International Journal of Computer Mathematics
Vol. 89, No. 16, November 2012, 2123–2142

Taylor & Francis
Taylor & Francis Group

Mathematical analysis and simulation of multiple keys and S-Boxes in a multinode network for secure transmission

Ajay Kakkar[a]*, M.L. Singh[b] and P.K. Bansal[c†]

[a]*Department of Electronics and Communication Engineering, Thapar University, Patiala, Punjab, India;* [b]*Department of Electronics Technology, Guru Nanak Dev University, Amritsar, Punjab, India;* [c]*M.I.M.I.T., Malout, India*

(*Received 15 September 2010; revised version received 1 June 2011; second revision received 2 September 2011; third revision received 1 June 2012; fourth revision received 11 June 2012; accepted 13 June 2012*)

The requirement of data security is an important parameter for all organizations for their survival in the world. Cryptography is the best method to avoid unauthorized access to data. It involves an encryption algorithm and the keys that are being used by the users. Multiple keys provide a more secure cryptographic model with a minimum number of overheads. There are various factors that affect the security pattern such as the number of keys and their length, encryption algorithm, latency, key shifting time, and users. In this paper, a new approach is proposed for generating keys from the available data. The analysis of various times, such as encryption, decryption, key setup, processing, and key shifting times, has been done. The model takes minimum time to replace the faulty keys with the fresh keys. In this paper, we consider all the above-mentioned factors and suggest an optimized way of using them.

Keywords: encryption; S-Boxes; keys; key shifting time; hacking

2010 AMS Subject Classification: 94A60

1. Introduction

Security attacks against networks are increasing significantly with time using latest software. Over the years, a number of techniques and approaches have been developed to ensure data confidentiality, integrity, and availability. The techniques used for data security include multiple passwords, cryptography, and biometrics. Cryptography is a technique used to avoid unauthorized access to data. It includes an encryption algorithm and keys. The basic problem that is concerned with keys is that their strength gets degraded with time. Using powerful software packages, it is quite easy to determine the keys. It has been observed that a single key does not provide the required secure model [4,26]. The key length and number of keys and their mutual arrangement provide better security; for the above, it is mandatory to use them in an optimized manner to avoid overheads. If we use 128 key lengths to encrypt 8-bit data, then it will consume more power and also more processing time. Therefore, the best method is to use multiple keys of short length to encrypt

*Corresponding author. Email: kakkar_ajay29@rediffmail.com
†Current address: Department of Electronics and Communication Engineering, Thapar University, Patiala, Punjab, India.

ISSN 0020-7160 print/ISSN 1029-0265 online

http://dx.doi.org/10.1080/00207160.2012.704022
http://www.tandfonline.com

the data [12,25]. The main limitations in this scheme are that the short keys are more sensitive and get attacked by the hacker easily. To tackle the problem, first we determine the failure rate of the keys and calculate the time for which their security level remains in the higher level [23]. Another problem with keys is that how these are transported from the sender to the receiver. Therefore, this can be avoided by generating the keys from the available data with the help of an encryption algorithm. Using the information regarding timing and power consumption by a device during the execution of a cryptographic algorithm, cryptanalysts can break the model [15]. Therefore, the main purpose of a secure encryption algorithm is to protect the interests of parties communicating in the presence of adversaries [14]. The modelling of the behaviour of cyber attackers is difficult and determining the appropriate level of attack is very important from the security point of view. We are aware of the fact that in a multinode network (MN), security decreases with an increase in the number of nodes [13]. It is clear that the flaws in the key designing account for 30–45% of security problems, and architectural risk analysis plays an important role in any secure program [7,14,19,24]. In view of this, multiple keys are used to provide resistance against the virtual and real attacks made by hackers. The following section describes the work of various researchers in the area of data security in MNs. Data Encryption Standard is one of the most widely accepted, publicly available cryptographic systems. It was developed by IBM in the 1970s but was later adopted by the US government as a National Standard. In 1990, International Data Encryption Algorithm (IDEA) was originally developed as the Proposed Encrypted Standard, and in 1992, it was renamed as IDEA. It is a block cipher that uses 64-bit data blocks and a 128-bit key. Aiello and Venkatesan [1] described that *n*selected plaintexts in an MN can be distinguished by the hacker with the numbers from a random function. This means that it is possible to hack the model with a determined probability. Banerjee *et al.* [3] gave an overview of signalling enhancement and recovery techniques used in an MN. Such techniques are useful to determine the security of a model. Eschenauer and Gligor [11] proposed a random key establishment technique for wireless sensor networks. Lee and Griffith [17] presented a hierarchical approach to resolve multiple failures in an MN in which various security levels have been proposed for different types of attacks, and a recovery mechanism can be selected on the basis of these security levels. Chan *et al.* [8] extended the technique of *n*random key establishment that enables two neighbouring nodes to establish a secure communication only when they share n common keys (where $n \geq 2$). Du *et al.* [10] developed two similar random key pre-distribution techniques that use the multi-space key pool to improve network resilience and memory usage efficiency [18]. Hundessa and Domingo-Pascual [16] presented a protection mechanism packed with multiple key(s) to handle multiple link/node failures. Furthermore, Backes and Pfitzmann [2] presented the relating symbolic and cryptographic secrecy technique for an MN. Bertino *et al.* [4] discussed an efficient time-bound hierarchical key management scheme for secure broadcasting. There are numerous cryptographic algorithms for data encryption and authentication techniques for an MN. Using encryption, an efficient generic solution for an MN was proposed by Naor *et al.* [22]. Naor's model was not compatible with multiple keys having different failure rates. Hundessa and Domingo-Pascual [16] provided data-gathering strategies over all the possible network routes. Blake and Kolesnikov [5,6] did not provide any practical ways to achieve secure re-routing schemes.

2. Motivation from the literature survey

From the literature survey, the following observations have been drawn:

- A single key with a fixed length cannot be used to provide secure communication in an MN. By knowing the data and key length, the hacker is able to generate side-channel and middle-line attacks.

- If the key length is short (1–1024 bits), it is very easy for the hacker to get the hold of the key by using various permutations and combinations. On the other hand, if large key lengths are used, it results in complexity, which increases the probability of error.
- A single key with a variable length provides little bit more secure communication than a single key having a fixed length. The technique is preferred only for short data streams in an MN having less number of nodes.
- Multiple keys having different failure rates can be achieved by varying the key length. They are always preferred for encrypting the data in an MN having a large number of nodes. Multiple keys have different failure rates:
  (i) if the length of the keys is of different order,
  (ii) if different polynomials are used for the encryption, and
  (iii) if the size of the data block varies.
- In case of node failure, the algorithm immediately generates new keys for the corresponding node. It has been found that for an efficient and reliable model, keys should be generated from the available data. Key recovery mechanisms should be available in the model in order to take care of the failure situation. There is a need to minimize the key shifting time (δ) from the first key to the second key in the case of a multiple key encryption-based system.

Keeping in mind the importance of multiple keys for secure data transmission, this work incorporated the use of multiple keys. Multiple keys were generated from the available data to reduce the overheads such as the need of sending additional bits along with the data. Eight to 16 S-Boxes were used to perform random round functions for the generation of multiple keys. There is a design and development procedure of an optimized encryption algorithm that is based on an efficient key management scheme in order to provide secure data transmission in an MN. For better security, multiple keys having a minimum key shifting time were used. The failure rate of multiple keys was evaluated and analysed mathematically to make the model secure. The analysis shows that the failure rate plays a vital role in reducing the time available to the hackers for the various attempts made to destroy the model. In the encryption process, a slight increase in the processing time was observed at various nodes, but it is acceptable because it is very small in comparison with that consumed by the existing encryption algorithms.

The objective of this work is to develop an optimized efficient key management technique(s) in order to

- generate random key(s) from the data by the algorithm,
- determine the failure rate of multiple key(s) used by various S-Boxes,
- reduce the time available for the hacker for making attempts to destroy the model, and
- minimize the key shifting time (δ) from the first key to the second key and so on.

3. Proposed work

Modern cryptography involves the use of keys for data signing, encoding, and decoding. Some keys are distributed privately between the parties, while others allow the parties to use public keys that can be broadcast openly [8,9]. The level of protection is varied for every situation and also dependent upon the work and technique used; some encryption techniques provide a virtually unbreakable barrier to information theft; others just require a determined attacker with moderate resources to be broken. One way of comparing the techniques on this level is to estimate how much CPU time would be required on a machine of a given processing speed to iterate through all the possible keys to the encoded data, based upon the permutation [27]. System-wide security is always required to make sure that the data are safe; data are safe for some time while using

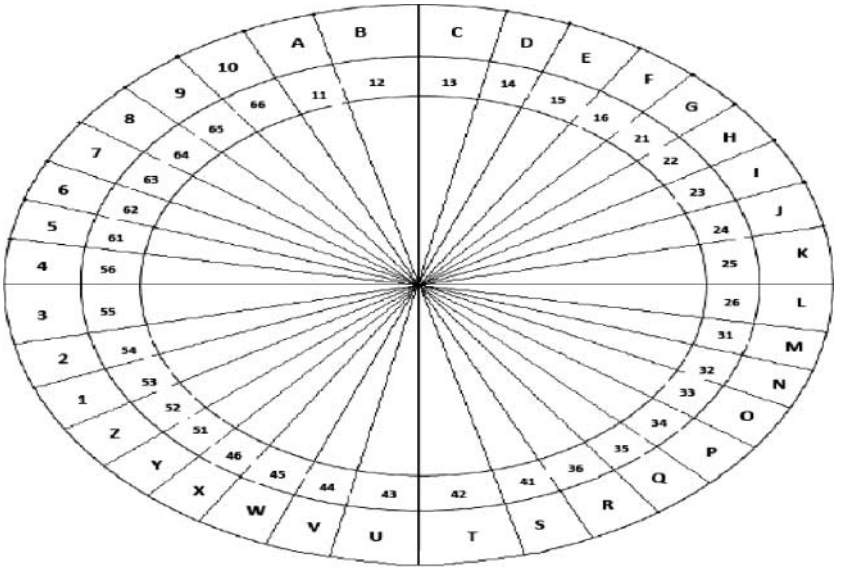

Figure 1. Conversion of alphabets into their equivalent codes.

