Journal of Engineering Science and Technology
Vol. 7, No. 4 (2012) 505 - 516
© School of Engineering, Taylor's University

SECURE COMMUNICATION BY USING MULTIPLE KEYS HAVING VARIABLE LENGTH IN A REAL TIME ENVIRONMENT FOR MULTIPLE STATIONS

AJAY KAKKAR[1,]*, M. L. SINGH[2], P. K. BANSAL[3]

[1]Thapar University, Patiala, Punjab, India
[2]ECE Department, Guru Nanak Dev Univeristy, Amritsar, Punjab, India
[3]MIMIT, Malout, Punjab, India
*Corresponding Author: kakkar_ajay29@rediffmail.com

Abstract

The number of security attacks against network is increasing dramatically with time. To protect the model against these attacks; techniques like passwords, cryptography, and biometrics are used. Even multiple passwords are not preferred due to low entropies. The biometric is a very costly technique; moreover they may produce harmful effects to the body. Cryptography is used to avoid unauthorized access of data. It includes quite simple encryption algorithm and keys generated from the data. This paper deals with the parameters such as number of users, failure rate of various keys, recovery mechanism, encryption, transmission, decryption and latency time, etc.; required for secure transmission of data. The proposed algorithm shows that all the stages are mutually independent from each other. It also provides the facility to use compression and re-encryption techniques.

Keywords: Encryption, S-Boxes, Padding, Keys, Failure rate.

1. Introduction

Security of data is the prime requirement of all the organization. It can be achieved by using various techniques such as passwords, cryptography, biometrics and many more, but out of all these, cryptography with good encryption algorithm having multiple keys provides reliable model. Hacking is done on the internet by creating the spy-ware; which is sent to the victim's machine. User cannot realize it because after opening it, it will be deleted automatically. Then after this moment as soon as user fills his/her user name and password, it will be sent to the attacker. At this point it is possible that the attacker can disable the victim's machine.

Nomenclatures

B	Bandwidth of the channel
$b_k(t)$	Data signal of the k^{th} user
C	Capacity of the channel
C_F	Compression factor
D_t	Decryption time
E_r	Re-Encryption
E_t	Encryption time
E'_1, E'_2	First and second level encryption
F_i	Number of failures during the installation
f_c	Centre frequency
L_t	Latency time
R	Data rate of bit (=1/T_b)
rt, $(r+1)T$	Rectangular pulse in the interval of T
S	Signal power
$S_1, S_2, ..., S_n$	Stations
S_{ti}	Synchronization time
T_b	Time required to transmit 1 bit
T_t	Transmission time
$U_1, U_2, .., U_n$	Users

The normal window XP is easily hacked by using server software's without changing/obtaining the window password. Normally we observe that the system is locked by the administrator and one should require username and password in order to access it. There are various ways to achieve the security in data communication such as; passwords, multiple passwords, cryptography and biometrics. These techniques are used to keep the data secured from the hacker. Passwords are not treated as reliable for this task. It is easy to guess passwords due to its short range. Cryptography is a technique used to protect the data from unauthorized access. It is the best method of saving the documents from the competitors in business. It consists of two components Encryption algorithm and Keys. There are many cryptographic algorithms available in the market for encrypting the data. Encryption is the process in which original data (plaintext) is converted into the encoded format (cipher text) with the help of key. The user's machine can be hacked by two methods:

Method 1: by pressing *Ctrl+Alt+Del* twice, it is observed that new operating system (created during the installation process) is available for the user having username "administrator" and blank "password". By pressing *Enter* you are able to do any kind of task by using that machine.

Method 2: if new operating system does not have blank "password" then restart the machine and press *F*8, go for "safe mode command prompt" type "net user administrator". A new screen is available where you can put your new password or you may leave it blank.

To protect the machine from these attacks encryption is the best method. The ultimate aim of cryptography is to provide the security even if the hacker is able to access the data. Data security is related with both, physical and wireless security. The main motive of the hacker's is to destroy the wireless security, means to generate the virtual attacks because all the systems are based upon the wireless that provides the freedom of mobility to the users. Moreover it is not accepted that the information available on the web always of high quality. Techniques and organizational solutions are required to access and attest the quality of data. They should provide more effective integrity semantics verification and the use of tools for the assessment of data quality [1,2]. Real operations require immediate response from the device when called by the main station.

