The Triple-DES-96 Cryptographic System

V. M. Silva-García

Centro de Innovación y Desarrollo Tecnológico en Cómputo
Instituto Politécnico Nacional, México
vsilvag@ipn.mx

R. Flores-Carapia

Centro de Innovación y Desarrollo Tecnológico en Cómputo
Instituto Politécnico Nacional, México
rfloresca@ipn.mx

C. Rentería-Márquez

Escuela Superior de Física y Matemáticas
Instituto Politécnico Nacional, México
crenteriam@gmail.com

B. Luna-Benoso

Escuela Superior de Cómputo
Instituto Politécnico Nacional, México
blunabenoso@yahoo.com

Copyright © 2013 V. M. Silva-García et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Abstract

This paper proposes a variant of the Triple-DES cryptosystem; the latter, has a 64-bits block input according to the international standard, FIPS PUB 146-3. However, the Triple-DES cryptosystem input blocks can be extended from 64 to 96 bits without losing computational complexity, especially before a brute force attack. This change incorporates a modification to Triple-DES cryptosystems that appears in more recent systems as Advanced Encryption Standard (AES), FIPS PUB
As a permutation is removed from the Triple-DES system and the length of the input block is increased, it has as a result Triple-DES-96 encryption consuming time in less than half the time used by Triple DES, that is, if a M file is encrypted (with \(m \approx 61.44\) Mb or bigger) using the Triple-DES system and runs in time \(t\), with the Triple-DES-96 cryptosystem it will run in less time than \((1/2)t\). Developing this new cryptosystem intends to apply the factorial theorem, i.e. any permutation on an array of 96 positions can be built from 3 permutations on arrangements of 64 positions. According to JV Theorem, given a number \(n\) with \(0 \leq n \leq 64! - 1 \approx 10^{89}\), it can associate a permutation with an arrangement of 64 positions in 63 steps. Therefore we can apply a variable permutation on a chain of 96 bits in the Triple-DES-96 cryptosystem at the start of the first cycle third round and its inverse at the entrance of the third cycle fifteenth round, using \(3n_i\) numbers with \(0 \leq n \leq 10^{89}\) for \(i = 1, 2, 3\). This strengthens Triple-DES-96 with respect to Triple-DES.

Mathematics Subject Classification: 14G50

Keywords: JV Theorem, Triple-DES, Factorial Theorem, Variable permutation

1 Introduction

According to the international standard encryption process Triple-DES starts with a fixed 64 bit block permutation [6]. However, this paper applies a variable permutation at the start of the first cycle third round and its inverse at the beginning of the third cycle fifteenth round of the encryption process. In this paper, variable permutation means that it is unknown to a possible opponent since it can be changed in each file for a ciphering process, and not as Triple-DES, which is fixed and known. Two important advantages of the proposed cryptosystem with regard to the Triple-DES cryptosystem are: first, the Triple-DES-96 cryptosystem ciphers 96-bit blocks and the second, the cryptosystem proposes the elimination of the Triple-DES cryptosystem P permutation [6]. If this is taken into account these results give a faster encryption when using Triple-DES-96 than when using Triple-DES, in fact more than double. Also, the complexity is benefitted because if we desire decrypting a plaintext with a different permutation, maintaining fixed 64-bit keys, two or three as is the case [2], we do not get the original plaintext. This research does not present a formal proof of increased computational complexity.

In another vein, the variable permutation creation will trade through the algorithms used in JV and Factorial Theorems [14, 15]. Specifically, it is performed from \(n_i\) positive integers with \(0 \leq n_i \leq 10^{89}\) for \(i = 1, 2, 3\) which are
chosen randomly. This reduces the number of bit operations [10], as it is better
to apply three permutations using 3 numbers $n_i$ with the characteristics men-
tioned above, instead of $0 \leq n \leq 96! - 1 \approx 10^{150}$ numbers to get permutations
on 96 arrangement positions directly [15].