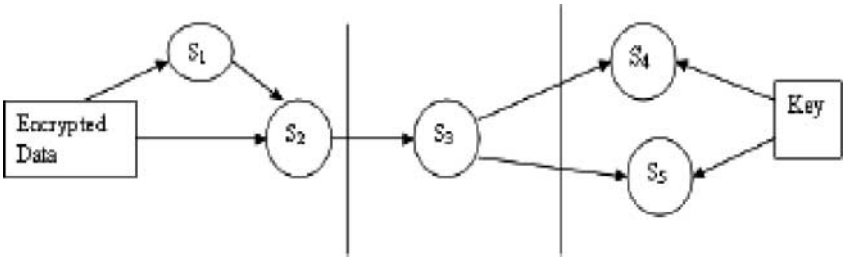

Figure 2. Various stations in a model with buffers.

security techniques, but overall system is not safe; using the information about timing, power consumption, and radiation of a device when it executes a cryptographic algorithm, cryptanalysts have been able to break the system.

Keeping in mind the importance of multiple keys for secure data transmission, this work incorporated the use of multiple keys. Multiple keys were generated from the available data to reduce the overheads such as the need of sending additional bits along with the data. The data were encoded using Figure 1 in which all the alphabets and the number having their weight were further converted into binary (0/1), for example, $25 = 010101$, $55 = 101101$, and so on. All the binary data were encoded with the help of a key and further passed through S-Boxes (having a different key for each round) as shown in Figure 2. During the transmission of data if any station fails, due to the attacks made by the hacker or by other means (atmospheric conditions), then it makes the overall model weak. To maintain the security over a model, all the parameters such as nodes, key generation mechanism, and latency need continuous attention; they must be upgraded with time. Each station is required to be packed up by the recovery mechanism. It is important to calculate the latency time concerned with a particular station:

$$D(S,K) = \sum_{i=1}^{N} d(S_i) + d(k_i),$$

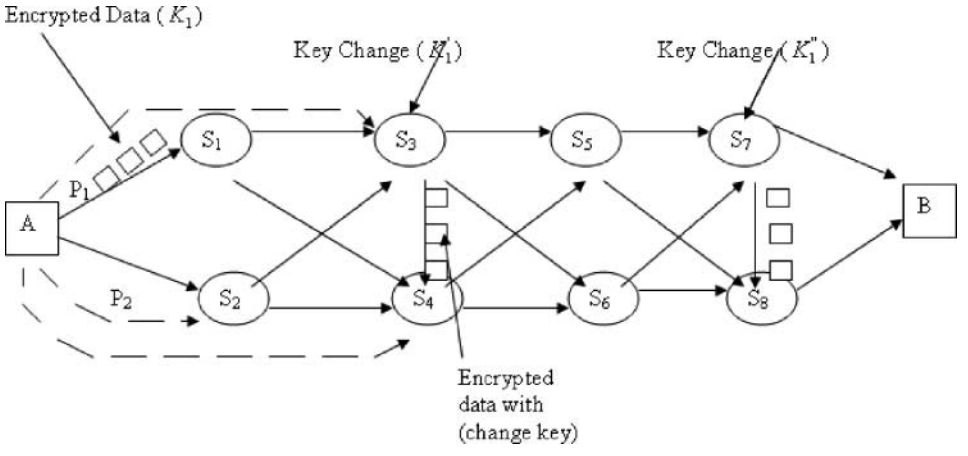

Figure 3. Different stations in a model.

where $D(S)$ and $D(K)$ are the delay caused by various stations due to the logical effort in the encryption process and delay caused by the various keys to get generated from the data, respectively. In case of failure of a station, let us take a case in which station S_2 is hacked by the hacker, then the data of the same station are moved to the neighbouring station S_3; in such a case, the buffers g and h available at S_3 also result in a delay (Figure 3). The total delay is calculated as

$$D(S,K) = \sum_{i=1}^{N} d(S_i) + d(k_i) + d(g_i) + d(h_i).$$

User A wishes to transmit the data in an MN having nodes $(S_1, S_2, \ldots, S_8)$. Multiple paths are provided to send the data over the path P_1, whereas encrypted packets are transmitted by A. Similarly, one can select the other paths P_2 and P_3 by considering the delay and reliability. For a secure system, all the intermediate nodes should be under the control of master node A. Failure notification of a node is immediately forwarded to the nearest node, preferably the neighbouring node, in order to reduce the latency and congestion. Fast reroute methods are employed in such situations, which deal with the change of path and are known as dynamic routing for a network. Dynamic re-routing is effective when a model has less number of nodes. Whenever the security level of a node falls below a certain level, then master node A has the power to immediately change the path and re-encrypt the data using another key. In the above exercise, we also make sure that there is no faulty node in the new path and determine the failure rates of S-Boxes:

$$\begin{aligned}\text{Path}\, P_1 : A &\longrightarrow S_1 \in (K_1) \longrightarrow S_3 \in (K_1, K_1') \longrightarrow S_4 \longrightarrow S_5 \\ &\longrightarrow S_7(K_1, K_1', K_1') \longrightarrow S_8 \longrightarrow B.\end{aligned}$$

In the above equation, if S_3 is the weak station, then either change the key for the encryption

$$\begin{aligned}\text{Path}\, P_1 : A &\longrightarrow S_1 \in (K_2) \longrightarrow S_3 \in (K_2, K_2') \longrightarrow S_4 \longrightarrow S_5 \\ &\longrightarrow S_7(K_2, K_2', K_2'') \longrightarrow S_8 \longrightarrow B\end{aligned}$$

or change the path

(i) $\text{Path}\, P_1 : A \to S_2 \in (K_1) \to S_3 \in (K_1, K_1') \to S_4 \to S_5 \to S_7(K_1, K_1', K_1'') \to S_8 \to B,$
(ii) $\text{Path}\, P_2 : A \to S_2 \in (K_2) \to S_3 \in (K_2, K_2') \to S_4 \to S_5 \to S_7(K_2, K_2', K_2'') \to S_8 \to B.$

For a highly secure system, transmission is done from $A \rightarrow S_4$.

Note: All the stations have the power to change the key and encrypt the data with it only iff A permits:

$$\text{If } X_1 = \text{inputs},\ S_1 = \text{weak stations},\ \overline{S_1} = \text{strong stations}.$$

Now, we assume that the S-Boxes are under threat due to a high failure rate of keys. The probability of recovering the data and the latency time and making the system reliable by the help of a shifting key is determined in the following section.

3.1 *Case 1: when two stations are under attacks made by the hacker (failures of two S-Boxes in a given model)*

In this case, it is required to change the key for particular stations, but this will be not treated as a reliable method, so it will be preferred to change the path; for this, it is required to determine the input data as those concerning weak stations:

$$X_1 \xrightarrow{\text{Key}} S_1,\ X_2 \xrightarrow{\text{Key}} S_2 \,\&\, X_{1,2} \xrightarrow{\text{Key}} S_1 \,\&\, S_2.$$

We know that

$$S_1 = X_1 \cup X_{1,2}, \quad S_2 = X_2 \cup X_{1,2}. \tag{1}$$

On the other hand,

$$S_1 \cap S_2 = (X_1 \cap X_2) \cup X_{1,2}. \tag{2}$$

If $x_i = P_r(X_i), x_{i,j} = P_r(X_{i,j})$.
Complement

$$P_r(X_i^C) = 1 - P_r(X_i), \tag{3}$$

$$\sum_{i=1}^{N} P(X_i) = 1. \tag{4}$$

Complement

$$\sum_{i=1}^{N} P_r(X_i^C) = \sum_{i=1}^{N} 1 - P_r(X_i) = N - 1. \tag{5}$$

Similarly, we can determine the probabilities of S-Boxes as

$$s_i = P_r(S_i), \quad s_{i,j} = P_r(S_{i,j}). \tag{6}$$

Using Equation (2), we can write

$$\begin{aligned} s_1 &= x_1 + x_{1,2} - x_1 \cdot x_{1,2}, \\ s_2 &= x_2 + x_{1,2} - x_2 \cdot x_{1,2}, \\ s_{1,2} &= x_1 \cdot x_2 + x_{1,2} - x_1 \cdot x_2 \cdot x_{1,2}. \end{aligned} \tag{7}$$

Rearrange the equation in order to get the probability of x_1, x_2, and $x_{1,2}$:

$$\begin{aligned} x_1 &= \frac{s_1 - x_{1,2}}{1 - x_{1,2}}, \\ x_2 &= \frac{s_2 - x_{1,2}}{1 - x_{1,2}}, \\ x_{1,2} &= \frac{s_{1,2} - x_1 \cdot x_2}{1 - x_1 - x_2 - x_{1,2}}. \end{aligned} \tag{8}$$

Similarly, for four weak stations in a model, it is required to determine the single failure S_i, double failures $S_{i,i}$, triple failures $S_{i,j,k}$, and quadrate failures $S_{i,j,k,m}$. For strong stations R_i, the probabilities are determined using complements (Equation (3)). $P_r(R_i) = P_r(X_i^C)$ or $r_i = P_r(R_i) = 1 - x_i$.

