Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 3, Issue. 5, May 2014, pg.1119 – 1128

RESEARCH ARTICLE

A Unique Approach to Multimedia Based Dynamic Symmetric Key Cryptography

Vinay Verma[1], Rajesh Kumar[2]

[1]Department of Computer Science and Engineering, Punjab Technical University, Jalandhar, India

[2]Department of Computer Science and Engineering, Punjab Technical University, Jalandhar, India

[1] vinay21_verma@yahoo.co.in; [2] rajeshkengg@gmail.com

Abstract: In the past decades, Most of the cryptography and steganography based techniques emphasized on securing text or images only. They made use of traditional approaches like transpositions or substitutions for text security and image transformation techniques for image security. None of them completely ensures the safety of data from bruit-force methods, applied by spy-bots. To provide more security large key size and complex procedures have been adopted. This creates extra computational overhead for CPU. Some algorithms introduced security of MMS and Videos, but they primarily focused on encrypting few image frames from entire video with traditional image cryptography. In this paper, we propose an efficient symmetric key based security approach that generates dynamic key from a multimedia file to encrypt any type of multimedia data. In this technique, the dynamic key is chosen randomly from the multimedia file using a special function. This approach gives zero loss results in terms of all the user data and the multimedia content used for encryption. The use of simple operations to select the key and to encrypt the data makes it faster.

Keywords: Steganography, Cryptography, Multimedia Dynamic Key, Symmetric Key, Zero Loss, Video Cryptography

I. INTRODUCTION

Nowadays, user data gets highest priority in the field of data communication on internet. The data must be communicated securely so as to keep the internet usage reliable. For this security of data, several techniques have been introduced. Some popular techniques are cryptography and steganography, water marking etc. Each of these techniques has some problems associated with them.

In case of traditional cryptography the sender either uses the transposition cipher or substitution cipher. Whereas, all the standard symmetric key algorithms use large key size and complex operations to achieve more invisibility, such as, Data Encryption Standard (DES), Advance Encryption Standard (AES), and Blowfish etc. Some other algorithms like 3DES use multiple keys and multiples levels of cryptography to make it even more difficult for an attacker to decipher user data. No doubt, these techniques are very much reliable and secure, but they also put large computational overhead on CPU and consume network bandwidth. As the security levels increases, security increases, but the overall complexity also increases [1].

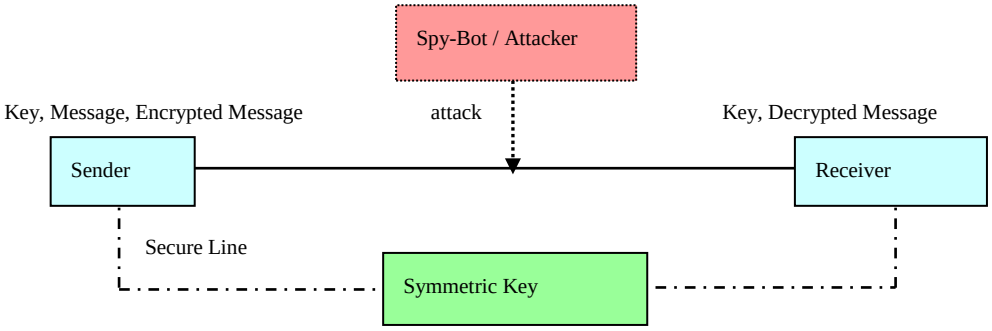

Fig. 1 Traditional Cryptographic System

In the present scenario of cryptosystems, the algorithms make use of multiple keys in the ciphering process, which is a good alternative [2]. But, multiple keys are hard to generate and exchange. It also consumes network bandwidth with high traffic.

Some algorithms in recent times have introduced the encryption of images and video streams by using various transformations like DCT, orthogonal transformations, confusion and diffusions, But they do not guarantee lossless operation in decryption process [3]. In order to overcome these problems, we have designed a dynamic symmetric key algorithm to achieve fast, accurate and Zero Loss results.

The work proposed here suggests a technique which uses a dynamic bit pattern (key) to encrypt the data. We can apply this technique to many types of multimedia files. As the dynamic key is generated from the multimedia file, Greater the size of file, the greater the number of keys for encryption. Also pixel depth, bit position and any other multimedia file properties are the part of bit pattern and hence enhance security without affecting the cryptography process.

Rest of this paper is organized as follows. Section 2 discusses the computational overhead or security limitations in some popular existing symmetric key algorithms. Section 3 explores the proposed work (i.e. Key Generation and Dynamic Key Symmetric Encryption algorithm). Section 4 will discuss various performance metrics to assess our work. Simulation results will be discussed in section 5 followed by conclusion in section 6.

