0.9</td>
<td>51.8</td>
<td>63.2</td>
<td>74.4</td>
<td>85.5</td>
<td>96.6</td>
<td>107.6</td>
<td>118.5</td>
<td>(h+1.28)²/2</td>
</tr>
<tr>
<td>0.95</td>
<td>55.8</td>
<td>67.5</td>
<td>79.1</td>
<td>90.5</td>
<td>101.9</td>
<td>113.1</td>
<td>124.3</td>
<td>(h+1.64)²/2</td>
</tr>
<tr>
<td>0.975</td>
<td>59.3</td>
<td>71.4</td>
<td>83.3</td>
<td>95.0</td>
<td>106.6</td>
<td>118.1</td>
<td>129.6</td>
<td>(h+1.96)²/2</td>
</tr>
<tr>
<td>0.99</td>
<td>63.7</td>
<td>76.2</td>
<td>88.4</td>
<td>100.4</td>
<td>112.3</td>
<td>124.1</td>
<td>135.8</td>
<td>(h+2.33)²/2</td>
</tr>
<tr>
<td>0.995</td>
<td>66.8</td>
<td>79.5</td>
<td>92.0</td>
<td>104.2</td>
<td>116.3</td>
<td>128.3</td>
<td>140.2</td>
<td>(h+2.58)²/2</td>
</tr>
<tr>
<td>0.999</td>
<td>73.4</td>
<td>86.7</td>
<td>99.6</td>
<td>112.3</td>
<td>124.8</td>
<td>137.2</td>
<td>149.4</td>
<td>(h+3.09)²/2</td>
</tr>
</tbody>
</table>

In the last column is \( h = \sqrt{2m - 1} \) wherein \( m \) the number of degrees of freedom is.