The factors should be kept in my mind while designing a network are: (i) Users (U_1, U_2, .., U_n), (ii) Type of hardware/software, (iii) Channel Capacity, C, (iv) Number of failures during the installation, F_i, and transmission (a, b, c, …), Number of check points, (v) Encryption, transmission, decryption, latency time (E_t, T_t, D_t, L_t), (vi) Level of protection; first and second level encryption, E_1', E_2' and re-encryption, E_r, (vii) Active stations with reliable key(s), (viii) synchronization time, S_{ti}, (ix) Compression factor, C_F, and, (x) Padding.

The objectives of this work are:

- To achieve security with minimum number of overheads. For this the encryption algorithm is quite simple and to achieve high degree of security; multiple keys are used generated from the data.
- It is clear that keys play an important role in encryption process, which is a main part of cryptography. If the key is slightly changed then almost all the data is worthless.
- Sometimes in transmitting the data, the length of data does not match with the bit length offered by the encryption process. Therefore padding techniques are used to sort out the problem.
- Key should be changed with respect to time because security is not a product, it is a process. With the increase of time the chances of model breakage increases. Therefore the failure time of each station should be determined. If any station goes down then recovery mechanism runs (in that encryption is done by using other key, process called re-encryption). Again failure rate is calculated if station passes the test then the data is send to the next station, otherwise particular station will be cut-off from the model.

2. Proposed Model

System-wide security is always required to make sure that data is safe. By using the information about timing, power consumption, and radiation of a device during the execution of a cryptographic algorithm; cryptanalysts have been able to break the system.

So a motive should be to make the combination secure. Practical model consists of transmitter(s) and other processing units capable to encrypt which is further transmitted over the channel (guided or unguided). The data rate is dependent on the bandwidth further affected by the noise and signal strength. Let consider a model having User A, multiple intermediate stations $S_1, S_2....S_n$ and a receiver (User B) as shown in Fig. 1.

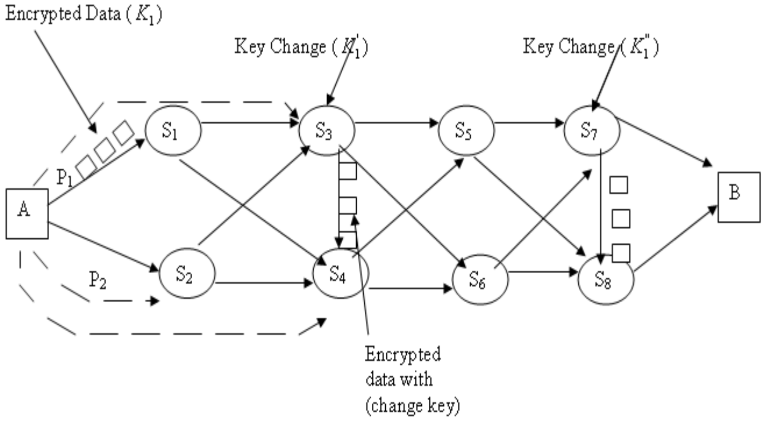

Fig. 1. Multi-Node Network Having Various Nodes Encrypted with Variable Keys.

If K users transmitting the data (binary) simultaneously in a common band with a centre frequency f_c, then the transmitted signal from the K^{th} user is:

$$s_k(t) = \sqrt{2P}b_k(t)a_k(t)\cos(2\pi f_c t + \theta_k) \tag{1}$$

where P is the total power available to each user, $b_k(t)$ is the data signal of the k^{th} user, which consists of a rectangular pulse in the interval of (rt, (r+1)T). The n^{th} bit, $b_{k,n}$ in the K^{th} user is assumed to be ±1 with equal probability and independent of all the bits, $b_{i,j}$. Then we can write the data signal for the K^{th} user:

$$b_k(t) = \sum_{r=\infty}^{\infty} b_{k,r}\, p(t - rt) \tag{2}$$

$$\begin{aligned} p(t) &= 1, \quad 0 \le t \le T \\ &= 0 \quad otherwise \end{aligned}$$

The signal a_k is the periodic coded rectangular pulse having T second duration. It is used to spread to the data signal b_k. The receiver for K^{th} user of station S_1 receives the signal

$$r_k(t) = \sqrt{2P}\sum_{q=1}^{L}\sum_{k=1}^{M}\beta_{kq}(t - \tau_{kq})a_{kq}(t - \tau_{kq}) \tag{3}$$

$$b_{kq}(t - \tau_{kq})\cos(2\pi f_c t + \phi_{kq}) + n_k(t) \tag{4}$$

where θ_k is the carrier phase uniformly spread over random variable (0, 2π).