II. LITERATURE SURVEY

Various encryption algorithms have been developed in the past decades. Some algorithms worked very efficiently, but they still have some overheads relating to them. There has been a miraculous improvement in the field of cryptography, since the internet security concept has come into existence, the old algorithms have still their importance in various applications and so before discussing the current research work, it is important to have a glance on the old popular symmetric key algorithms.

Horst Feistel developed Data Encryption Standard, a symmetric key algorithm to encrypt block of data which is based upon Balanced Feistel network. The algorithm uses 16 different iterations on the data and was designed to operate on different modes. The key size of DES is 56 bits only, which can be broken by using bruit-force techniques [4].

To enhance the performance of DES, Triple DES was introduced in 1998 which has a key size of 64 bits. Triple DES encrypts the data by using three different keys. Each block of data is encrypted by different three 56-bit keys. The sender and receiver have to follow complex procedures to accomplish the encryption and decryption process in 3 levels. Also the large key size increases the complexity of algorithm. Moreover the keys are of fixed size and do not change with time. Thus the attacker has always a chance to find the key by applying bruit-force techniques [5].

Rijmen V, Daemen J (1998) developed The Advanced Encryption Standard (AES). In AES block size of 128 bits is fixed, and key lengths are varied which can be of 128-bits, 192-bits or 256-bits [6].

Kim et al. (2005) have proposed symmetric key cryptography algorithm for a video file. A protection theme for MPEG-4 video file format has been developed by authors. During this protection theme, minimum segments of each Video Object Plane in an every MPEG-4 video file can be encrypted with any symmetric key cryptography algorithm. So people that have not received permission and/or paid to use the contents wouldn't be able to read and view them. Author applied this scheme to all kinds of MPEG-4 Video Object Plane types that is I-, P- and B severally [7].

Debnath Bhattacharya et al. (2009) have proposed an approach to steganography to ensure data security [3]. In this algorithm they combined cryptography, steganography and an extra security layer to encrypt text messages. The algorithm used two public keys and one primary key to encrypt data. This layer concept increases computation time thrice to encrypt/decrypt the text message. Moreover the algorithm works only for text messages [1].

V.S.Shankar Sriram et al. (2010) introduced an interesting Block Cipher Multiple Key Symmetric Encryption (BCMKSE) algorithm to reduce computational time as well as message overhead [8]. As this approach is symmetric key based, two private keys which are NOBS (no of bits) and the Key K are computed and shared by

sender and receiver by some other modes. The key K is computed by various components like generating MINi (client node related), MINs (server related), SRMPNi (based on screen resolution and mouse position) and Ti (which is a time component). In the same manner NOB is calculated by applying various XOR, addition functions on these components. It is very difficult for eavesdropper to find out the private keys. This symmetric approach is very much secure, but this approach is there is a lot of overhead to compute the values of Key K and NOB which are fixed values and can be assumed as well to reduce computational overhead. Also the key comprises of a combination of server and client related information, which can be easily accessed and used by an attacker to generate the key and the number of permutations for NOB are very less. Any sophisticated attacker can find the key with little overhead of applying permutations. Moreover, this technique has an extra overhead of using MD5 to ensure message integrity [2].

Aditee Gautam et al. (2011) proposed a new technique using block based transformation which is used for image encryption. In this algorithm image is transformed into other image before encryption process is done. Blowfish algorithm is used for this transformation [1]. Blowfish has a 64-bit block size and a variable key length from 32 bits up to 448 bits. It is based upon 16 rounds. There are number of key dependent permutations, variable keys used on data at each round with XOR, addition operations. Blowfish is fast block algorithm, when the same key is used. But computation time increases with changing keys. Each new key requires pre-processing equivalent to encrypting text, which is very slow compared to other algorithms. Secondly, if there is lack of synchronization between two parties then it will not be able to use the same key of encryption and decryption [8].

Mazloom et al. (2011) has proposed a new symmetric key cryptography for images. This algorithm use 128 bits of secret key. This algorithm widely used confusingly–diffusion architecture which utilizes the concept of chaotic (2D) Standard map and (1D) Logistic map. This algorithm is specifically designed for the color images, which are 3D arrays of RGB data stream. The initial conditions and system parameters of the chaotic maps constitute the secret key of the algorithm. This algorithm used mixing properties of horizontally and vertically adjacent pixels using a Logistic map, for getting higher security and complexity [9].