2 The Triple-DES-96 algorithm

This section begins by answering the following question: how to define the
Triple-DES-96 algorithm? In this case it is sufficient to give a cycle description
showing the DES-96 algorithm. Let’s say that the relationship between DES-96
and Triple-DES-96 is similar to DES and Triple-DES. Therefore, we present the
DES-96 encryption cycle as follows: $E_{96,K}(x)$, where $x$ is the 96-bit plaintext
and $K$ is the key. The following description will be made of round $i$, where
$i = 1, 2 \ldots, 16$. The DES-96 function $f_{96}(R_{i-1}, k_i)$, is different from the DES
function $f(R_{i-1}, k_i)$. The $f_{96}(R_{i-1}, k_i)$ function performs the following steps:

1.- The input string to the $i$-th round with 96 bits long is divided into 2,
$L_{i-1}$ and $R_{i-1}$ where each is 48 bits long. The operation runs xor $R_{i-1} \oplus k_i$
= Input. The $k_i$ chain is the $i$-th keys program which also has a length of 48
bits.

2. - The 48-bit Input chain is the entrance to the boxes (8 in total) in
which the rule of DES substitution is followed. The output of the boxes is a
32-bit called $C$.

3. - The output of the boxes is a string of 32 bits, that is $C$, and if P
permutation is applied to $C$ the result is a 32 bits long chain [6]. However, it
needs a 48 bits chain for the xor operation with the previous left chain. For
this reason, in this research we propose to employ the permutation $E$ used in
the standard [6]. In summary, this paper eliminates the permutation P and
instead uses the permutation $E$, i.e, $E(C)$. Then, we define the 48-bit chain
$f_{96}(R_{i-1}, k_i) = E(C)$.

Calculating the left and right blocks $L_i$ and $R_i$ is performed as follows:
the left block $L_i = R_{i-1}$, and the right block $R_i$ is obtained as $R_i = L_{i-1} \oplus
f_{96}(R_{i-1}, k_i)$.

Having defined a round, we are ready to describe in general the structure
of a DES-96 algorithm. The DES-96 algorithm is developed in 16 rounds and
starts with $L_0$, $R_0$ blocks which are 48 bits each one. All the input blocks in
each round, $R_i$ and $L_i$ with $i = 0, 1 \ldots, 16$ are 48-bit strings. Subsequently the
$f_{96}(R_{i-1}, k_i)$ function is applied. The $f_{96}(R_{i-1}, k_i)$ function result is applied
an xor operation with the previous left block $L_{i-1}$, to obtain the right block
$R_i$. To terminate a standard round, the left block $L_i = R_{i-1}$, and the last round
will have $L_{16}$, $R_{16}$ which is the cipher text $y = E_{96,K}(x)$. The keys program
is generated in the same way as DES. It begins with a chain of 64 or 56 bits to
obtain 16 strings of 48 bits each, following the same steps as the DES algorithm [6].

The Triple-DES-96 algorithm is made from 3 cycles. The first is a DES-96 with the inclusion of the $PV$ permutation at the input of the third round, i.e., it is used in the $L_2, R_2$ chain. The second cycle is a DES-96 decryption algorithm and the third is a DES-96 encryption algorithm. The last cycle includes the $PV^{-1}$ permutation at the entrance to the fifteenth round, i.e., it is applied to the $L_{14}, R_{14}$ chain. In the encryption process 2 or 3 keys can be used [2]. This paper will use two keys $K^1, K^2$ with $K^1 \neq K^2$, specifically, the first cycle is encrypted using the $K^1$ key and the second is decrypted with the $K^2$ key, so that the third cycle is encrypted again with the $K^1$ key.

Regarding the difference in encryption time, roughly, between Triple-DES-96 and Triple-DES is based on the following:

1. - For modern computers running on 64-bit words, there is virtually no difference in time when it is working with 48-bit blocks or with 32 bit blocks.
2. - The DES-96 cryptosystem eliminates the P permutation in each round, and instead of P permutation the E permutation is applied at the boxes exit.
3. - The Triple-DES-96 cryptosystem ciphers 96-bit blocks.