In this paper a number of parameters are considered.

2.1. Number of users

It is an important parameter; system is designed on the basis of user. All the equipments and the accessories are installed by keeping an eye on the total number of users. Normally, a rough estimate is taken about the user, after that it is taken care that that grade of service not fall below then 73% [1,2]. If model contains n number of stations; each capable of supporting x number of users then the total number of users in a model is calculated as $U_{TN} = n \times x$. The efficiency of the system is degraded by the faulty user. They simply waste the bandwidth and create an overburden on the other station (retransmission of data, re-encryption, generation of key(s) [3, 4].

2.2. Type of Software/Hardware

It provides the facility to the users to select/generate their own key(s). It is also used to determine the speed of the model. The length of a key, in bits, for a conventional encryption algorithm is a common measure of security. To attack an algorithm with n-bit key it will generally require roughly $2n$-1 operations. Hence, to secure a public key system one would generally want to use parameters that require at least $2n$-1 operations to attack [5]. Table 1 gives the key sizes used in the proposed algorithm which is the combination of DES and the Round Function based on contains ECB (Electronic Code Book), CBC, with Clock Speed: 2.4GHz, RAM: 1GB.

Table 1. Comparison of Performance of Measuring Parameters at Various Stations.

Operation	Stations 4	Stations 8	Stations 16	Stations 32
Encrypt Time (s) 8 bits	1450	1590	1755	1810
Decrypt Time (s) 8 bits	910	975	1010	1200
Key Setup Time (s)	8	4	64	8
Algorithm Setup Time (s)	8	8	8	8
Key Change	8	Based upon failure rate	Based upon failure rate	Based upon failure rate
Run Time (s)	0.0002	0.7	24	721

2.3. Channel Capacity

It is the capacity of the channel that how much data can be handled by the channel. The signal to noise ratio is an important parameter in the transmission of digital data. Channel capacity is determined by using Shannon's theorem $C = B\log_2(1+S/N)$ [6]. In ideal case the channel bandwidth $C = B$. In order to increase the data rate either signal strength or bandwidth is increased. When signal strength is increased it leads to nonlinearities (inter-modulation noise) which make the system unstable [7]. On the other hand increase in the bandwidth decreases the signal to noise ratio [8, 9]. We focused on a new parameter, ratio of energy per bit to noise power density/hertz. Energy/bit in a signal are $E_b = ST_b$. Where C and B are capacity and bandwidth of the channel respectively, S is the signal power, T_b is the time required to transmit 1 bit. Then the data rate can written as R=1/T_b. Therefore Bit Error Rate (BER) with respect to E_b/N_o can be determined in order to know the performance of the model. For Channel = AWGN, Modulation order 2, 8, 16, for Modulation type = DPSK (Fig. 2) and for Modulation type = PSK (Fig. 3).

The results have been obtained for 0 and 90% confidence level. From Figs. 2 and 3 it has been observed that it is worthy to use PSK with modulation order 8 for a reliable transmission.

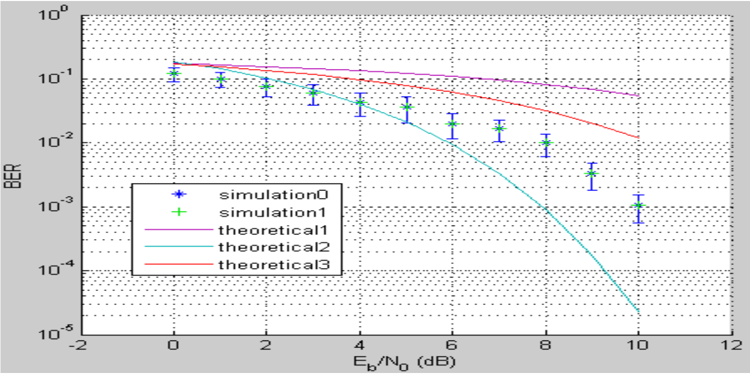

Fig. 2. Performance of Proposed Model in AWGN Channel with DPSK.