Niraj et al. (2013) discussed various issues and challenges in symmetric key based cryptographic algorithm for videos and images. Author has highlighted that when the ciphered text is created, it is decrypted into plain text without any loss, whereas, the encrypted images may not be decrypted to actual images without loss. Also, traditional cryptosystems take much time to encrypt/decrypt the images because the image size is usually larger than the text message. The author has also discussed that Video data cannot be directly encrypted or decrypted. First video data is converted into a number of image frames and then traditional cryptography algorithms like DES, AES and RSA are applied on individual image frames. These algorithms are not suitable for video as well as color images. Although some of the lossless encryption systems use the symmetric properties of the orthogonal transform to evaluate the inverse of the orthogonal matrix in the decryption process to speed up the operations and reduce the cost of performance but they also have impose additional overhead of other transforms, Key-Gen algorithms and XOR operations that has increased the encryption time severely [10]. The authors have also focused on the issues and challenges in multimedia cryptography techniques. The author discussed various techniques of image cryptography and video cryptography including [9] and [7]. These techniques use complex procedures and chaotic confusions and diffusions for better security. As these techniques are based on traditional text based encryption algorithms like DES, AES, and RSA, therefore it may be possible that the decrypted image may lose some data.

For all the other above mentioned algorithms, two common problems are there. First is the much computation is required for different iterations. Second is the unique key size. The key is fixed and it doesn't change with the respective session. So, using a single key for long time doesn't empower the message secrecy. Along with this, these techniques encrypt the message to a satisfactory level, therefore the cipher text can be cracked by using some complex permutation algorithm

Multi-Layer Data Security algorithms combining Cryptography and steganography are used to increase the level of security, but it also tend to increase the response time and hence increasing the time/space complexity, which is not acceptable for large volumes of data.

III. PROPOSED APPROACH

Although many different types of techniques are available for solving encryption problem, but keeping in mind the above discussed problems by using Multimedia Based "Dynamic Symmetric Key Cryptography" (DSKC) is more suitable due to the following important reasons:

- Multiple Keys.
- It can handle any type of data like audio, video, text, images etc
- Zero Loss of User data & Key File (Data Integrity).
- High speed of computation (Minimum Computational Overhead)
- Enhanced Security.

Objectives

- Development of a Multimedia Based "Dynamic Symmetric Key Cryptography" algorithm.
- Coding of the algorithm in an appropriate programming package, ensuring its portability.
- To ensure maximum security of data by using the multiple keys.
- To take the advantage of both steganography and cryptography for innovation of new Dynamic key symmetric encryption algorithm.

To reduce the computational overhead of encryption/ decryption, as compared to earlier technologies by keeping the computations as minimum as possible To provide more security as well as more options to secure different types of data Multimedia based dynamic symmetric key cryptography algorithm has been proposed to provide more security by using multiple keys so that it became hard for eavesdropper to find out the sequence of key which is being used to encrypt the data. The proposed approach will be tested on some examples. It will be observed with experimental results that how the proposed approach could generate the most appropriate encryption for getting the most desired results as expected. The footsteps towards the proposed approach are as follows.

1) Generation of Parameter INC (to decide the bit pattern as key from the multimedia file)
2) Generation of dynamic key from multimedia file (Repeatedly for each sat of data values)
3) Applying the key with minimum computations on user data to get cipher data.
4) Sharing the key generation parameters. (either privately or by using other most secure methods/public key algorithms)
5) At receiver end, decrypting the data, with the same key and same version of the software.

Encryption algorithm

1) Select the multimedia file (S) that is needed to be secured.
2) Select the multimedia key file (k) by which data is to be encrypted. /* The size of multimedia key file (K) should be greater than user file (S).*/
3) Choose any number (Num) randomly./* which works as symmetric key between sender and receiver and it is to be shared privately. This random number helps in calculating the dynamic multimedia key from the multimedia key file*/
4) Calculate the encryption parameter by equating-

Inc= [Size of (K)* (Num* 2)] / Size of (S)

5) Repeat while EOF(S)

Open the user file S and multimedia key file K in binary mode

Read N bit data from the source file (User file S).

Use the encryption parameter INC as follows to move at random position in multimedia key file (K), form where N bit key is selected (equal to number of bits read from source file).

(Num/2) + i*inc.

/*Select the N bit key from key file K for encrypting the current set of source data. The N bit key is calculated every time for each set of N bit user data. (It does not pose extra overhead for CPU because it involves only read operation.) */

6) Calculate the encrypted data bits by X-ORing source N data bits and N key bits. Store the encrypted data bits in a new file named ENC. (This file is to be sent to the receiver via unsecured line). /* The encryption process has been kept simple because the key used is dynamic and the time complexity is to be reduced*/

Decryption Algorithm

Select the encrypted file (ENC) that is to be decrypted.