In terms of strong stations,

$$\begin{aligned} P_r(\overline{S_1}) &= 1 - x_1 = r_1 \cdot r_{1,2} \cdot r_{1,3} \cdot r_{1,4} \cdot r_{1,2,3} \cdot r_{1,3,4} \cdot r_{1,2,4} \cdot r_{1,2,3,4}, \\ P_r(\overline{S_2}) &= 1 - x_2 = r_2 \cdot r_{1,2} \cdot r_{2,3} \cdot r_{2,4} \cdot r_{1,2,3} \cdot r_{1,2,4} \cdot r_{2,3,4} \cdot r_{1,2,3,4}, \\ P_r(\overline{S_3}) &= 1 - x_3 = r_3 \cdot r_{1,3} \cdot r_{2,3} \cdot r_{3,4} \cdot r_{1,2,3} \cdot r_{1,3,4} \cdot r_{2,3,4} \cdot r_{1,2,3,4}, \\ P_r(\overline{S_4}) &= 1 - x_4 = r_4 \cdot r_{1,4} \cdot r_{2,4} \cdot r_{3,4} \cdot r_{1,2,4} \cdot r_{1,3,4} \cdot r_{2,3,4} \cdot r_{1,2,3,4}. \end{aligned} \tag{9}$$

Also,

$$\begin{aligned} P_r(\overline{S_1} \cap \overline{S_2}) &= P_r(\overline{S_1 \cup S_2}) = 1 - s_1 - s_2 + s_{1,2} = r_1 \cdot r_2 \cdot r_{1,2} \cdot r_{1,3} \cdot r_{2,3} \cdot r_{1,4} \\ &\quad \cdot r_{2,4} \cdot r_{1,2,3} \cdot r_{1,3,4} \cdot r_{1,2,4} \cdot r_{1,2,4} \cdot r_{2,3,4} \cdot r_{1,2,3,4}, \\ P_r(\overline{S_1} \cap \overline{S_3}) &= P_r(\overline{S_1 \cup S_3}) = 1 - s_1 - s_3 + s_{1,3} = r_1 \cdot r_3 \cdot r_{1,2} \cdot r_{1,3} \cdot r_{2,3} \\ &\quad \cdot r_{1,4} \cdot r_{3,4} \cdot r_{1,2,3} \cdot r_{1,3,4} \cdot r_{1,2,4} \cdot r_{2,3,4} \cdot r_{1,2,3,4}, \\ P_r(\overline{S_1} \cap \overline{S_4}) &= P_r(\overline{S_1 \cup S_4}) = 1 - s_1 - s_4 + s_{1,4} r_1 \cdot r_4 \cdot r_{1,2} \cdot r_{1,4} \cdot r_{1,3} \cdot r_{2,4} \\ &\quad \cdot r_{3,4} \cdot r_{2,4} \cdot r_{1,2,3} \cdot r_{1,2,4} \cdot r_{1,3,4} \cdot r_{1,2,4} \cdot r_{2,3,4} \cdot r_{1,2,3,4}, \\ P_r(\overline{S_2} \cap \overline{S_3}) &= P_r(\overline{S_2 \cup S_3}) = 1 - s_2 - s_3 + s_{2,3} = r_2 \cdot r_3 \cdot r_{1,2} \cdot r_{1,3} \cdot r_{2,3} \\ &\quad \cdot r_{2,4} \cdot r_{3,4} \cdot r_{1,2,3} \cdot r_{1,2,4} \cdot r_{1,3,4} \cdot r_{2,3,4} \cdot r_{1,2,3,4}, \\ P_r(\overline{S_2} \cap \overline{S_4}) &= P_r(\overline{S_2 \cup S_4}) = 1 - s_2 - s_4 + s_{2,4} = r_2 \cdot r_4 \cdot r_{1,2} \cdot r_{1,4} \cdot r_{2,3} \cdot r_{2,4} \cdot r_{2,4} \\ &\quad \cdot r_{3,4} \cdot r_{1,2,3} \cdot r_{1,2,4} \cdot r_{1,2,4} \cdot r_{1,3,4} \cdot r_{2,3,4} \cdot r_{1,2,3,4}, \\ P_r(\overline{S_3} \cap \overline{S_4}) &= P_r(\overline{S_3 \cup S_4}) = 1 - s_3 - s_4 + s_{3,4} = r_3 \cdot r_4 \cdot r_{1,2} \cdot r_{1,3} \cdot r_{1,4} \cdot r_{2,3} \\ &\quad \cdot r_{2,4} \cdot r_{3,4} \cdot r_{1,2,3} \cdot r_{1,2,4} \cdot r_{1,3,4} \cdot r_{2,3,4} \cdot r_{1,2,3,4}, \\ P_r(\overline{S_1} \cap \overline{S_2} \cap \overline{S_3}) &= P_r(\overline{S_1 \cup S_2 \cup S_3}) = 1 - s_1 - s_2 - s_3 + s_{1,2} + s_{1,3} + s_{2,3} - s_{1,2,3} \\ &= r_1 \cdot r_2 \cdot r_3 \cdot r_{1,2} \cdot r_{1,3} \cdot r_{2,3} \cdot r_{1,4} \cdot r_{2,4} \cdot r_{3,4} \cdot r_{1,2,3} \\ &\quad \cdot r_{1,3,4} \cdot r_{1,2,4} \cdot r_{2,3,4} \cdot r_{1,2,3,4}, \\ P_r(\overline{S_1} \cap \overline{S_2} \cap \overline{S_4}) &= P_r(\overline{S_1 \cup S_2 \cup S_4}) = 1 - s_1 - s_2 - s_4 + s_{1,2} + s_{1,4} + s_{2,4} - s_{1,2,4} \\ &= r_1 \cdot r_2 \cdot r_4 \cdot r_{1,2} \cdot r_{1,4} \cdot r_{1,3} \cdot r_{2,3} \cdot r_{2,4} \cdot r_{1,4} \cdot r_{2,4} \cdot r_{3,4} \cdot r_{1,2,3} \\ &\quad \cdot r_{1,2,4} \cdot r_{1,3,4} \cdot r_{2,3,4} \cdot r_{1,2,3,4}, \end{aligned}$$

$$
\begin{aligned}
P_r(\overline{S_1} \cap \overline{S_3} \cap \overline{S_4}) &= P_r(\overline{S_1 \cup S_3 \cup S_4}) = 1 - s_1 - s_3 - s_4 + s_{1,3} + s_{1,4} + s_{2,4} - s_{1,2,4} \\
&= r_1 \cdot r_3 \cdot r_4 \cdot r_{1,2} \cdot r_{1,3} \cdot r_{1,4} \cdot r_{2,3} \cdot r_{2,4} \cdot r_{3,4} \cdot r_{1,2,3} \cdot r_{1,2,4} \\
&\quad \cdot r_{1,3,4} \cdot r_{2,3,4} \cdot r_{1,2,3,4}, \\
P_r(\overline{S_2} \cap \overline{S_3} \cap \overline{S_4}) &= P_r(\overline{S_2 \cup S_3 \cup S_4}) = 1 - s_2 - s_3 - s_4 + s_{1,2} + s_{2,3} + s_{2,4} - s_{1,2,4} \\
&= r_2 \cdot r_3 \cdot r_4 \cdot r_{1,2} \cdot r_{1,3} \cdot r_{1,4} \cdot r_{2,3} \cdot r_{2,4} \cdot r_{3,4} \cdot r_{1,2,3} \\
&\quad \cdot r_{1,2,4} \cdot r_{1,3,4} \cdot r_{2,3,4} \cdot r_{1,2,3,4}, \\
P_r(\overline{S_1} \cap \overline{S_2} \cap \overline{S_3} \cap \overline{S_4}) &= P_r(\overline{S_1 \cup S_2 \cup S_3 \cup S_4}) = 1 - s_1 - s_2 - s_3 - s_4 + s_{1,2} + s_{1,3} + s_{1,4} \\
&\quad + s_{2,3} + s_{2,4} + s_{3,4} - s_{1,2,3} - s_{1,2,4} - s_{1,3,4} - s_{2,3,4} + s_{1,2,3,4} \\
&= r_1 \cdot r_2 \cdot r_3 \cdot r_4 \cdot r_{1,2} \cdot r_{1,3} \cdot r_{1,4} \cdot r_{2,3} \cdot r_{2,4} \cdot r_{3,4} \cdot r_{1,2,3} \cdot r_{1,2,4} \\
&\quad \cdot r_{1,3,4} \cdot r_{2,3,4} \cdot r_{1,2,3,4}.
\end{aligned}
$$