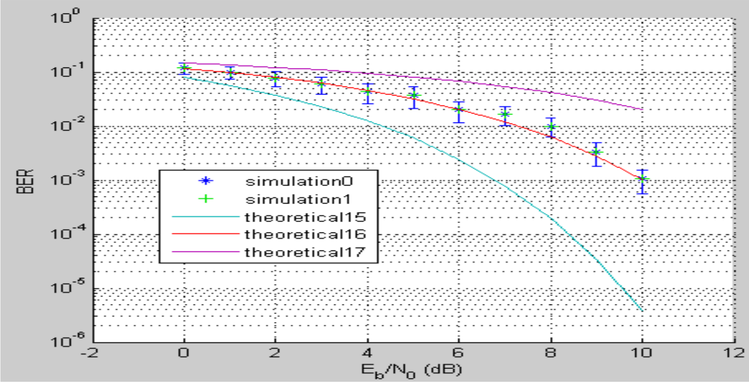

Fig. 3. Performance of Proposed Model in AWGN Channel with PSK.

2.4. Failures occurred during the installation and transmission

The failures occurred during the installation and transmission needs to be evaluated for a secured model. During the installing operation it is observed that 10-35% machines fails, and after running the recovery mechanism it can be reduced to approximately 20%.The failure of station is time dependent process [5, 10-12].

For a model having n number of identical stations the probability of failure rate of stations can be determined as

$$s_i = P(S_i) = (1/T)\int_0^T (1 - e^{-at})dt \tag{1}$$

where $s_i \approx 0.5$aT.

The result holds good when single station fails. For a huge model containing large number of nodes the quantity AT is too less than unity. Therefore in that case failure of anyone station can be ignored and the data is passed over the other path. On the other hand for the multiple failures the joint probabilities are to be determined.

Case: 1 Analysis of single key having same failure rate

The case of same failure rate, *a*, key, k_i, and identical stations, S_i, where $1 \leq i \leq N$, has been used to encrypt the data for the various nodes present in the model. Each node encrypts the data with single key (different for node 1, 2, ... , *N*) but having same bit length.

$$k_i = P(K_i) = (1/T)\int_0^T (1 - e^{-at})dt \tag{2}$$

$$k_i = P(K_i) = (1 - 1/T)\int_0^T (1 - e^{-at})dt = \frac{aT + e^{-aT} - 1}{aT} \tag{3}$$

By using the expansion of $e^{-x} = 1 - x + \frac{x^2}{2!} - \frac{x^3}{3!} + ...$, let $x = aT$ and from Eqs. (2) and (3) we get:

$$\frac{aT + e^{-aT} - 1}{aT} = \frac{aT + \left\{1 - aT + \frac{(aT)^2}{2!} - \frac{(aT)^3}{3!} + ...\right\} - 1}{aT}$$

$$= \frac{1}{aT}\left\{\frac{(aT)^2}{2!} - \frac{(aT)^3}{3!} + ...\right\} = \frac{(aT)}{2\times 1} - \frac{(aT)^2}{3\times 2\times 1} + ... \tag{4}$$

So, in the above expression we neglect the higher terms, i.e., $= \frac{(aT)^2}{3\times 2\times 1} + ...$ we get $\frac{aT}{2\times 1}$, which is approximately equals to $0.5aT$.