7) Select the same multimedia key file (k) that was used in encryption process. /* The eavesdropper can not find the set of random bits used as dynamic key because the INC parameter is secret.*/
8) Use the private key (Num) to calculate the decryption parameter that is shared privately or by some other secure means.
9) Calculate the decryption parameter by equating-

Inc= [Size of (K)* (Num* 2)]/ Size of (ENC)

10) Repeat while EOF(ENC)

Open the encrypted file ENC and multimedia key file K in binary mode

Read N bit data from the encrypted file ENC.

Use the decryption parameter INC as follows to move at random position in multimedia key file (K), form where N bit key is selected (equal to number of bits read from encrypted file).

(Num/2) + i*inc.

/*Select the N bit key from key file K for decrypting the current set of encrypted data. */

11) Calculate the decrypted data bits by X-ORing encrypted N data bits and N key bits.

IV. PERFORMANCE / EFFICIENCY METRICS

The organizations suffer big losses due to growing number of attacks. The wastage of time and resources caused to users result in lost revenues as time is money in on-line business. Also the terrorist organizations can harm a country by stealing their valuable secret information from the network. As most of the cryptography techniques do not assure complete and efficient security, there is a need to analyze their features in detail, so as to come up with an innovative approach that resolves all issues. In current work, we propose a powerful and efficient technique that ensures safety of users' data from nearly all types of attacks. To measure the efficiency and performance of our algorithm, it is required to be compared with several similar existing algorithms using the benchmarked performance metrics. We have measured performance using following metrics:

Encryption and Decryption Time

These are the basic parameters for measuring the performance of any algorithm. These parameters are used to evaluate the computational overhead of the cryptography algorithms [11].

Encryption and Decryption Throughput

To evaluate computational overhead of cryptography and steganography algorithms several time based parameters are used. Some of the algorithms use Encryption and Decryption throughput, whereas some use Encryption and Decryption time. The former one is more realistic and accurate because throughput gives more accurate results, as being independent of file size. The performance of every algorithm is computed mainly by using this basic parameter [11].

Invisibility

The invisibility is one of the most important requirements of a steganography algorithm, because the strength of steganography lies in its ability to be unnoticed by the human eye. If an Eavesdropper comes to know that an image has been tampered, the algorithm is likely to be compromised. We have used this metric because our mechanism also makes use of multimedia file as key, like it happens in steganography.

Payload capacity

The main principle of steganography is to hide communication and therefore requires sufficient embedding capacity keeping in mind that it can lead to increasing the space complexity of the algorithm [12].

Robustness against statistical attacks

Statistical analysis is the practice of detecting hidden information by applying statistical tests. Most of the algorithms leave a mark that can be easily detected through statistical analysis. To be able to pass by an eavesdropper without being detected, an algorithm must not leave such a mark in the encrypted file as be statistically significant [13].

Robustness against manipulation

In the secure communication, the file may undergo changes by an eavesdropper in an attempt to remove hidden / encrypted information. File manipulation, can be performed before it reaches its destination. Depending on the manner in which the message is embedded, these manipulations may destroy the hidden message. It is preferable for the algorithms to be robust against either malicious or unintentional changes [13].

Independent of file type

With many different file types used for communication on the Internet, it might seem suspicious that only one type of file is continuously communicated between two parties. The most powerful algorithms thus possess the ability to encrypt any type of information in any type of file [13].

Key Strength

The key strength of an algorithm can be expressed in many ways. Some algorithms use real time parameters (screen resolution & time stamp etc.) and complex procedures to calculate a difficult key. Some of the algorithms

prefer using large key size in order to make it difficult for the eavesdropper to find the key or crack the encrypted file, whereas some algorithms employ multiple keys to encrypt the data in multiple rounds and increase levels of secrecy. The actual definition of key strength is not practical one until the key really makes the user's data safe.

V. RESULTS AND DISCUSSIONS

This section will show the results of DSKC which are compared with performance of existing algorithms for different input sizes.

A. Encryption and Decryption Time:

In the Following Table and Figure we show the performance of various cryptography algorithms. Here we also compare the encryption and Decryption time of standard algorithm with DSKC.

