Chapter 8

Proposed Model
8.1 Introduction

In this chapter a model of cryptographic technique through cascaded implementation embodied with proposed techniques has been discussed. The approach of cascaded implementation is an attempt to integrate the independent techniques which are discussed and analyzed in earlier chapters. The technique proposed in this chapter introduces a new dimension in the endeavor of ensuring secured session key generation and exchange for encryption/decryption to the maximum possible level.

Section 8.2 represents a brief description of the proposed technique. Section 8.3 deals with the detailed analysis of the results. Conclusions are drawn in section 8.4. Future scopes are described in section 8.5.

8.2 The Model

Five independent secured session key generation techniques proposed in various chapters, termed as KSOFM (say $S_1$), DHLP (say $S_2$), CDHLP (say $S_3$), CTHLP (say $S_4$), CGTHLP (say $S_5$) and five independent secured encryption/decryption techniques proposed in various chapters, termed as Fractal Triangle based encryption/decryption (say $E_1$), Simulated Annealing based encryption/decryption (say $E_2$), Genetic Algorithm based encryption/decryption (say $E_3$), ACI based encryption/decryption (say $E_4$), PSI based encryption/decryption (say $E_5$) have been integrated to generate the cascaded model. Number of cascaded stages, say $n$, is selected randomly which forms the part of the composite key of this model. This model is based on cascaded implementation of n number of techniques which are chosen randomly from among five key generation techniques and five encryption/decryption techniques with or without repetition of the same. Repetition of the same technique in consecutive cascading stage is not allowed. No technique be implemented more than $t$ number of times where $t < n$. It may so happen that one or more out of the five key generation techniques and five encryption/decryption techniques for cascading not implement at all. The plaintext is the input stream of the first encryption technique of the sequence and the output stream generated from the nth stage of cascading in the cipher text. Whenever a technique is selected for encryption, a session key generation technique also gets selected. As a result n numbers of different encryption/decryption sub keys get generated for
this implementation. A session key also gets generated using any of the session key generation techniques among five techniques. This session key helps to transmit the information regarding value of \( n \) (number of cascading stages), the order of encryption/decryption techniques for \( n \) cascading stages (say \( E_i E_j E_k \ldots \) where \( i \neq j, j \neq k, \ldots \)) and the context of \( n \) encryption/decryption keys \( (K_1, K_2, K_3, ..., K_n) \). During decryption, the cipher text is considered as binary bit stream and passes through each of \( n \) decryption techniques in exactly the reverse order of the sequence followed during encryption. The final output stream generated from \( n \)th stage of cascading reproduced the plaintext. At any intermediate stages of this technique, the output stream of the technique of that stage is the input stream to the next cascading stage.

Section 8.2.1 describes the generation of session key of the proposed cascaded implementation and that of encryption and decryption process of the same described in the section 8.2.2 and 8.2.3 respectively.

### 8.2.1 Session Key Generation

The detailed mechanisms of session key generation for individual cryptographic techniques have been discussed in respective chapters. Proposed model has \( n \) number of cascading stages. At each stage, the input binary bit stream \( P_i \) passes through the encryption/decryption key generator to generate the corresponding encryption/decryption key \( K_i \) where \( i \in N \), the set of first \( n \) natural numbers. A session key \( S_i \) is generated for the proposed model and this session key used to transmit the following information to the other party.

1. **The value of \( n \) (number of cascading stages)**

2. **The order of encryption techniques for \( n \) cascading stages (say \( E_i E_j E_k \ldots \) where \( i \neq j, j \neq k, \ldots \), and every \( E_i \in \{E_1, E_2, E_3, E_4, E_5\} \)**