Based on the three points above, it is assumed that the Triple-DES-96 encryption time is less than Triple-DES. The results section will show that the proposed cryptosystem of this research deciphers in less than half the time of Triple-DES. It was also mentioned that Triple-DES-96 included an amendment that is similar to recent cryptosystems such as AES [8]. The latter, because the proposed cryptosystem starts with xor operation, $R_0 \oplus k_1$. The difference with AES is because this performs an xor operation with 128 bit strings and Triple-DES-96 does it with 48-bit strings. Moreover, this feature of Triple-DES-96 determines the round in which $PV$ and $PV^{-1}$ are applied.

Why do we apply the $PV$ permutation at the entrance to the third round of the first encryption cycle and the reverse $PV^{-1}$, at the entrance to the fifteenth round of the third cycle?

It is desirable that before applying the Triple-DES-96 $PV$ permutation, an xor operation using program keys with $L_0$, $R_0$ is needed, as in the recent algorithm family of the "Substitution-Permutation-Network" (SPN) [4], AES is in this family. Taking this idea into account, it follows that both plaintext $L_0$, $R_0$ blocks are applied to the xor operation after the second round. Recall that in the first round $R_0 \oplus k_1$ is carried out, thus, the xor operation between $L_0$ block and $k_1$ is performed through $L_0 \oplus f_{96}(R_0, k_1)$. However, the xor is not applied to $R_0$ right block because $L_1 = R_0$. In the second round an xor operation is used between $R_0$ and $k_2$ when $L_1 \oplus f_{96}(R_1, k_2)$ is done. Following the same reasoning, after applying the $PV^{-1}$ permutation continuing with xor operations between left, right blocks and program keys before reaching the cipher text is recommended. Therefore, in the third cycle the $PV^{-1}$ permu-
The triple-DES-96 cryptographic system

929

ination is applied to $L_{14}$, $R_{14}$ block, which is the input of the fifteenth round. In fact, the $PV$ and $PV^{-1}$ permutations are protected by the xor operation using $k_1$, $k_2$ first cycle keys and $k_{15}$, $k_{16}$ third cycle keys.

3 Proof that the Triple-DES-96 algorithm defines a one to one function

It is important that encryption algorithms do not have collisions, in other words, the encryption algorithm defines a one to one or Bijective function. So, it will be necessary to show that Triple-DES-96 defines a one to one function. It will be sufficient to show that first, second and third cycles are each a one to one function. This follows from the fact that if $f$, $g$ are two one to one functions, it implies that $f \circ g$ –composite function– is also a one to one function. In this sense, the following theorem is presented.

Theorem 3.1. The Triple-DES-96 algorithm described above defines a one to one function.

Proof. As noted earlier, each cycle has 16 rounds so it follows that each cycle can be divided into 16 functions. Then, if we demonstrate that these 16 functions are Bijective in each cycle it can be concluded that the Triple-DES-96 algorithm defines a one to one function. The demonstration that each of the functions in each cycle is Bijective can be made by using reductio ad absurdum. In order to demonstrate the theorem, the rounds can be classified into 3 types, namely:

i) A standard round can be defined as one that does not apply any permutations such as $PV$ or $PV^{-1}$ during the cipher process, i.e. it is a round as described above. Suppose the $i$-th round is denoted as $g(L_{i-1}, R_{i-1}; k_i) = (L_i, R_i)$.

ii) The rounds that are the $g$ inverse function, i.e. $g^{-1}(L_i, R_i; k_i) = (L_{i-1}, R_{i-1})$. Such functions operate in the second cycle.

iii) Finally there are two rounds, one in the first cycle and the other in the third cycle, using in each case $PV \circ PV^{-1}$ permutations. These functions can be written as follows: $g_{PV}$ and $g_{PV^{-1}}$.