Case: 2 Analysis of single key having different failure rate

For different failure rates, (*a* and *b*) keys, k_i, where $1 \leq i \leq N$, has been used to encrypt the data for the various nodes present in the model. Each node encrypts the data with single key having different key lengths.

$$k_i = P(K_i) = (1/T)\cdot\left[\int_0^{T_1} (1 - e^{-at})(1 - e^{-bt})dt\right] = \left\{\frac{\left(1 - bT_1 - e^{-bT_1}\right)}{b}\right\} + (2 - e^{-a(t+T_1)}) \tag{5}$$

Case: 3 Analysis of multiple (two) keys having same failure rate

For different failure rates, (*a* and *b*) keys, k_i and k_j, where $1 \leq i \leq N$, have been used to encrypt the data for the various nodes present in the model. Each node encrypts the data with keys having same key length.

$$k_{i,j} = P(K_i \cap K_j) = (1/T)\cdot\int_0^T (1 - e^{-at}).(1 - e^{-at})dt \tag{6}$$

$$k_i = P(K_i \cap K_j) = 1 - \frac{2}{t}\int_0^T 1 - e^{at}dt + \frac{1}{T}\int_0^T e^{-2at}dt = 1 + \frac{2}{aT}\left(e^{aT} - 1\right) - \frac{1}{2aT}\left(e^{-2aT} - 1\right)$$

$$k_i = \frac{2aT + 4e^{-aT} - e^{-2aT} - 3}{2aT} \tag{7}$$

Again by using the expansion of e^{-x}, it can be shown that:

$$k_i \approx \frac{(aT)^3}{3aT} \tag{8}$$

which is approximately equals to $0.33(aT)^2$.

Case: 4 Analysis of multiple key having different failure rate

For different failure rates, (a and b) keys, k_i and k_j, where $1 \le i \le N$, have been used to encrypt the data for the various nodes present in the model. Each node encrypts the data with keys having different key length.

$$k_{i,j}, k_{i,j}^{'} = P(K_{i,j} \cap K_{i,j}^{'}) = (1/T) \cdot \left[\int_0^{T_1} (1 - e^{-at})(1 - e^{-bt}) dt \cdot \int_0^{T_2} (1 - e^{-at})(1 - e^{-bt}) dt \right] \tag{9}$$

By solving the first half part of Eq. (5), i.e., $= (1/T) \cdot \left[\int_0^{T_1} (1 - e^{-at})(1 - e^{-bt}) dt \right]$

This has been done because the in the reaming part all the terms are same, only the limits are changed, i.e., T_1 to T_2 so, the calculation for the first part is given below:

$$= (1 - e^{-at}) \int_0^{T_1} (1 - e^{-bt}) dt - \int_0^{T_1} \left\{ \frac{d(1 - e^{-at})}{dt} \cdot \int_0^{T_1} (1 - e^{-bt}).dt \right\} dt$$

$$= (1 - e^{-at}) \left\{ \left| t - \frac{e^{-bt}}{-b} \right|_0^{T_1} \right\} - \int_0^{T_1} \left\{ ae^{-at} \cdot \left| t - \frac{e^{-bt}}{-b} \right|_0^{T_1} \right\} dt$$

$$= (1 - e^{-at}) \left\{ \frac{bT_1 + e^{-bT_1}}{b} - \frac{1}{b} \right\} - \int_0^{T_1} \left\{ ae^{-at} \cdot \left(\frac{bT_1 + e^{-bT_1}}{b} - \frac{1}{b} \right) \right\} dt$$

$$= (1 - e^{-at}) \left\{ \frac{\left(1 - bT_1 - e^{-bT_1}\right)}{b} \right\} + \left[\frac{a}{b} \int_0^{T_1} \left\{ e^{-at} \cdot \left(1 - bT_1 - e^{-bT_1}\right) \right\} dt \right]$$

$$= (1 - e^{-at}) \left\{ \frac{\left(1 - bT_1 - e^{-bT_1}\right)}{b} \right\} + \left[\frac{\left(1 - bT_1 - e^{-bT_1}\right)}{b} \cdot (1 - e^{-aT_1}) \right]$$

$$= \left\{ \frac{\left(1 - bT_1 - e^{-bT_1}\right)}{b} \right\} + (2 - e^{-a(t + T_1)}) \tag{10}$$

At this point similarly one can determine the second term and given as:

$$= \left\{ \frac{\left(1 - bT_2 - e^{-bT_2}\right)}{b} \right\} + (2 - e^{-a(t + T_2)}) \tag{11}$$

Using Eqs. (9), (10) and (11) we can write:

$$k_{i,j}, k_{i,j}^{'} = \left\{ \frac{\left(1 - bT_1 - e^{-bT_1}\right)}{b} + (2 - e^{-a(t+T_1)}) \right\} \cdot \left\{ \frac{\left(1 - bT_2 - e^{-bT_2}\right)}{b} + (2 - e^{-a(t+T_2)}) \right\} \qquad (12)$$

By using Eq. (12) the response of a station having different failure rates for variable time intervals ($0 \le t \le 100$) is shown in Table 2.