3. **The information of \( n \) number of encryption/decryption keys (say \( K_1, K_2, K_3, ..., K_n \)) which are generated at the corresponding cascading stage of encryption using the input binary bit stream for that stage.**
The key space of the session key $S$ is very large. The value of $n$ can be represented by a character having ASCII value from 1 to 255. At each time a session key generation technique is selected for the whole process out of five different session key generation technique ($S_1, S_2, S_3, S_4, S_5$) randomly based on some constraints and at each cascading stage a cryptographic technique is selected out of five different encryption/decryption technique ($E_1, E_2, E_3, E_4, E_5$) randomly based on some constraints. For each session key generation technique only three bits are required to store (the three bits combinations range 000 to 111 is sufficient to store the session key generation technique index 1 to 5). For each encryption/decryption technique only three bits are required to store (the 3 bit combinations range 000 to 111 is sufficient to store the encryption/decryption technique index 1 to 5). So $(3 \times n)$ number of bits i.e. $\left(\frac{3 \times n}{8}\right)$ number of characters are required to store the sequence of cryptographic techniques for $n$ cascading stages. In various chapters, five different, independent encryption/decryption key generation techniques have been discussed in detail. All of these techniques generate 128/192/256 bits encryption/decryption keys. Proposed model has $n$ number of cascading stages and at each stage of encryption an encryption key is generated for that corresponding technique. Each encryption key has a length of 128/192/256 bits So, the length of $n$ number of encryption/decryption keys is $(128 \times n)$ to $(256 \times n)$ number of bits In various chapters, five different, independent session key generation techniques have been discussed in detail. All of these techniques generate 128/192/256 bits tuned session keys. This tuned session key get *Exclusive-OR* with the session key produced by this proposed model and transmitted to the other party. The receiving party has the same tuned session key, using this tuned session key receiving party perform the *Exclusive-OR* operations on the receiving stream to get back the session key of the proposed model. The proposed model has a session key of length [(value of number of cascading stages in bits) + (three bits combinations of encryption/decryption technique index) + (length of $n$ number of encryption/decryption keys in bits) + (length of $n$ number of session keys in bits)] i.e. \[
[8 + (3 \times n) + (128 \times n) + (128 \times n)] \text{ bits to } [8 + (3 \times n) + (256 \times n) + (256 \times n)] \text{ number of bits. So, }
\]
\[
\frac{8 + (3 \times n) + (128 \times n) + (256 \times n)}{8} = 1 + \left(\frac{3 \times n}{8}\right) + 16n + 16n = 32n \text{ to }
\]
\[
\frac{8 + (3 \times n) + (256 \times n) + (256 \times n)}{8} = 1 + \left(\frac{3 \times n}{8}\right) + 32n + 32n = 64n \text{ numbers of characters this confirms a huge variability of the key space in terms of randomness.}
\]
8.2.2 Encryptor Module

The detailed discussion on encryption techniques of all schemes \((E_1, E_2, E_3, E_4, E_5)\) have been made individually into its respective chapters. Proposed model has \(n\) number of cascaded stages where \(n\) is a finite random integer. The plaintext is a binary bit stream which is the input for the first chosen encryption technique. The output stream of \(n^{th}\) technique is the cipher text. At any intermediate stages of this approach, the output stream of the encryption technique is made input to the next cascading stage. The sequence of encryption techniques is selected randomly. The assumption of the model is that the consecutive repetition of same technique is not permitted. No technique be implemented more than \(t\) number of times where \(t < n\). One or more out of the five available techniques for cascading \((E_1, E_2, E_3, E_4, E_5)\) may not be implemented at all. In maiden stage any one out of five techniques can be chosen in five ways and then for remaining stages \((n - 1)\) times, at each cascading stage any one out of four (since consecutive repetition of same technique is not allowed) techniques can be chosen in four ways. So, there are as many as \(5 \times 4^{n-1}\) ways to choose a cascading sequence. Now, \(5 \times 4^{n-1} = \frac{5}{4} \times 4^n = 1.25 \times 4^n\), which means that the formation of session key is order of \(4^n\) ways which is a huge one. It also indicates that the key space of the session key is very large. The most importantly, it is to be noted that the session key is used only once for each transmission. So there is a time stamp of minimum span which expires automatically at end of transmission. By notation, the sequence of encryption techniques for \(n\) cascading stages is represented as \(E_iE_jE_k... E_uE_vE_w\) where, \(i \neq j, j \neq k, ... , u \neq v, v \neq w\).
Input : Source stream i.e. plaintext
Output : Encrypted stream i.e. cipher text
Method : The process takes binary stream and generates encrypted bit stream through cascaded encryption operations.