Now, to show that type i) functions are Bijective we proceed as follows: Suppose one of these rounds, say, the $i$-th round has two different input strings $L_{1,i-1}, R_{1,i-1} \neq L_{2,i-1}, R_{2,i-1}$ and the outputs are equal, which means that $L_{1,i}, R_{1,i} = L_{2,i}, R_{2,i}$. Thus if $L_{1,i}, R_{1,i} = L_{2,i}, R_{2,i}$ it follows that $R_{1,i-1} = L_{1,i}$ and $R_{2,i-1} = L_{2,i}$ so $R_{1,i-1} = R_{2,i-1}$ since by hypothesis $L_{1,i} = L_{2,i}$. However, if the right blocks entries are the same, then the left input blocks are equal. Note that the same key is used, and $R_{1,i} = R_{2,i}$ by hypothesis; but this implies that both input
strings are equal. Thus, the last conclusion contradicts the hypothesis, that is, they are different. Thus, if \( L_{1,i-1}, R_{1,i-1} \neq L_{2,i-1}, R_{2,i-2} \), can be concluded as \( L_{1,i}, R_{1,i} \neq L_{2,i}, R_{2,i} \).

For part ii), the proof is simple since the \( g \) inverse when \( g \) is a one to one function implies \( g \) inverse is a Bijective function too. It was proved in subsection i) that \( g \) is Bijective, so the conclusion is immediate.

In case iii) first it can be shown permutations defining a one to one function, in particular \( PV \) or \( PV^{-1} \) are Bijective functions. The test is simple by using \textit{reductio ad absurdum}, i.e. the hypothesis that two different input strings are considered, involving the same output and finally this contradicts the hypothesis. So it can be concluded that type iii) rounds define Bijective functions, since each can be seen as a –composite function– between a permutation and a type i) function. \( \square \)

4 Results presentation

This section starts with an example giving particular values, such as: the plaintext, \( n_i \) with \( i = 1, 2 \) and 3, plus \( K^1 \) and \( K^2 \) 64 bit keys. Moreover, the plaintext was selected in a pseudo random way, using the following criterion: a prime number is selected in a pseudo random way and later multiplied by \( \pi \) [9], and then taking the decimal point to the right of the 96 bits block and encrypting it according to the Triple-DES-96 algorithm.

Example 4.1. \textit{Let}s have the following values:

\[
\begin{align*}
n_1 &= 545454541212121212165456465465 \\
n_2 &= 455464512121545402122111316546 \\
n_3 &= 654564515454548745455665566566
\end{align*}
\]

The resultant permutation from the three previous numbers is shown in the table 1.

\[
\begin{align*}
K^1 &= 6123456789abcdef \\
K^2 &= 8120456789fbcdea
\end{align*}
\]

The selected plaintext is presented next: F3D7DB48DC3983E109D376B4

The text encrypted is: F26237E72276E3D900BCA972

With regard to the percentage of time used by Triple-DES-96 with respect to Triple-DES the following was found: \((0.4308)(t)\) approximately. Which means Triple-DES-96 is more than twice as fast as Triple-DES. The procedure for reaching the last result is written below.

Denoted by \( s \) the S file size is selected in a pseudo random way; calling the cipher times of S file such as \( t_1, t_2 \), where \( t_1 \) is the Triple-DES cryptosystem encryption time and \( t_2 \) is the proposed cryptosystem encryption time, using a variable permutation. Then \( t_2/t_1 = r(s) \) is the ratio between triple-DES-96
The triple-DES-96 cryptographic system

| 0 1 2 8 31 9 13 19 |
|---|---|---|---|---|---|---|---|
| 25 30 5 7 4 18 12 14 |
| 15 10 16 24 27 17 21 3 |
| 29 6 11 28 20 22 26 23 |
| 32 33 34 58 59 50 47 60 |
| 45 36 38 41 52 44 55 51 |
| 46 37 61 42 63 57 53 43 |
| 35 40 39 56 48 62 54 49 |
| 64 65 66 91 74 84 81 67 |
| 87 90 68 73 71 88 86 93 |
| 75 92 72 79 69 76 80 83 |
| 85 82 77 70 95 89 94 78 |

Table 1: Permutation obtained by $n_1$, $n_2$, and $n_3$

encryption time and the Triple-DES cryptosystem encryption time. Therefore, what is sought can be expressed as follows:

$$\lim_{s \to \infty} r$$

In other words, find the $r$ value as the file size increases.