Table 2. Failure Rate and Encryption Time for Various Stations.

Stations	a	B	T_1	T_2
$S_{1,1}, S_{1,1}^{'}$	0.1	0.2	1	5
$S_{1,2}, S_{1,2}^{'}$	0.3	0.2	1.1	5.3
$S_{1,3}, S_{1,3}^{'}$	0.3	0.5	0.7	2.7
$S_{1,4}, S_{1,4}^{'}$	0.4	0.4	1.2	0.8
$S_{1,5}, S_{1,5}^{'}$	0.5	0.1	4	2.8

3. Algorithm

The algorithm checks the input data stream and then further key generation process has been used to design the multiple keys by using S- Boxes. The input data streams are broadly classified into four categories named as; (i) alphabets and (ii) numeric data. This binary data string has been used for the random key generation process specified by Table 3. Once the key has been generated then the encryption is carried by using the fixed and variable length key. The variable length key has been preferred due to less overheads and it also provide more secured model. The padding overheads are very small in this case. The failure rate of key has been checked by using mathematical tools; if key strength is weak then the re-encryption has been done by using 2nd key which has been generated by using the same procedure as used for 1st key. The requirement of 2nd key is required if there is a node failure or input data sequence is very large. The key strength is very high in case of hybrid data sequences because more combinations are available for the keys generation.

Table 3. Specifications of Random Key generation Process.

Data Type	Conditions	Operation performed by S-Boxes	Round Functions (RF)
Alphabets	$A > B$ $A = B$ $A < B$	$A + B$ $A - B$ $A \div B$	8 RF are used if input data stream is ≤ 16 bits, otherwise 16 RF are used
Number	$A > B$	$\overline{A} + B$	8 RF are used if input data stream is ≤ 8 bits, otherwise 16 RF are used

The encryption time depends upon the number and type of operations used in S-boxes and key(s). The Round Keys are derived from the Cipher Key by means of the key schedule. This consists of two components: the Key Expansion and the Round Key Selection. The total number of Round Key bits is equal to the block length multiplied by the number of rounds plus 1. Encryption and Decryption time is determined by using equation (14). Whereas transmission is based on the channel; latency time $l_{Si} = \left[\frac{2lK}{M-2n} - \sum_{i=1}^{n-1} l_i \right]$, where, l_{Si} is the smallest value of latency and M is the total number of messages.

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 to compare 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. For example, let’s take a system having a key of length 1bit then it is sure that the hacker is able to break the system after 35 attempts (maximum), technique is called First level encryption $E_1^{'}$. If the key length is increased then hacker has to made more attempts for the task, technique known as Second level encryption $E_2^{'}$. For highly secure model Re-Encryption E_r is used where multiple keys are used by each station/sub-station along with S-Boxes then it will be very difficult task for the hacker.

The Synchronization Time (S_{ti}) has been very important because transmitter and receivers needs to be run the encryption and decryption algorithm at the same time for optimized and secured data transmission. In public key cryptography every user that wants to send or receive secure electronic mail needs a valid key pair. In a networked environment, the user might need to use electronic mail from multiple computers, all with different operating systems. This creates a need for some kind of key sharing scheme. Therefore it is required to synchronize the speed of transmitter(s) and receiver(s). In Fig. 1, it is necessary to match the transmitting speed of A with the receiving speed of S_1 and so on. Also required to determine the other parameters such as time delay, processing time of individual stations, bandwidth of channel, and number of collisions occurred during the transmission. Compression techniques are used to make the system effective for adhoc networks. If the cipher text is transmitted through the web then it should be compressed either by using Direct Cosine Transformation (DCT) or by wavelet technique [10]. In this JPEG compression technique is used, where the image is split up into 8×8 squares. Each square is transformed by using DCT results a multi dimensional array of 63 coefficients at the output. Then a quantizer rounds each of these coefficients, which is the required compression stage (where data is lost, small unimportant coefficients are rounded to 0 where as larger ones lose some of their precision. Then array of streamlined coefficients, are used further compressed (Huffman encoding may be used). At the receiver section decompression and decryption are done by an inverse DCT and decryption model [1, 5].