TABLE I
Encryption Time Comparison with Existing Symmetric Key Algorithms

Input size in Kbytes	AES	3DES	DES	RC6	Blowfish	DSKC
49	56	54	32	41	36	14
59	38	48	35	24	36	16
100	90	81	45	60	37	20
247	112	111	55	77	45	22
321	164	167	70	109	45	23
694	210	226	94	123	46	26
899	258	299	88	162	64	27
963	208	283	250	125	66	27
5345.28	1237	1466	448	695	122	62
7310.336	1366	1786	1695	756	107	74

TABLE II
Average Encryption Time Comparison with Existing Symmetric Key Algorithms

	AES	3DES	DES	RC6	Blowfish	DSKC
Average Encryption Time	374	452	389	217	60.3	31.1

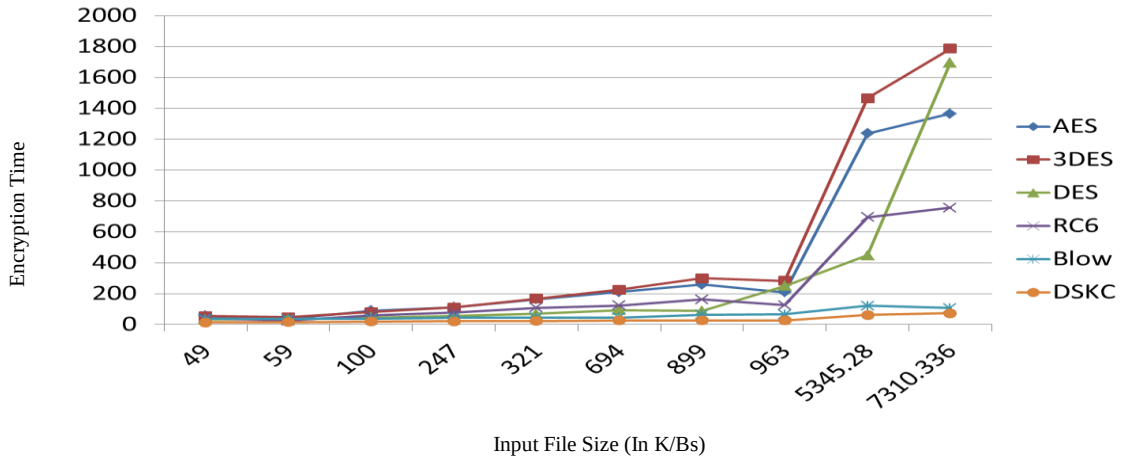

Fig. 2 Encryption time comparison with existing symmetric key algorithms

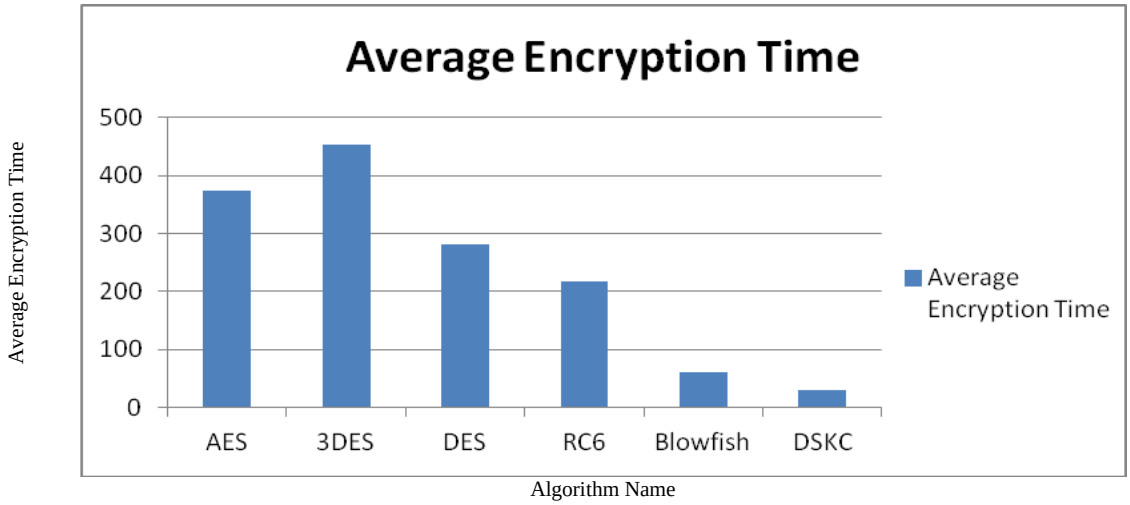

Fig. 3 Average Encryption time comparison with existing symmetric key algorithms

TABLE III
Decryption Time Comparison with Existing Symmetric Key Algorithms

Input size in Kbytes	AES	3DES	DES	RC6	Blowfish	DSKC
49	63	53	23	35	38	14
59	58	51	33	28	26	16
100	60	57	42	58	52	21
247	76	77	53	66	66	22
321	149	87	67	100	92	23
694	142	147	88	119	89	25
899	171	171	121	150	102	27
963	164	177	157	116	80	28
5345.28	655	835	910	684	149	59
7310.336	882	1101	953	745	140	70