Now, determine the values of $x_i, x_{i,j}, x_{i,j,k}, x_{i,j,k,l}$:

$$
\begin{aligned}
x_1 &= 1 - \frac{P_r(\overline{S_1} \cap \overline{S_2} \cap \overline{S_3} \cap \overline{S_4})}{P_r(\overline{S_2} \cap \overline{S_3} \cap \overline{S_4})}, \\
x_2 &= 1 - \frac{P_r(\overline{S_1} \cap \overline{S_2} \cap \overline{S_3} \cap \overline{S_4})}{P_r(\overline{S_1} \cap \overline{S_3} \cap \overline{S_4})}, \\
x_3 &= 1 - \frac{P_r(\overline{S_1} \cap \overline{S_2} \cap \overline{S_3} \cap \overline{S_4})}{P_r(\overline{S_1} \cap \overline{S_2} \cap \overline{S_4})}, \\
x_4 &= 1 - \frac{P_r(\overline{S_1} \cap \overline{S_2} \cap \overline{S_3} \cap \overline{S_4})}{P_r(\overline{S_1} \cap \overline{S_2} \cap \overline{S_3})},
\end{aligned}
$$

$$
\begin{aligned}
x_{1,2} &= 1 - \frac{P_r(\overline{S_1} \cap \overline{S_3} \cap \overline{S_4}) \cdot P_r(\overline{S_2} \cap \overline{S_3} \cap \overline{S_4})}{P_r(\overline{S_1} \cap \overline{S_2} \cap \overline{S_3} \cap \overline{S_4}) \cdot P_r(\overline{S_3} \cap \overline{S_4})}, \\
x_{1,3} &= 1 - \frac{P_r(\overline{S_1} \cap \overline{S_2} \cap \overline{S_4}) \cdot P_r(\overline{S_2} \cap \overline{S_3} \cap \overline{S_4})}{P_r(\overline{S_1} \cap \overline{S_2} \cap \overline{S_3} \cap \overline{S_4}) \cdot P_r(\overline{S_2} \cap \overline{S_4})}, \\
x_{1,4} &= 1 - \frac{P_r(\overline{S_1} \cap \overline{S_2} \cap \overline{S_3}) \cdot P_r(\overline{S_2} \cap \overline{S_3} \cap \overline{S_4})}{P_r(\overline{S_1} \cap \overline{S_2} \cap \overline{S_3} \cap \overline{S_4}) \cdot P_r(\overline{S_2} \cap \overline{S_3})}, \\
x_{2,3} &= 1 - \frac{P_r(\overline{S_1} \cap \overline{S_2} \cap \overline{S_4}) \cdot P_r(\overline{S_1} \cap \overline{S_3} \cap \overline{S_4})}{P_r(\overline{S_1} \cap \overline{S_2} \cap \overline{S_3} \cap \overline{S_4}) \cdot P_r(\overline{S_1} \cap \overline{S_4})}, \\
x_{2,4} &= 1 - \frac{P_r(\overline{S_1} \cap \overline{S_2} \cap \overline{S_3}) \cdot P_r(\overline{S_1} \cap \overline{S_3} \cap \overline{S_4})}{P_r(\overline{S_1} \cap \overline{S_2} \cap \overline{S_3} \cap \overline{S_4}) \cdot P_r(\overline{S_1} \cap \overline{S_4})}, \\
x_{3,4} &= 1 - \frac{P_r(\overline{S_1} \cap \overline{S_2} \cap \overline{S_3}) \cdot P_r(\overline{S_1} \cap \overline{S_2} \cap \overline{S_4})}{P_r(\overline{S_1} \cap \overline{S_2} \cap \overline{S_3} \cap \overline{S_4}) \cdot P_r(\overline{S_1} \cap \overline{S_2})},
\end{aligned}
$$

$$
\begin{aligned}
x_{1,2,3} &= 1 - \frac{P_r(\overline{S_1} \cap \overline{S_4}) \cdot P_r(\overline{S_2} \cap \overline{S_4}) \cdot P_r(\overline{S_3} \cap \overline{S_4}) \cdot P_r(\overline{S_1} \cap \overline{S_2} \cap \overline{S_3} \cap \overline{S_4})}{P_r(\overline{S_4}) \cdot (P_r(\overline{S_1} \cap \overline{S_2} \cap \overline{S_4}))(P_r(\overline{S_1} \cap \overline{S_3} \cap \overline{S_4}))(P_r(\overline{S_2} \cap \overline{S_3} \cap \overline{S_4}))}, \\
x_{1,2,4} &= 1 - \frac{P_r(\overline{S_1} \cap \overline{S_3}) \cdot P_r(\overline{S_2} \cap \overline{S_3}) \cdot P_r(\overline{S_3} \cap \overline{S_4}) \cdot P_r(\overline{S_1} \cap \overline{S_2} \cap \overline{S_3} \cap \overline{S_4})}{P_r(\overline{S_3}) \cdot (P_r(\overline{S_1} \cap \overline{S_2} \cap \overline{S_3}))(P_r(\overline{S_1} \cap \overline{S_3} \cap \overline{S_4}))(P_r(\overline{S_2} \cap \overline{S_3} \cap \overline{S_4}))},
\end{aligned}
$$

Table 1. Round functions with key size in encryption process.

Round functions	*Key size and data length after encryption*		
	36	1296	46656
8	4.5	162	5832
16	2.25	81	2916
32	1.125	40.5	1458
64	0.5625	20.25	729

$$x_{1,3,4} = 1 - \frac{P_r(\overline{S_1} \cap \overline{S_2}) \cdot P_r(\overline{S_2} \cap \overline{S_3}) \cdot P_r(\overline{S_2} \cap \overline{S_4}) \cdot P_r(\overline{S_1} \cap \overline{S_2} \cap \overline{S_3} \cap \overline{S_4})}{P_r(\overline{S_2}) \cdot (P_r(\overline{S_1} \cap \overline{S_2} \cap \overline{S_3}))(P_r(\overline{S_1} \cap \overline{S_2} \cap \overline{S_4}))(P_r(\overline{S_1} \cap \overline{S_3} \cap \overline{S_4}))},$$

$$x_{2,3,4} = 1 - \frac{P_r(\overline{S_1} \cap \overline{S_2}) \cdot P_r(\overline{S_1} \cap \overline{S_3}) \cdot P_r(\overline{S_1} \cap \overline{S_4}) \cdot P_r(\overline{S_1} \cap \overline{S_2} \cap \overline{S_3} \cap \overline{S_4})}{P_r(\overline{S_1}) \cdot (P_r(\overline{S_1} \cap \overline{S_2} \cap \overline{S_3}))(P_r(\overline{S_1} \cap \overline{S_2} \cap \overline{S_4}))(P_r(\overline{S_1} \cap \overline{S_3} \cap \overline{S_4}))},$$

$$x_{1,2,3,4} = 1 - \frac{P_r(\overline{S_1}) \cdot P_r(\overline{S_2}) \cdot P_r(\overline{S_3}) \cdot P_r(\overline{S_4}) \cdot P_r(\overline{S_1} \cap \overline{S_2} \cap \overline{S_3}) \cdot}{(P_r(\overline{S_1} \cap \overline{S_2} \cap \overline{S_3} \cap \overline{S_4})) \cdot (P_r(\overline{S_1} \cap \overline{S_2})) \cdot (P_r(\overline{S_1} \cap \overline{S_3}))}$$
$$\cdot \frac{P_r(\overline{S_1} \cap \overline{S_2} \cap \overline{S_4}) \cdot P_r(\overline{S_1} \cap \overline{S_3} \cap \overline{S_4}) \cdot P_r(\overline{S_2} \cap \overline{S_3} \cap \overline{S_4})}{\cdot (P_r(\overline{S_1} \cap \overline{S_4})) \cdot (P_r(\overline{S_2} \cap \overline{S_3})) \cdot (P_r(\overline{S_2} \cap \overline{S_4})) \cdot (P_r(\overline{S_3} \cap \overline{S_4}))}$$

Using the above expressions, all the values of $x_i, x_{i,j}, x_{i,j,k}, x_{i,j,k,l}$ can be evaluated. The cryptographic key pair needs maintenance; in order to keep up with the increasing processing power available for breaking the keys, the keys need to be replaced periodically [5,20]. This will also limit the possibility of damage in a situation where somebody has managed to steal a copy of the secret key. Buffers g and h (Figure 1) were provided to each station to store the incoming data and the key used to re-encrypt the data. The probability of an optimized linear expression was calculated by

$$|P_l - 0.5| \leq 2^{N_S} \cdot |P_o - 0.5|^{N_S}. \tag{10}$$

The size of the key and the data length after the encryption process provide the flexibility to choose the desired round functions as shown in Table 1.