Padding is required when the plaintext and the key length does not matches. Consider a block cipher with a block size of B bits. It is often the case that the length of the natural input will not be of multiple of B bits. Then, two options are possible: (1) the user should detect the partial block and handle it accordingly, or (2) the implementation should accept partial blocks and use a strategy to map it to a full block before the ciphering translation. Zero padding or PKCS padding can be used depending upon the requirement [12]. The method basically acts as overheads but it does not cost the overall model so much. In case of zero padding the zeros are added automatically as a result it does not take much time. Uniform lengths are always preferred in order to achieve reliable and optimized results. If mismatch sequences are taken then it will increases the probability of error. Also the latency time increases in such cases. So partial blocks are selected and processed by the padding process to make them of equal sizes.

Table 4. Padding Techniques.

Key	**11010101**
Plaintext	01001
Zero Padding	01001**000**
Encrypted Data	10011101
Decrypted Data	01001**000**

4. Conclusion and Future Scope

Practical systems are not easy to handle, sometimes they behave very awkwardly, and while designing a system we always have to make a system by keeping an eye on the worst case. Design issues are not easy to adopt, depending upon the nature they are taken. The algorithm used in the paper shows that all the stages are mutually independent from each other, provides the facility to use compression and re-encryption techniques. It is still required to reduce the time available for the hacker to make the system more reliable. Same can be done by re-encrypting the data by another key. The aim is to develop an encryption algorithm which provides multiple keys generated from the data itself and must be upgraded automatically.

References

1. Zou, C.C.; Gao, L.; Gong, W.; and Towsley, D. (2003). Monitoring and early warning for internet worms. *Proceedings of the* 10th *ACM Conference of Computer and Communication Security*, 190-199.
2. Zou, C.C.; Gong, W.; and Towsley, D. (2002). Code red worm propagation modeling and analysis. *Proceedings of* 9th *ACM Conference of Computer and Communication Security*, 138-147.
3. Deng, J.; and Han, S. (2008). Multipath key establishment for wireless sensor networks using just-enough redundancy transmission. *IEEE Transactions on Dependable and Secure Computing*, 5(3), 177-190.
4. Chien, H.-Y. (2004). Efficient time-bound hierarchical key assignment scheme. *IEEE Transactions on Knowledge and Data Engineering*, 16(10), 1301-1304.

5. Mhatre, V.P.; Rosenberge, C.; Kofman, D.; Mazumdar, R.; and Shroff, N. (2005). A minimum cost heterogeneous sensor network with a lifetime constraint. *IEEE Transactions on Mobile Computing*, 4(1), 4-15.
6. Ross, S. (19965). *Stochastic Processes*, (2nd Ed.), John Wiley & Sons.
7. Vaurio, J.K. (2002). Treatment of general dependencies in system fault-tree and risk analysis. *IEEE Transactions on Reliability*, 51(3), 278-287.
8. Tzeng, W.G. (2002). A time-bound cryptographic key assignment scheme for access control in a hierarchy. *IEEE Transactions on Knowledge and Data Engineering*, 14(1), 182-188.
9. Yi, X. (2005). Security of Chien's efficient time-bound hierarchical key assignment scheme. *IEEE Transactions on Knowledge and Data Engineering*, 17(9), 1298-1299.
10. Prasad, M.K.; and Lee, Y.H. (1994). Stack filters and selection probabilities. *IEEE Transactions on Signal Processing*, 42(10), 2628-2643.
11. Amari, S.V.; Dugan, J.B.; and Misra, R.B. (1993). A separable method for incorporating imperfect fault-coverage into combinatorial models. *IEEE Transactions. Reliability*, 48 (3), 267-274.
12. Lingqun, W; Yingping, Z.; and Shu, Z. (2005). Application of data mining and data protection in medicine. *Computer Engineering*, 31(10), 54-56.