TABLE IV
Average Encryption Time Comparison with Existing Symmetric Key Algorithms

	AES	3DES	DES	RC6	Blowfish	DSKC
Average Decryption Time	242	275.6	244	210	83.4	30.5

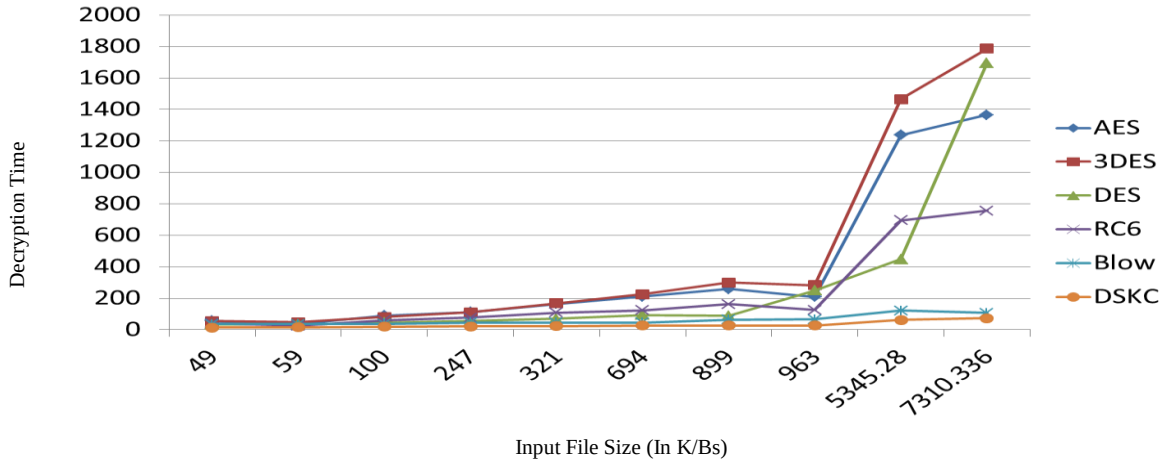

Fig.4 Decryption time comparison with existing symmetric key algorithms

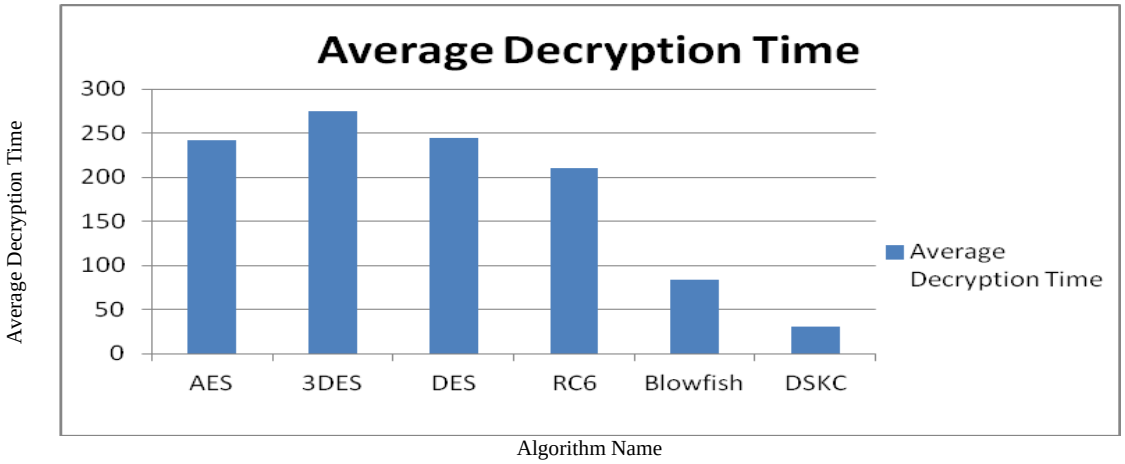

Fig.5 Average Decryption time comparison with existing symmetric key algorithms

Encryption and Decryption Throughput:

In the following Table and Figure we compare the encryption and decryption throughput of standard algorithm with DSKC.