4. Latency and energy consumption

The latency and energy consumption for encryption and decryption processes were evaluated in order to get the information about the time which will be available for the hacker during the channel; if this time is observed to be large, then one has to make more number of attempts to break the model; there is the possibility that during this exercise, the hacker would fail to break it but cause harm to it [20]. Using the experimental results, the smallest value of latency was calculated (Table 2).

Moreover, the efficiency for the 8-bit processors (Motorola 6811) and for AMD 4600 was calculated (Figure 4).

4.1 *Generation of a symmetric key*

If C' is the arrangement of characters in a set and the nth alphabet (A) in a set is $A = \{A_0, A_1, A_2, \ldots, A_{n-1}\}$, then the function is given by $F = \{f(A_0), f(A_1), \ldots, f(A_{n-1})\}$.

One-to-one mapping was done for each character of A w.r.t. C'.

Table 2. Latency in the encryption and decryption process.

Operation			
Encryption			
Key size (bits)	Round functions	Latency (μs)	Energy consumption (μJ)
8	8	22.32	24.54
8	16	34.23	32.55
16	8	15.55	23.43
16	16	21.25	44.45
Decryption			
Key size (bits)	Round functions		
8	8	24.18	23.50
8	16	36.67	27.57
16	8	19.10	19.09
16	16	22.10	42.10

(i)Efficiency

(a) For 8 Bit Processors (Motorola 6811)

Operation	S-box = 4	S-box = 8	S-box = 16	S-box = 32
Encrypt (8 bits)	2100	2453	2562	2955
Decrypt (8 bits)	1443	2155	2310	1055
Key Setup	2	8	256	(32/4)/2 =16
Algorithm Setup	4	4	4	4
Key Change	8	8,64,11 53,2600	8,30,10 90,2345	8,16,10 75,1101

Clock Speed: 2MHz , RAM: 256MB

(b) For AMD 4600.

Operation	S-box = 4	S-box = 8	S-box = 16	S-box = 32
Encrypt (8 bits)	1450	1590	1755	1810
Decrypt (8 bits)	910	975	1010	1200
Key Setup	2	4	64	8
Algorithm Setup	4	4	4	4
Key Change	8	Based upon Algorithm	Based upon Algorithm	Based upon Algorithm

Clock Speed: 2.4GHz. RAM: 1GB

Figure 4. Various factors for different processors.

Encryption:

$$E_k(M) = f(m_0)f(m_1)\cdots f(m_{n-1}). \tag{11}$$

For substitution cipher,

$$N = \frac{\log_2 S'}{D} \cdot d.$$

Using Vernam cipher $C_i = (m_i + k_i) \bmod 2$, Equation (11) becomes $E_k(M) = (m_1 + k_1) \cdot (m_2 + k_2)$.

The $\oplus$ operation with the same key $E_k(M) = M_i$ results in a symmetric key [$\because k_i \oplus k_i = 0$].

For an asymmetric key, different operations are required: (i) X-OR and use of S-Boxes and (ii) generation of keys from the data.

4.2 *Generation of keys from the data*

Let us assume a key set $K = (k_n)$ having a security parameter n (Figure 5).

If x_i is equal to the input data, then y_i is equal to the input data, $1 \le i \le N$, B is the Boolean function, and A_n is the adversary A.

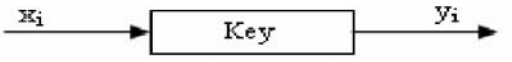

Figure 5. Key generation process.

Then,

$$P_r[B(y_n) = 1 : k \leftarrow K_n, \{y_i \leftarrow A_i(x_i)\}_{1 \le i \le N}] < \frac{1}{p(k)}.$$

For any positive polynomial P and $k \to \infty$,

$$P_i = P_r[B(y_n) = 1 : \{y_i \leftarrow A_i(x_i)\}_{1 \le i \le N}]$$

is the conditional probability of success of A for a fixed i, and we know that

$$P_r\left[\left\{k \in \frac{K_n}{p_i} > \frac{1}{2p(k)}\right\}\right].$$

$$\therefore P_r[B(y_n) = 1 : k \leftarrow K_n, \{y_i \leftarrow A_i(x_i)\}_{1 \le i \le N}] = \sum_{k \in K_n} P_r[k] \cdot P_k$$

$$= \sum_{P_r < 1/2(p(k))} P_r[k] \cdot P_k + \sum_{P_r > 1/2(p(k))} P_r[k] \cdot P_k < \sum_{P_r < 1/2(p(k))} P_r[k] \cdot \frac{1}{2P(k)} + \sum_{P_r > 1/2(p(k))} P_r[k] \cdot 1,$$

$$P_r\left[\left\{k \in \frac{K_n}{p_i} \le \frac{1}{2P(k)}\right\}\right] \cdot \frac{1}{2P(k)} + P_r\left[\left\{i \in \frac{I_n}{P_i} > \frac{1}{2P(k)}\right\}\right] < \frac{1}{2P(k)} + \frac{1}{2P(k)},$$

$$P_r\left[\left\{k \in \frac{K_n}{p_i} \le \frac{1}{2P(k)}\right\}\right] \cdot \frac{1}{2P(k)} + P_r\left[\left\{i \in \frac{I_n}{P_i} > \frac{1}{2P(k)}\right\}\right] < \frac{1}{P(k)}.$$

For M and R positive polynomials and $k \to \infty$,

$$\frac{1}{M(k)R(k)} > P_r[B(y_n) = 1 : k \leftarrow K_n, \{y_i \leftarrow A_i(x_i)\}_{1 \le i \le N}],$$

$$\sum_{k \in K_N} P_r[k] \cdot P_k \ge \sum_{P_i > 1/M(k)} P_r[k] \cdot P_k,$$

$$\sum_{i \in I_N} P_r[i] \cdot P_i \ge \sum_{P_i > 1/R(k)} P_r[i] \cdot P_i.$$

From the above calculations, we can obtain the asymmetrical keys used for the S-Boxes.

5. Cryptographic algorithm used for the generation of keys from the data

The proposed algorithm uses the available data to generate keys and also avoids the need of transmitting additional bits along with the cipher text. It improves the bandwidth and performance of the model, which enhances the data rate. The key generation mechanism was used to know both the parties (sender and receiver), so that the correct combination of keys can be used for retrieving the data. The key generation process also depends upon the input data stream, and Table 3 was used to generate the keys. It also represents the various conditions for the operation

Table 3. Algorithm to generate the keys from the available data with S-Boxes.

Data type	Conditions	Operation performed by S-Boxes	Key length (KL)	Round functions (RFs)
Alphabets	$A > B$ $A = B$ $A < B$	$A = B$ $A - B$ $A \div B$	An 8-bit KL is used if the input data stream is $\leq$ 16 bits; otherwise a 16-bit KL is used	8 RFs are used if the input data stream is $\leq$ 16 bits; otherwise 16 RFs are used
Number	$A > B$ $A = B$ $A < B$	$\bar{A} + B$ $\bar{A} - B$ $\bar{A} \div B$	An 8-bit KL is used if the input data stream is $\leq$ 8 bits; otherwise a 16-bit KL is used	8 RFs are used if the input data stream is $\leq$ 8 bits; otherwise 16 RFs are used
Alphanumeric	$A > B$ $A = B$ $A < B$	$A + \bar{B}$ $A - \bar{B}$ $A \div \bar{B}$	An 8-bit KL is used if the input data stream is $\leq$ 16 bits; otherwise a 16-bit KL is used	8 RFs are used if the input data stream is $\leq$ 16 bits; otherwise 16 RFs are used
Hybrid	$A > B$ $A = B$ $A < B$	$\bar{A} \oplus B$ $A \oplus B$ $A \oplus \bar{B}$	An 8-bit KL is used if the input data stream is $\leq$ 32 bits; otherwise a 16-bit KL is used	8 RFs are used if the input data stream is $\leq$ 32 bits; otherwise 16 RFs are used

to be performed by the S-Boxes in order to design an MN having single or multiple keys of a fixed/variable length. If the key size is small and has a fixed length, then it is discarded due to its poor security response. There is a need to determine the failure rate of all the keys in an MN for secure data transmission. The algorithm checks the input data stream, and further key generation process was used to design the multiple keys using S-Boxes. The input data streams are broadly classified into four categories, namely (i) alphabets, (ii) numeric, (iii) alphanumeric, and (iv) hybrid (includes the combination of alphabets, numbers, and special characters). The input data streams were processed using Figure 1 into numeric values and then further converted into binary strings. This binary data string was used for the random key generation process specified by Table 3. Once the key was generated, the encryption was carried out using the fixed and variable length keys. The variable length key was preferred due to less overheads, and it also provided a more secure model. The padding overheads were very less in this case. The failure rate of a key was checked using mathematical tools; for weak nodes, the re-encryption was done using the second key, which undergoes a procedure the same as that undergone by the first key. The second key is required if there is a node failure or the input data sequence is very large. The key strength is very high in the case of hybrid data sequences because more combinations are available for the generation of keys.