Step 1. The input stream, say $P_0$, is taken as a stream with finite number of binary bits
Step 2. Obtain the number of cascaded stages, say $n$, randomly
Step 3. Set $i = 0$ and initialize $T_0 = E_0$ (i.e. Null)
Step 4. The encryption key $K_{i+1}$ is generated using the binary bit stream $P_i$
Step 5. Select $T_{i+1} \in \{E_1, E_2, E_3, E_4, E_5\}$ randomly in such a way that $T_{i+1} \neq T_i$
Step 6. The bit stream $P_i$ is encrypted into $P_{i+1}$ using the encryption technique $T_{i+1}$ and the key encryption $K_{i+1}$
Step 7. Set $i = i + 1$. If $i < n$ then go to step 8 else go to step 9
Step 8. $P_i$ is the input stream for the next cascading stage and go to step 4
Step 9. $P_i \cong P_n$ is the final output of the encryptor module i.e. $P_n$ is the cipher text.

Figure 8.1 shows the flowchart of encryptor module for the proposed model.
Figure 8.1: Pictorial representation of the flow chart of encryption for the proposed cascaded model

Start

Input $P_0$ is taken as binary stream

Initialize the number of cascaded stages, $n$

Set $i = 0$ and $T_0 = E_0$ (Null)

Encryption key $K_{i+1}$ is generated using the $P_i$ as a binary bit stream with key generator

Select $T_{i+1} \in \{ E_1, E_2, E_3, E_4, E_5 \}$ randomly in such a way that $T_{i+1} \neq T_i$

The bit stream $P_i$ is encrypted into $P_{i+1}$ using the encryption technique $T_{i+1}$ and the key encryption $K_{i+1}$

Set $i = i + 1$

Is $i < n$ ?

Yes

No

$P_i \cong P_n$ is the cipher text

Stop
8.2.3 Decryptor Module

The detailed discussion on decryption techniques of all schemes \((D_1, D_2, D_3, D_4, D_5)\) have been discussed individually into its respective chapters. Proposed model has \(n\) number of cascaded stages where \(n\) has selected at random during decryption. Processing the information of the session key \(S\), the value of \(n\) (number of cascading stages), the order of the encryption technique for \(n\) cascading stages \((E_i, E_j, E_k, ... E_v, E_w\) where, \(i \neq j, j \neq k, ..., u \neq v, v \neq w\) and the information of \(n\) number of encryption/decryption keys \((K_1, K_2, ..., K_n)\) are fetched during decryption. Order of decryption is fetched from the session key \(S\) which is exactly reverse of the sequence of encryption and initialized into the variables \(T_1, T_2, T_3, ..., T_n\) (i.e. \(T_1\) is the first decryption technique, \(T_2\) is the second decryption technique and so on) where \(T_i \in \{D_1, D_2, D_3, D_4, D_5\}\) for \(i \in N\), the set of first \(n\) natural numbers. The order of the decryption is exactly the reverse of the sequence followed during encryption i.e. \(D_w, D_v, D_u, ... D_i, D_j, D_i\) and the decryption key which will be used during decryption in the order of \(K_w, K_{n-1}, ..., K_2, K_1\). The first decryption technique \(D_w\) considers the input cipher text \(P_n\) as a binary bit stream. The final output stream generated from the final stage of cascading using the decryption technique \(D_i\) reproduced the plaintext. At the intermediate stages of this approach, the output stream of any decryption technique is the input stream to the next cascading stage.

The decryption algorithm is described as follows:

**Input** : Encrypted stream i.e. cipher text and the session key \(S\)

**Output** : Source stream i.e. plaintext

**Method** : The process takes encrypted binary stream and generates decrypted bit stream through cascaded decryption operations.