TABLE V
Average Encryption Time Comparison with Existing Symmetric Key Algorithms

	AES	3DES	DES	RC6	Blowfish	DSKC
Encryption Throughput (Mb/sec)	4.27	3.53	5.56	7.36	26.47	51.41

TABLE VI
Average Encryption Time Comparison with Existing Symmetric Key Algorithms

	AES	3DES	DES	RC6	Blowfish	DSKC
Decryption Throughput (Mb/sec)	6.452	5.665	6.38	7.43	18.72	51.189

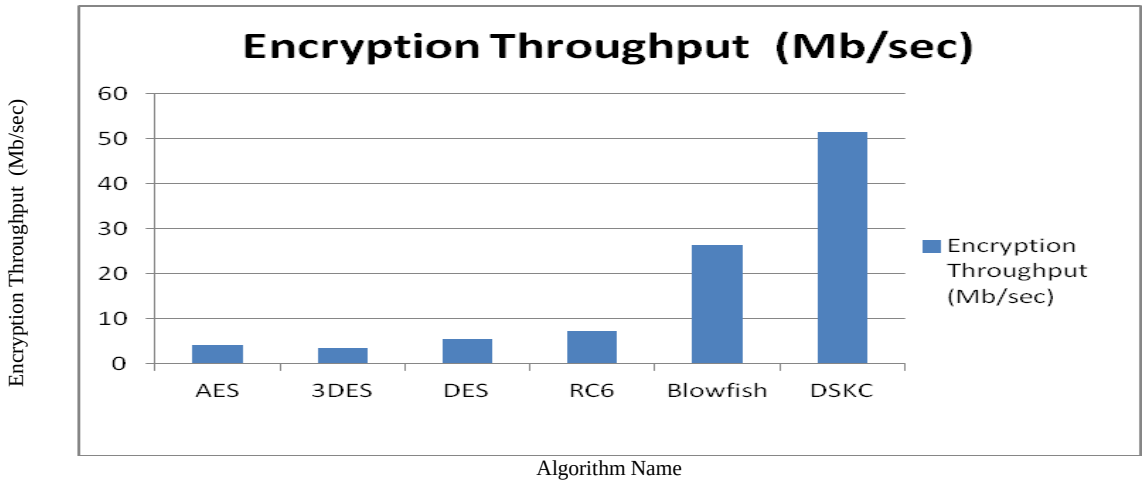

Fig.6 Encryption Throughput comparison with existing symmetric key algorithms

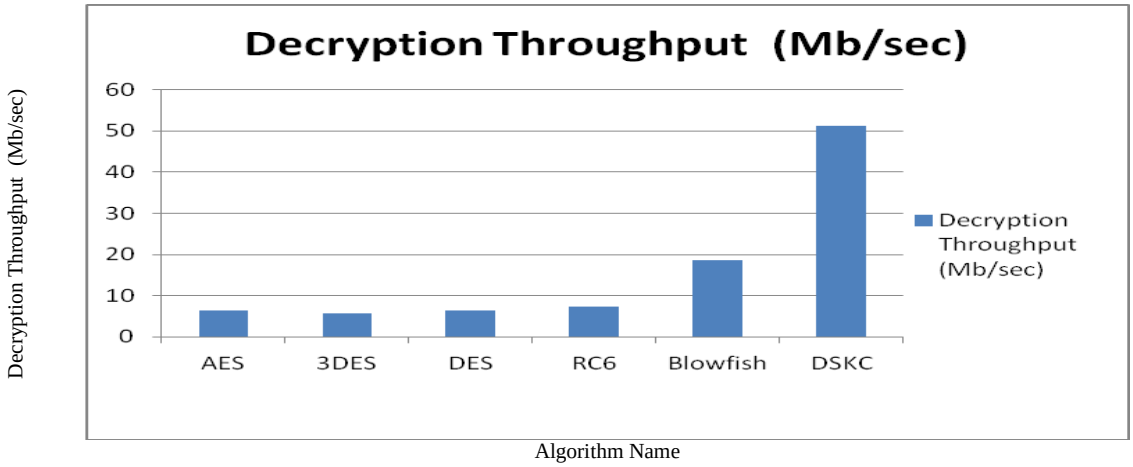

Fig.7 Decryption Throughput comparison with existing symmetric key algorithms

Comparison of various Algorithms with DSKC:

In the Following Table we compare the various standard algorithms with DSKC by using performance metrics discussed in previous section.

TABLE VII
Comparison of other algorithms with DSKC using various performance metrics

	AES	**3DES**	**DES**	**RC6**	**Blowfish**	**DSKC**
Invisibility	Low	Low	Low	Low	Low	High
Payload Capacity	Medium	Medium	Medium	Medium	High	Very High
Robustness against statistical attacks	High	Medium	Medium	High	Low	Low
Robustness against manipulation	Low	Low	Low	Low	Low	Low
Independent of file type	Some	Some	Some	Some	Some	All
Key Strength	Average	Low	Low	High	High	Very High

CONCLUSION

This work presents a performance comparison of standardized algorithms with DSKC. The performance metrics are Encryption and Decryption time, throughput, Invisibility, Payload Capacity, Robustness against statistical attacks, Robustness against manipulation, Independent of file type, Key Strength. The results show that DSKC has better performance and key strength as compared to other techniques of cryptography. Along with this DSKC has another advantage of multimedia cryptography i.e. this algorithm can be used to encrypt any type of file. This paper also shows that, using a large key size doesn't necessarily increase the strength of cryptography algorithm, but it surely increases time and space complexity of the algorithm. So our techniques emphasizes on using dynamic key so as to provide even more security without compromising with efficiency.