6. Analysis and simulation results for the failure rate of single and multiple keys having variable lengths for each node

Multiple keys were generated from the available data sequences using Table 3. Initially, data were placed in a pool and divided into nearly two sections. Both the sections were compared, and further based upon the conditions, proper operations were performed. For example, if $A > B$ and the data are in the terms of alphabets only, then the $A + B$ operation can be performed in the initial phase and is named as the first round function. The output of this operation is used by the second round operation, which requires another operation for its working. These operations are randomly selected and they provide different outputs even if the pool has the same data for multiple keys. This procedure continues for 8 and 16 iterations depending upon the required security level for a given MN. For $A > B$ and data stream greater than 16 bits, a 16-bit key can be used for the encryption of data. The first key was generated using the following method: assume

that $A = 1000001011010001$ and $B = 0010100000101110$ are the two data streams available in the pool, then the first round function uses the first operation, which is given as follows:

$$\begin{aligned} A &= 1000001011010001, \\ B &= 0010100000101110, \\ \hline C &= 1010101011111111. \end{aligned}$$

C is the output of the first round function, which is further used by the second round function, which is based upon the left shifting of the stream by 1 bit and is given as

$$\begin{aligned} C &= 1010101011111111, \\ D &= 0101010111111110. \end{aligned}$$

The output of the second round function is further used by the next S-Box and the process continues for 8 or 16 iterations. Similarly, the second key was generated by keeping an eye on the length and type of input data.

6.1 *Case 1: when the input data are in the form of Alphabets*

When the data are in terms of only alphabets, then the following steps are used for the generation of keys:

(a) Converting the data (text) into a number using Figure 1.
(b) Checking the conditions based upon the data evaluated using Table 3.
(c) Performing the specified operation mentioned in Table 3.
(d) Converting a number into a binary format.
(e) Using S-Boxes in order to perform round functions for the generation of keys. Checking the next input if it is still in terms of alphabets and then following the same procedure; otherwise, switching to case 2, 3, or 4 as required.

6.2 *Generation of symmetrical and unsymmetrical keys*

This section involves the analysis of symmetrical and unsymmetrical key generation mechanisms from the data streams. In the proposed algorithm, unsymmetrical keys were preferred because it eliminates the key transportation problem. For a small MN, the use of symmetrical keys is also required; that is why the symmetrical keys were been also considered. The nth alphabet A in a set is given as

$$A = \{A_0, A_1, A_2, \ldots A_{n-1}\}.$$

Then, the function for the same can be expressed as

$$F = \{f(A_0), f(A_1), \ldots, f(A_{n-1})\}.$$

One-to-one encryption was done for each character of A w.r.t. the key. As a result, the encryption for the messages was achieved, and it is expressed as

$$E_k(M) = f(m_0)f(m_1)\cdots f(m_{n-1}),$$

where m is the total number of messages.

For a symmetrical key, the valid condition is $f(1) = f(0)$.

The second key function is given as $f'(x) \neq 0$ in $[0,1]$; then $f(1) \neq f(0)$ leads to an asymmetrical key.

Proof

$$\frac{f(1)-f(0)}{1-0} = f'(x),$$

which is not equal to 0:

$$f(1) \neq f(0).$$

■

6.3 *Case 2: when the input data are in the form of Number*

(a) Converting a number into a binary format.
(b) Checking the conditions based upon the data evaluated using Table 3.
(c) Performing the specified operation mentioned in Table 3.
(d) Using S-Boxes in order to perform round functions for the generation of keys.

6.4 *Case 3: when the input data are in the form of Alphanumeric*

It has been observed that all the alphabets have their own frequency, that is, the occurrence of a particular alphabet in the given information is not the same for a given message. If all the common alphabets are processed in one step, then it will reduce the overheads of the system.

For a given message M in a set of $Y_0, Y_1, \ldots, Y_{n-1}$, the probability is defined as

$$\sum_{i=0}^{n-1} P(Y_i) = 1.$$

The conditional probability of message X in a given message Y is $P_Y(X)$, which can also be written as $P(X/Y)$. The joint probability messages X and Y are given as

$$P(X,Y) = P_Y(X)P(Y).$$

The entropy is calculated as

$$H_Y(X) = \sum_{X,Y} P(X,Y)\log_2\left(\frac{1}{P_Y(X)}\right),$$

$$H_Y(X) = -\sum_{X,Y} P(X,Y)\log_2 P_Y(X)$$

or

$$H_Y(X) = \sum_{Y} P(Y)\sum_{X} P_Y(X)\log_2\left(\frac{1}{P_Y(X)}\right),$$

$$H_X(Y) = \sum_{X} P(X)\sum_{Y} P_X(Y)\log_2\left(\frac{1}{P_X(Y)}\right).$$

The entropy of the key is expressed as

$$P_i = \sum_{i=0}^{n-1}\left(M - \frac{1}{n}\right)^2.$$

The probability of the occurrence of an event always lies in the interval $0 \leq P_i \leq 1$. For $n \to \infty$, the chance of getting the exact alphabet reduces to $P_i \cong 0$; it means that an increase in the bits

of the given information in a processing unit always increases the security of the model. It would be preferable to increase the number of bits and the round functions at the transmitter in order to provide an equal number of bits for A and B. If a system has nearly equal length sequences for the data and key, then the padding time is reduced, which results in fast processing. For a system having 26 alphabets and 0–9 numeric digits, the encryption of data takes place with the help of multiple keys designed by the S-Boxes. The analysis for the probability of finding the correct message when the input is in terms of alphanumeric is given as

$$P_i = \sum_{i=0}^{35} \left(M - \frac{1}{36}\right)^2,$$

$$P_i = \sum_{i=0}^{35} (M)^2 - 2/36 \sum_{i=0}^{35} (M)^2 + 36 \left(\frac{1}{36}\right)^2.$$

For identical messages M, the second data are the same as the first data, that is, $\sum_{i=0}^{35} (M)^2 = 1$.

$$\therefore P_i = \sum_{i=0}^{35} (M)^2 - \frac{2}{36} + \frac{1}{36}, \quad P_i = \sum_{i=0}^{35} (M)^2 - 0.084.$$

The above example is applicable if a model contains less number of nodes (2–30, the results were verified on TMS 320 ADP6713). For larger stages, the probability of obtaining the correct message P_{M} is given as

$$P_{\mathrm{M}} = \frac{S(k)}{D} = \frac{\log_2 d!}{D},$$

where $S(k)$ is the number of stages, D is the total data handled by the model, and d is the data of an individual node.

For $D \to \infty$ and large data for a stage $d \to n$, the equation changes to

$$P_{\mathrm{M}} = \frac{\log_2 n!}{D},$$

where n is the number of nodes used in the network.

Using the above expression, one can obtain the desired message that would provide the information about the number of bits used for the encryption process. The probability of obtaining the correct message also depends upon the number of nodes used in the model. It is very clear that if D increases rapidly, then the decryption process takes more time to decrypt the cipher text. The processing time for each node depends upon the number of keys used for the encryption process, which clearly indicates that the probability of obtaining the correct message is indirectly related to the keys. If the number of nodes is increased, then it suggests that the number of keys in a network also increases; therefore, more time would be required to get the correct message. The following steps are used for the encryption of data:

(a) Converting the data (text and number) into a numeric format using Table 1.
(b) Checking the conditions based upon the data evaluated using Table 3.
(c) Performing the specified operation mentioned in Table 3.
(d) Converting a number into a binary format.
(e) Using S-Boxes in order to perform round functions for the generation of keys.

Table 4. Coding for the special character used in the hybrid technique.

!	@	#	$	%	^	&	*
121	122	123	124	125	126	127	128
(	)	{	}	[	]	:	“
211	212	213	214	215	216	217	218
”	;	‘	’	<	>	?	,
321	322	323	324	325	326	327	328
.	/	‘	~	_	+	_	=
411	412	413	414	415	416	417	418

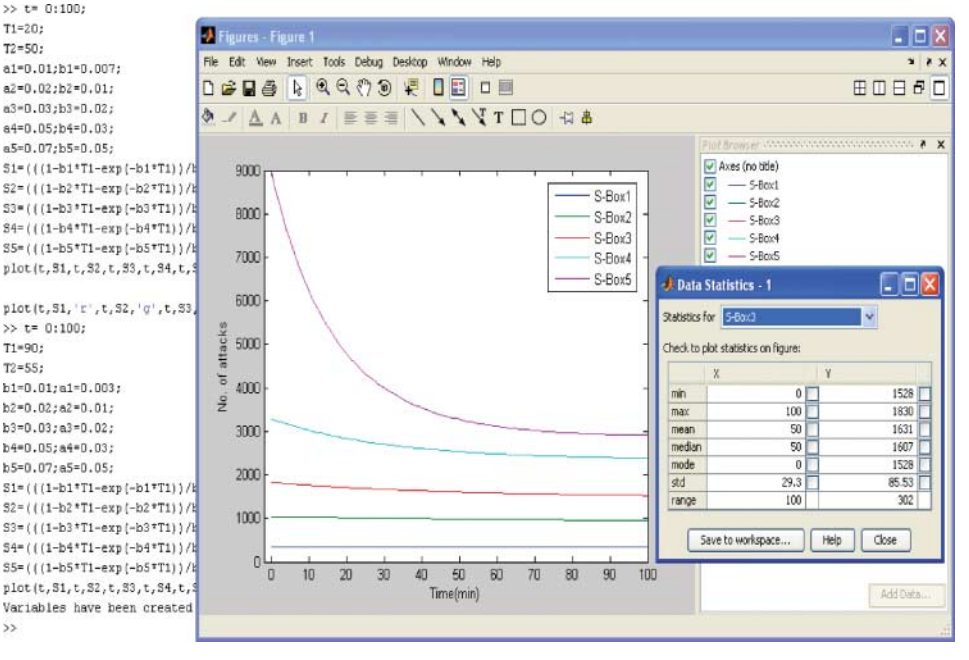

Figure 6. Evaluation of the failure rate of S-Boxes with two keys.