**Step 1.** The stream containing the information of the session key \(S\) obtained to get the information about the decryption key

**Step 2.** The value of \(n\) (number of cascading stages) and the decryption keys \(K_1, K_2, K_3, ..., K_n\) are extracted from the session key \(S\) and used for decryption

**Step 3.** The order of decryption is fetched from the session key \(S\) which is exactly reverse of the sequence of encryption and initialized into the
variables $T_1, T_2, T_3, \ldots, T_n$ (i.e. $T_1$ is the first decryption technique, $T_2$ is the second decryption technique and so on) where $T_i \in \{ D_1, D_2, D_3, D_4, D_5 \} \forall i \in N$, the set of first $n$ natural numbers

**Step 4.** The input stream, say $P_n$, is taken as a stream with finite number of binary bits.

**Step 5.** Set $i = n$

**Step 6.** Input bit stream $P_i$ is decrypted into $P_{i-1}$ using the decryption technique $T_{n-i+1}$ and the decryption key $K_i$

**Step 7.** Set $i = i - 1$. If $i > 0$ then go to step 8 else go to step 9

**Step 8.** $P_i$ is the input stream for the next cascading stage and goes to step 6

**Step 9.** $P_i \cong P_0$ is the final output of the decryptor module i.e. $P_0$ is the plaintext.

Figure 8.2 shows the flowchart of decryptor module for the proposed model.
Figure 8.2: Pictorial representation of the flow chart of decryption for the proposed cascaded model
8.3 Analysis

Extensive analysis has been made with a huge variability of the value of $n$, the number of cascading stages. Out of eighty samples files (four different types and twenty files for each type) only twelve files, arbitrarily chosen three from each type, taken for testability of the model and result are generated.

The encryption and decryption times taken are the differences between processor clock stick at the starting of execution and at the end of execution respectively. Since the CPU clock ticks taken as time, there might be a slight variation in actual time which is insignificant and may be ignored. Proposed model has $n$ number of cascading stages. So the encryption and decryption time of the proposed approach are near equal to the cumulative sums of $n$ number of encryption and decryption times respectively of the individual cryptographic techniques. Therefore the encryption and decryption times are larger than that of any individual method.

Comparison between the source and encrypted bytes has been performed and changes of bits within encrypted bytes has been observed for a change of single bit in the original message byte for the entire or a relative large number of bytes. Detail concept of Avalanche, Strict avalanche and Bit independence test has been discussed in chapter 7. The values of three above mentioned tests are based on pure numbers and this has no units. The calculated Avalanche, Strict avalanche and Bit independence values are very high which may indicate good security of the proposed approach. There are no significant differences observed between the calculated avalanche, Strict avalanche and Bit independence values for the Proposed approach using cascaded implementation and that for any individual technique.

Spectrum of the frequency distribution of the encrypted characters generated using the proposed approach are analyzed and it is observed that characters with ASCII values ranging from 0 to 255 appeared all with near equal frequencies which may indicate that it is very hard to regenerate the original file for a cryptanalyst. Difference between high and low value of frequencies in the frequency distribution curves is very small. So the spectrum of frequency distribution generated using the proposed approach are nearly smoother which may indicate that the degree of security of the proposed approach is good. No remarkable differences in the spectrum of frequency distribution have been observed for the cascaded implementation.
Chi-Square value is calculated from the character frequencies using the formula devised by Karl Pearson which is called “Pearsonian Chi-Square”. The higher the Chi-Square values the more deviation from the original message. In chapter 7 the detail concept of the test of non-homogeneity has been discussed. The calculated Chi-Square values for all the sample files using the proposed approach are very large compare to tabulated one which may indicate that the degree of security of the proposed approach using cascaded implementation is good. There is no noticeable difference has been observed from calculated Chi-Square values, which confirms the high degree of non-homogeneity of the encrypted stream with respect to the source stream.