Thus, it is necessary to present a practical result so; a table is presented with 5 increasing files size $s_1, s_2, ..., s_5$. These files were selected according to the following process: 5 prime numbers were chosen in a pseudo random way and subsequently each multiplied by $\pi$. In each case, the selection was that after the decimal point the necessary numbers in order to get the file sizes that are now written: 61.44 Mb, 122.88 Mb, 245.76 Mb, 491.52 Mb and 983.04 Mb. The Table 2 shows their values for each file mentioned above.

<table>
<thead>
<tr>
<th>File size (Mb)</th>
<th>Rate $r = t_2/t_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>61.44</td>
<td>45.16%</td>
</tr>
<tr>
<td>122.88</td>
<td>44.26%</td>
</tr>
<tr>
<td>245.76</td>
<td>43.08%</td>
</tr>
<tr>
<td>491.52</td>
<td>43.08%</td>
</tr>
<tr>
<td>983.04</td>
<td>43.08%</td>
</tr>
</tbody>
</table>

Table 2: The $r$ ratio between Triple-DES-96 and Triple-DES of 5 file sizes

5 Conclusions

This paper presents a symmetric cryptosystem, Triple-DES-96, which encrypts 96 bit blocks. Furthermore, it suppresses the P Triple-DES permutation, which
is applied in each encryption round [6]. Moreover, for modern computers working with 64-bit words, there is no difference in time when performing operations, in this context, with 32-bit strings than 48-bit chains. These conditions give this result: the proposed cryptosystem is more than twice as fast as the Triple-DES. On the other hand, Triple-DES is a current cryptosystem which is still used in some corporations [1].

The proposed cryptosystem has some characteristics of recent algorithms such as the SPN family (Substitution Permutation Network), since these algorithms start encryption with the xor operation using program keys. In fact, this ensures that the \( PV \) variable permutation at the third round entrance of the first cycle is protected by \( k_1, k_2 \) program keys; its inverse \( PV^{-1} \) at the beginning of the third cycle fifteenth round is protected by \( k_{15}, k_{16} \).

In the case of the Triple-DES algorithm that starts with a fixed and known permutation this does not increase the complexity of the system. Moreover, in the proposed cryptosystem the xor operations with the program keys, together with the \( PV \) and \( PV^{-1} \) permutations remove the temptation of possible linear [12] and differential [3] attacks, at least as currently conducted.

The Triple-DES-96 encryption program and Triple DES were run for various sized files intending to observe the stability of the time percentages with respect to one another. The data was presented in table 2. The Triple-DES-96 cryptosystem can also be used in the images encryption [16] and since it is a symmetric cryptosystem, it is faster to encrypt images than asymmetric cryptosystems, such as RSA [13], ElGamal [12] or Elliptic curve [11]. The proposed Triple-DES-96 is intended to be an alternative option to Triple-DES, particularly in those corporations that still use this cryptosystem.

It is also noted that variable permutation reinforces the Triple-DES-96 cryptosystem, although this work does not show how much Triple-DES-96 cryptosystem was reinforced such as in the DES cryptosystem with the variable permutation [7]. We can say that if a file is encrypted with the features mentioned above, three positive integers \( n_1, n_2 \) and \( n_3 \) indicate the permutation and \( K^1, K^2 \) keys. When decrypting the file, knowing the \( K^1, K^2 \) keys and only changing one \( n_i \) the result is not the original text, that is, it requires more information to decipher. In this sense it is said that the variable permutation reinforces the proposed cryptosystem. Future works will develop the Triple-DES-96 cryptosystem onto an FPGA in order to help companies that have implemented the Triple-DES cryptosystem to switch to Triple-DES-96.

**Acknowledgements.** The authors would like to thank the Instituto Politécnico Nacional (Secretaría Académica, COFAA, SIP, CIDETEC, ESFM and ES-CoM), the CONACyT, and SNI for their economical support to develop this work.
References


Received: September 12, 2013