6.5 *Case 4: when the input data are in the form of a hybrid*

For the hybrid structure, the alphanumeric numbers were processed using case 3, and the special symbols were processed using Table 4. The key strength is very high in the case of hybrid structure. It offers more resistance to the hacker, and as a result, the model remains secure for more time.

To protect the data from intruders, powerful encryption algorithms with multiple keys were used. After the encryption process, it is desirable to transmit the cipher text over the channel. The secure model was examined on the basis of its design, mode of transmission of data, and number of nodes. With an increase in the number of nodes, key length, number of keys, and data length, the model consumes more power and takes more time to generate keys from the available data. A new approach in which keys are generated and processed in the cryptographic model with the help of S-Boxes in order to reduce the processing time has been proposed. MATLAB 7.3 was used to determine the failure rate of various keys in an MN (Figure 6).

The model is designed in such a way that it comprises multiple keys and S-Boxes and enables the higher classes to retrieve the encrypted data related to the lower classes. The lower classes do not have the power to access the data concerned with the higher classes. A key management scheme was used to provide such kind of facility to the higher classes. Once a key is exchanged,

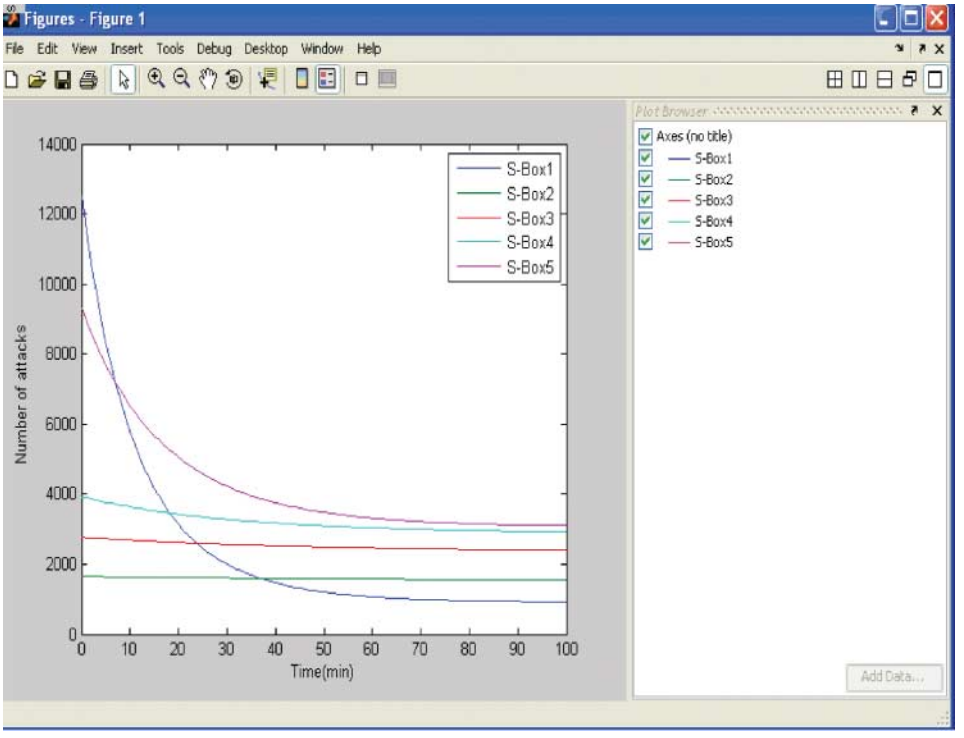

Figure 7. Determination of the failure rate of S-Boxes with three keys.

the bit string of the key becomes known to the receiver. In such cases, it is highly desirable to re-encrypt the same data with the replaced key. This has been adopted only in cases where the failure rate of the previous key exceeds a predefined value. The behaviour of the keys is unpredictable in a real-time environment; there is always a difference between the characteristics of ideal keys and those of the real keys. In order to achieve a secure model, one should determine the extent to which several security patterns are robust to the known categories of attacks. Various classes were created to represent the number of attacks in a given interval of time. Figure 7 shows that if multiple keys are used, a number of attacks (varies from 200 to 9000) are not able to break the system even after 100 min. A total of five S-Boxes were used to design the key using round functions. The S-Boxes result in variable key lengths used for the encryption of data in five stages. This work is focused on the determination of the failure rate of both types of keys (single and multiple) for each node in a specified interval (90 min for the first key and 55 min for the second key).

The first key was used to encrypt the data, and it provided only 20 min for the hacker to make the attacks. After 20 min, the first key was replaced by the second key, which remained active for 50 min. This combination can handle 1250 attacks in 100 min without collapsing. Similarly, one can calculate the failure rate of the other S-Boxes. The following observations have been drawn from Figures 6 and 7:

(a) The strength of the keys decreases with time; therefore, the keys are used to encrypt the data in a short interval. For the same parameters, if the number of keys is increased, a better secure model is achieved.
(b) S-Boxes are used to design the keys using round functions. The encryption of data with multiple keys always provides a better security level than that of data with a single key.
(c) Variable key lengths make the hacking process tougher and cause congestion and complexity.

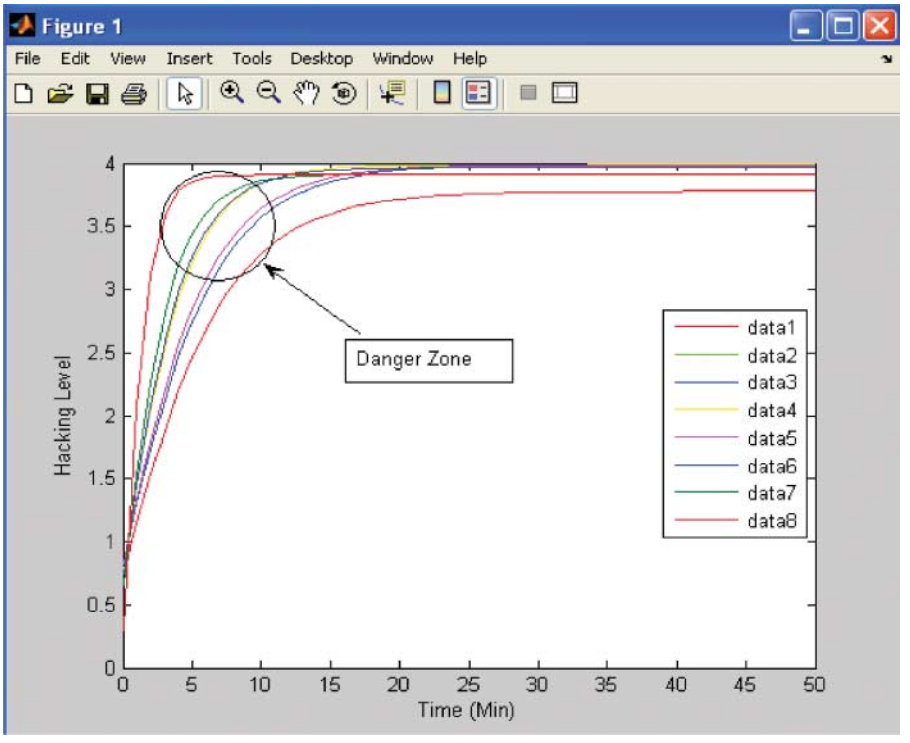

Figure 8. Single key having a variable length designed by eight S-Boxes.

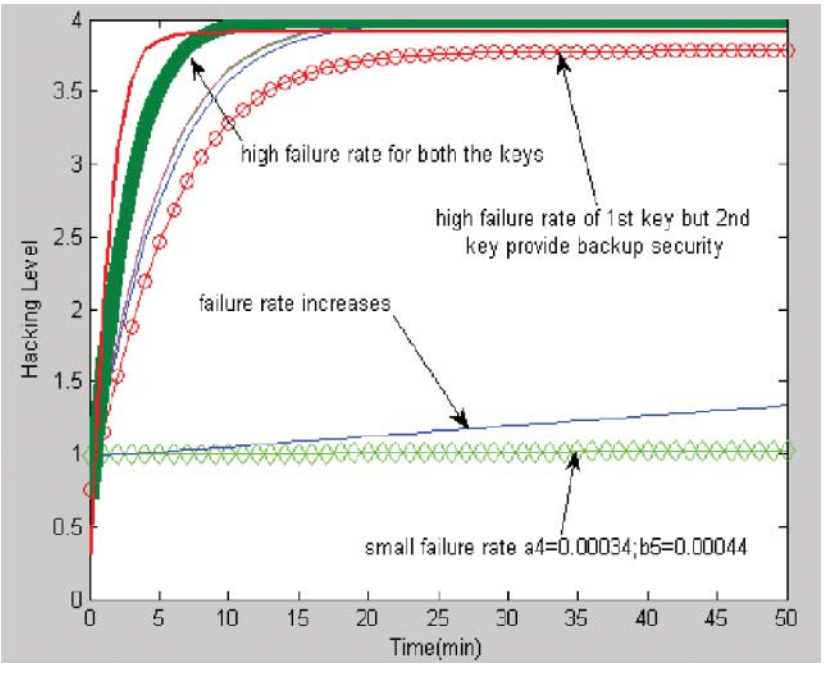

Figure 9. Multiple keys having variable failure rates.