Cryptographic algorithms are possible to break without keys where a cryptanalyst try all possible keys until get the success. The security of a cryptographic scheme depends on how much effort along with its time stamp is required for the cryptanalyst to break it. But it is always very difficult to estimate the amount of effort required to decrypt the cipher text successfully. In other word, an encryption scheme may be defined as unconditionally secured if the cipher text generated by the scheme does not contain enough information to determine uniquely the corresponding plaintext, no matter how much cipher text is available.

The complexity of any symmetric encryption algorithm is generally compared with the Brute-force attack. Brute-force approach simply involves computing every possible key until an intelligible translation of the cipher text into plaintext is obtained. On average, half of all possible keys must be tried to achieve success. Table 8.1 shows how much time is involved for various key spaces. The 56-bit key size is used with the DES algorithm and 168-bit key size is used for triple DES (i.e. TDES) algorithm. The minimum key size specified for AES is 128 bits. For each key size, the results are shown assuming that it takes 1µs and 10^6 µs to perform a single decryption respectively.
Table 8.1
Time involved for various key spaces

<table>
<thead>
<tr>
<th>Key Size</th>
<th>Number of Alternate Keys</th>
<th>Time required at 1 decryption/µs</th>
<th>Time required at $10^6$ decryption/µs</th>
</tr>
</thead>
<tbody>
<tr>
<td>32 bits</td>
<td>$2^{32} = 4.3 \times 10^9$</td>
<td>$2^{31} \mu s = 35.8$ minutes</td>
<td>$2.15$ milliseconds</td>
</tr>
<tr>
<td>56 bits</td>
<td>$2^{56} = 7.2 \times 10^{16}$</td>
<td>$2^{55} \mu s = 1142$ years</td>
<td>$10.01$ hours</td>
</tr>
<tr>
<td>128 bits</td>
<td>$2^{128} = 3.4 \times 10^{38}$</td>
<td>$2^{127} \mu s = 5.4 \times 10^{24}$ years</td>
<td>$5.4 \times 10^{18}$ years</td>
</tr>
<tr>
<td>168 bits</td>
<td>$2^{168} = 3.7 \times 10^{50}$</td>
<td>$2^{167} \mu s = 5.9 \times 10^{36}$ years</td>
<td>$5.9 \times 10^{30}$ years</td>
</tr>
<tr>
<td>26 characters (permutation)</td>
<td>$26! = 4 \times 10^{26}$</td>
<td>$2 \times 10^{25} \mu s = 6.4 \times 10^{12}$ years</td>
<td>$6.4 \times 10^6$ years</td>
</tr>
</tbody>
</table>

In section 8.2.1 of this chapter, the length of the session key of the proposed model has been discussed and it has approximately $64n$ number of characters (any character with ASCII value from 0 to 255). Therefore number of alternate keys = $256^{64n}$. Since on an average half of all possible keys must be tried to achieve success, so total time required at 1 decryption/µs = $0.5 \times 256^{64n} \mu s = 0.5 \times 2^{8 \times 64n} \mu s = 0.5 \times 2^{512n} \mu s = 2^{(512n-1)} \mu s$. Table 8.2 shows how much time is involved for exhaustive search of the session key $S$ for the proposed model with various $n$ values. Analyzing the data of this table it may be concluded that the proposed model is highly secured from Brute-force attack.