Our future work will try to incorporate public key cryptography for sharing the private key so as to come up with a next level of security. We will emphasize on making this technique even stronger by making it robust against manipulation and statistical attacks.

ACKNOWLEDGMENT

We would like to thank all the faculty members of Bhai Gurdas Iinstitute of Engineering & Technology, Sangrur for their valuable comments and suggestions that have improved the presentation of this paper.

REFERENCES

1. Debnath Bhattacharyya, Poulami Das, Samir Kumar Bandyopadhyay and Tai-hoon Kim, "*Text Steganography: A Novel Approach*", in *International Journal of Advanced Science and Technology*, Vol 3, Feb 2009.
2. V.S.Shankar Sriram, Abhishek Kumar Maurya, G.Sahoo, "*A Novel Multiple Key Block Ciphering Mechanism with Reduced Computational Overhead*", in *International Journal of Computer Applications*, Vol.1 (No.17):25–30, February 2010.
3. Guanrong Chen, Yaobin Mao and Charles K. Chui "*A symmetric image encryption scheme based on 3D chaotic cat maps*", in Elsevier Chaos, Solitons and Fractals 21 (2004) 749–761.
4. Federal Information Processing Standards Publication 46-3, "Data *Encryption Standard (DES)*", U.S. DoC/NIST, October 25,1999
5. American National Standard for Financial Services 1998, "*Triple Data Encryption Algorithm Modes of Operation*", American Bankers Association, Washington, D.C. X9.52- July 29, 1998.
6. Jawahar Thakur, Nagesh Kumar, "*DES, AES and Blowfish: Symmetric Key Cryptography algorithms simulation based performance analysis*", in *International Journal of Emerging technology and Advanced Engineering*, Volume 1, Issue 2,December (2011).
7. Gunhee Kim; Dongkyoo Shin; Dongil Shin, "*Intellectual property management on MPEG-4 video for hand-held device and mobile video streaming service*", Consumer Electronics, IEEE Transactions on , vol.51, no.1, pp.139,143, Feb. 2005.
8. Aditee Gautam, Meenakshi Panwar and Dr.P.R Gupta "*A New Image Encryption Approach Using Block Based Transformation Algorithm*", *international Journal of advanced engineering sciences and technologies* Vol No. 8, Issue No. 1, 090 – 096, 2011.
9. Mazloom, S.; Eftekhari-Moghadam, A.M., "Color *image cryptosystem using chaotic maps*", Computational Intelligence for Multimedia, Signal and Vision Processing (CIMSIVP), 2011 IEEE Symposium on , vol., no., pp.142,147, 11-15 April 2011.
10. Niraj Kumar, Prof. Sanjay Agrawal, "*Issues and Challenges in Symmetric Key based Cryptographic Algorithm for Videos and Images*", in *International Journal of Advanced Research in Computer Science and Software Engineering*, Volume 3, Issue 5, May 2013.
11. Diaa Salama Abdul. Elminaam, Hatem M. Abdul Kader and Mohie M. Hadhoud, "*Performance Evaluation of Symmetric Encryption Algorithms on Power Consumption for Wireless Devices*", in *International Journal of Computer Theory and Engineering*, Vol. 1, No. 4, October, 2009.
12. Pranam Paul, Saurabh Dutta, A K Bhattacharjee, "*An Approach to ensure Security through Bit-level Encryption with Possible Lossless Compression*", in *International Journal of Computer Science and Network Security*, VOL.8 No.2, February 2008.
13. Gajendra Singh Chandel, Ravindra Gupta, Swati Jain, "*Proposed Model of Dynamic encryption using Steganography*" in *International Journal of Emerging Technology and Advanced Engineering*, Volume 2, Issue 9, September 2012.
14. Nidhi Singhal, J.P.S.Raina, "*Comparative Analysis of AES and RC4Algorithms for Better Utilization*", in *International Journal of Computer Trends and Technology*, July to Aug Issue 2011.
15. D.S. Abdul Elminaam, H.M. Abdul Kader, M.M. Hadhoud, "*Performance Evaluation of Symmetric Encryption Algorithm",* in *Communications of the IBIMA*, Volume 8, 2009 ISSN: 1943-7765.