The key shifting time was reduced by starting the generation procedure of the second key whenever the failure rate of the first key increased from 34%. This work includes the response of the nodes having encrypted data with multiple keys. Multiple keys were used for the encryption of data having the key shifting time (0.01 ns) and a better response was achieved and this is shown in Figure 8.

Eight S-Boxes were used to design a single key of a variable length for the encryption of 8, 16, 32, 64, 128, 256, 512, and 1024 data sequences. Figure 8 shows that the response of multiple nodes having encrypted data with a single key (8-bit key length) is not acceptable from a security point of view. The failure rate of the key is fixed due to its length and slightly varies in accordance with the data streams. For higher data streams such as 1024 bits, the security level of the node is much poorer than the security level achieved when an 8-bit key is used to encrypt the 8-bit data.

For the same parameters, if keys having a low failure rate are used, then the response of the model for a particular node (a4 and b5 in Figure 9) falls below the danger level. Whenever the failure rate of the second key is more than that of the first key, the system reliability tends to decrease. Recovery mechanisms are required to get a smooth response; conversely, if the first key fails, then it does not affect the model much because the second key is used to encrypt the data in that situation.

7. Conclusion and future work

Secure and timely transmission of data is always an important aspect for an organization. An efficient encryption algorithm should consist of two factors: (i) fast response and (ii) reduced complexity. Key selection techniques and analysis for security provision of an MN were used in this study. The failure rate of multiple keys was calculated by considering the multiple failures in the model, and it has been analytically shown in the paper. The security also increases if the key size is increased and the key shifting time (δ) is reduced; the above combination may be adopted for secure transmission. This work can be extended if more number of S-Boxes (64 and 128) are used for the same task, and the key length would be reduced with nominal processing time.

References

[1] W. Aiello and R. Venkatesan, *Foiling birthday attacks in length-doubling transformations,* in *Advances in Cryptology – EUROCRYPT* '96, U. Maurer, ed., Lecture Notes in Computer Science Vol. 1070, Springer-Verlag, Berlin, 1996, pp. 307–320.

[2] M. Backes and B. Pfitzmann, *Relating symbolic and cryptographic secrecy*, IEEE Trans. Dependable Secure Comput. 2(2) (2005), pp. 109–123.

[3] A. Banerjee, L. Drake, L. Lang, B. Turner, D. Awduche, L. Berger, K. Kompella, and Y. Rekhter, *Generalized multiprotocol label switching: An overview of signaling enhancements and recovery techniques*, IEEE Commun. Mag. 39(7) (2001), pp. 144–151.

[4] E. Bertino, N. Shang, and S.S. Wagstaff Jr., *An efficient time-bound hierarchical key management scheme for secure broadcasting*, IEEE Trans. Dependable Secure Comput. 5(3) (2008), pp. 65–70.

[5] I.F. Blake and V. Kolesnikov, *Strong conditional oblivious transfer and computing on intervals*, Proceedings of Advances in Cryptology – ASIACRYPT 2004, 10th International Conference on the Theory and Application of Cryptology and Information Security, Jeju Island, Korea, December 5–9, 2004, Lecture Notes in Computer Science Vol. 3329, Springer-Verlag, Berlin, Heidelberg, Germany, 2004, pp. 515–529.

[6] I.F. Blake and V. Kolesnikov, *Conditional encrypted mapping and comparing encrypted numbers*, Proceedings of the 10th International conference on Financial Cryptography and Data Security, Springer-Verlag, Berlin, Heidelberg, 2006, pp. 206–220.

[7] A. Bobbio and K.S. Trivedi, *Computing cumulative measures of stiff Markov chains using aggregation*, IEEE Trans. Comput. 39(10) (1990), pp. 1291–1297.

[8] H. Chan, A. Perrig, and D. Song, *Random key predistribution schemes for sensor networks*, Proceedings of IEEE Symposium on Security and Privacy (S & P '03), IEEE Computer Society, Washington, DC, USA, 2003, pp. 197–213.

[9] G. Ciardo, R. Marmorstein and R. Siminiceanu, *Saturation unbound*, Proceedings of the 9th international conference on Tools and algorithms for the construction and analysis of systems, Springer-Verlag, Berlin, Heidelberg, 2003, pp. 379–393.

[10] W. Du, J. Deng, Y. S. Han, P. K. Varshney, J. Katz and A. Khalili, *A pairwise key predistribution scheme for wireless sensor networks*, Proceedings of the 10th ACM conference on Computer and communications security (CCS'03), ACM, New York, NY, USA, 2003, pp. 42–51.
[11] L. Eschenauer, V. D. Gligor, *A key-management scheme for distributed sensor networks*, Proceedings of the 9th ACM conference on Computer and communications security (CCS'02), ACM, New York, NY, USA, 2002, pp. 41–47.
[12] M. Fischlin, *A Cost-Effective Pay-Per-Multiplication Comparison Method for Millionaires*, Proceedings of the 2001 Conference on Topics in Cryptology: The Cryptographer's Track at RSA, Springer-Verlag, London, UK, 2001, pp. 457–472.
[13] Z. Fu, H. Luo, P. Zerfos, S. Lu, and L. Zhang, *The impact of Multihop wireless channel on TCP performance*, IEEE Trans. Mobile Comput. 4(2) (2005), pp. 209–221.
[14] S.T. Halkidis, N. Tsantalis, A. Chatzigeorgiou, and G. Stephanides, *Architectural risk analysis of software systems based on security patterns*, IEEE Trans. Dependable Secure Comput. 5(3) (2008), pp. 129–142.
[15] X. He, M. Zhang, and Q.K. Yang, *SPEK: A storage performance evaluation kernel module for block-level storage systems under faulty conditions*, IEEE Trans. Dependable Secure Comput. 2(2) (2005), pp. 138–149.
[16] L. Hundessa, *Optimal and guaranteed alternative LSP for multiple failures*, Proceedings of 13th International Conference on Computer Communications and Networks, IEEE Conference, Illinois, Chicago, 2004, pp. 59–64.
[17] S.K. Lee, C. Kim, and D. Griffith, *Hierarchical Restoration Scheme for Multiple Failures in GMPLS Networks*, Proceedings of International Conference on Parallel Processing Workshops (ICPPW'02), IEEE Computer Society, Vancouver, BC, Canada, 2002, pp. 177–182.
[18] D. Liu and P. Ning, *Establishing pairwise keys in distributed sensor networks*, Proceedings of the 10th ACM conference on Computer and communications security (CCS'03), ACM, New York, NY, USA, 2003, pp. 52–61.
[19] V.B. Livshits and M.S. Lam, *Finding security vulnerabilities in java applications with static analysis*, Proceedings of the 14th conference on USENIX Security Symposium (SSYM'05), USENIX Association, Berkeley, CA, USA, 2005, pp. 19–36.
[20] B.B. Madan, K. Goseva-Popstojanova, K. Vaidyanathan, and K.S. Trivedi, *A method for modeling and quantifying the security attributes of intrusion tolerant systems*, Perform. Eval. 56(1) (2004), pp. 167–186.
[21] J. Muppala, M. Malhotra, and K. Trivedi, *Stiffness-tolerant methods for transient analysis of stiff Markov chains*, Microelectronics Reliab. 34(11) (1994), pp. 1825–1841.
[22] M. Naor, B. Pinkas, and R. Sumner, *Privacy preserving auctions and mechanism design, EC'99*, ACM Press, New York, 1999, pp. 129–139.
[23] P. Paillier, *Public-key cryptosystems based on composite degree residuosity classes*, Proceedings of the 17th International conference on Theory and application of cryptographic techniques, Springer-Verlag, Berlin, Heidelberg, 1999, pp. 223–238.
[24] P. Papadimitratos and Z.J. Haas, *Secure message transmission in mobile ad hoc networks*, Ad Hoc Networks 1(1) (2003), pp. 193–209.
[25] J.T. Park, J.W. Nah, and W.H. Lee, *Dynamic path management with resilience constraints under multiple link failures in MPLS/GMPLS networks*, IEEE Trans. Dependable Secure Comput. 5(3) (2008), pp. 143–154.
[26] A. Reibman and K.S. Trivedi, *Numerical transient analysis of Markov models*, Comput. Oper. Res. 15(1) (1988), pp. 19–36.
[27] J. Ren and L. Harn, *Generalized ring signatures*, IEEE Trans. Dependable Secure Comput. 5(3) (2008), pp. 153–164.