Table 8.2
Average time required for exhaustive key search

<table>
<thead>
<tr>
<th>n values</th>
<th>Number of Alternate Keys</th>
<th>Time required at 1 decryption/µs</th>
<th>Time required at $10^6$ decryption/µs</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$256^{64 \times 1} = 2^{512}$</td>
<td>$2^{511} \mu s = 2.13 \times 2^{140}$ years</td>
<td>$2.13 \times 2^{134}$ years</td>
</tr>
<tr>
<td>2</td>
<td>$256^{64 \times 2} = 2^{1024}$</td>
<td>$2^{1023} \mu s = 2.85 \times 2^{294}$ years</td>
<td>$2.85 \times 2^{288}$ years</td>
</tr>
<tr>
<td>3</td>
<td>$256^{64 \times 3} = 2^{1536}$</td>
<td>$2^{1535} \mu s = 3.82 \times 2^{448}$ years</td>
<td>$3.82 \times 2^{442}$ years</td>
</tr>
<tr>
<td>4</td>
<td>$256^{64 \times 4} = 2^{2048}$</td>
<td>$2^{2047} \mu s = 5.12 \times 2^{602}$ years</td>
<td>$5.12 \times 2^{596}$ years</td>
</tr>
<tr>
<td>5</td>
<td>$256^{64 \times 5} = 2^{2560}$</td>
<td>$2^{2559} \mu s = 6.87 \times 2^{756}$ years</td>
<td>$6.87 \times 2^{750}$ years</td>
</tr>
<tr>
<td>6</td>
<td>$256^{64 \times 6} = 2^{3072}$</td>
<td>$2^{3071} \mu s = 9.21 \times 2^{910}$ years</td>
<td>$9.21 \times 2^{904}$ years</td>
</tr>
<tr>
<td>7</td>
<td>$256^{64 \times 7} = 2^{3584}$</td>
<td>$2^{3583} \mu s = 1.23 \times 2^{1065}$ years</td>
<td>$1.23 \times 2^{1059}$ years</td>
</tr>
<tr>
<td>8</td>
<td>$256^{64 \times 8} = 2^{4096}$</td>
<td>$2^{4095} \mu s = 1.66 \times 2^{1219}$ years</td>
<td>$1.66 \times 2^{1213}$ years</td>
</tr>
</tbody>
</table>
Let $T$ be the average time in years required at $10^6$ decryptions per $\mu$s for exhaustive search of the session key for the proposed model. If $n$, number of cascading stages, is plotted along X-axis and $\log_{10} T$ along Y-axis then the generated curve is a straight line which is shown in figure 7.3. Extending this straight line along positive X-axis, it may predict the required average time $T$ in years for any large value of $n$. Since the slope value of that straight line is very high and the value of $T$ is plotted as $\log_{10} T$, so the value of $T$ is increased sharply with the increase of $n$.

![Graphical representation of average time $T$ in years (T in logarithmic scale as $\log_{10} T$) against n, number of cascading stages](image)

figure.8.3: Graphical representation of average time $T$ in years (T in logarithmic scale as $\log_{10} T$) against n, number of cascading stages

An ideal encryption procedure should be sensitive with the secret key. It indicates that the change of a single bit in the secret key should produce a completely different cipher stream and the decryption with a slightly different key fails completely. It is observed that the proposed model generates an entirely different cipher stream with the change of a single bit randomly in the key $K$. It is also noticed that the model totally fails to decrypt the cipher stream into plaintext with a slightly different secret key. From this point of view, it may be concluded that the proposed model is highly key sensitive.
8.4 Conclusions

The approach of cascaded implementation is very simple and logical. The analysis of results also indicates enhanced security of this approach. The strength of this proposed approach through cascaded implementation is not highlighted through the metrics for evaluation. No universal conclusion can be drawn from that above discussion regarding this approach. The real strength of the proposed approach lies in the possible formation of a large key space. The key space increases drastically with allowing the much more cascading stages. The proposed model is highly secured from Brute-force attack. Other strength of this proposed model is the adoption of complexity based on energy and resource available in the wireless communication, infrastructure for computing in a node or mesh in wireless communication. For a wireless network having low energy, the number of cascading stages be less. So, the model is very much suitable for the security of the system where energy and resource is one of the main constraints. One of the most important features of the proposed model is that the model is idle to trade-off between security and performance of light weight devices having very low processing capabilities or limited computing power. The proposed model is applicable to ensure very high security for file transmission in any form in any size.

Some of the salient features of proposed technique can be summarized as follows:

a) Session key generation and exchange – The session key can be formed in order of $4^n$ ways which is a vast one and the length of session key is approximately $64n$ number of characters where $n$ is the number of cascading stages. It indicates that the key space of the session key is very large. Again identical tuned session key can be generate after the tuning of network in both sender and receiver side using any of the proposed key generation techniques. This tuned session key can be used to encrypt session key generated by the proposed model for transmission to the other party which provides another level of security. Since the session key is used only once for each transmission, so there is a minimum time stamp which expires automatically at the end of each transmission of information.

b) Degree of security – Proposed technique does not suffers from cipher text only Attack, known plaintext attack, chosen plaintext attack, Chosen cipher text only attack, brute
force attack. The number of alternate keys for the proposed model is approximately $256^{64^n}$. So, model is highly secured from Brute-force attack.

c) Variable block size – Encryption algorithm can work with any block length and thus not require padding, which result identical size of files both in original and encrypted file. So, proposed technique has no space overhead.

d) Variable size key –variable size session key with high key space can be used in different session. Since the session key is used only once for each transmission, so there is a minimum time stamp which expires automatically at the end of each transmission of information. Thus the cryptanalyst will not be able guess the session key for that particular session.

e) Complexity – Proposed technique has the flexibility to adopt the complexity based on infrastructure, resource and energy available for computing in a node or mesh through wireless communication. So, the proposed technique is very much suitable in wireless communication.

f) Non-homogeneity – Measures of central tendency, dispersion and Chi-Square value have been performed between the source and corresponding cipher streams generated using proposed technique. All the measures indicate that the degree of non-homogeneity of the encrypted stream with respect to the source stream is good.

g) Floating frequency – In this proposed technique it is observed that floating frequencies of encrypted characters are indicates the high degree of security of proposed technique.

h) Entropy – In this proposed technique it is observed that entropy of encrypted characters is near to eight which indicate the high degree of security of proposed technique.

i) Correlation between source and encrypted stream – The cipher stream generated through proposed technique is negligibly correlated with the source stream. Therefore the proposed technique may effectively resist data correlation statistical attack.

j) Key sensitivity – Proposed method generates an entirely different cipher stream with a small change in the key and technique totally fails to decrypt the cipher stream with a slightly different secret session key.
k) **Trade-off between security and performance** – The proposed technique may be ideal for trade-off between security and performance of light weight devices having very low processing capabilities or limited computing power in wireless communication.

### 8.5 Future Scope

The work presented in this thesis leaves further investigations in some areas. Some of the apparent future investigations may be:

a) **Intermixing of other popular soft computing based approach** – Some well-known and popular soft computing based approaches like Artificial Immune System (AIS), Differential evolution, Support Vector Machine (SVM), Fuzzy Logic can be intermixed for generation of session key by tuning in wireless communication.

b) **Effectiveness**- Comparisons of the proposed model with other well-known and popular cryptographic algorithms like Blowfish, RC2, RC5, TEA, XTEA, IDEA, Serpent etc. can be performed to ensure the effectiveness further of the proposed technique.

c) **Differential Analysis** - It may be possible to find out a meaningful relationship between the source stream and encrypted stream making a slight change such as modifying a single bit of the encrypted stream. If one minor change in the source stream can cause a significant change in the encrypted stream then this differential attack would become very inefficient and practically useless. To resist the differential attack differential analysis on the encrypted stream is necessary.

d) **Encryption Quality** – A measure of encryption quality of the proposed model may be expressed as the deviation between the source stream and the encrypted stream. The encryption quality is also a function of secret key length.

In spite of various limitations and scope of future upgradability, there are good potential in each of individual proposed soft computing based cryptographic techniques and also in the proposed model. Incorporation of tuning of networks over public channel for generation of session key introduced a novel idea out of which more security may be obtained. In the proposed model, a huge variability of key space has been introduced which is most sensitive to the cipher text with the minimal change. The proposed model is highly flexible to adopt
the complexity of any light weight computing system and idle to trade-off between security and performance of light weight device in wireless communication having limited resources. From the study, incorporation of proposed techniques the security of wireless communication may be enhanced. As a result, the proposal of the thesis may be useful for the researchers and stakeholders.