International Journal of Computer Theory and Engineering, Vol. 1, No. 5, December, 2009
1793-8201

Impact of Multiencryption in Data Security

W.R. Sam Emmanuel, *Member, IAENG*

***Abstract*—The protection of data is the key mechanism for data security. The challenges faced to protect the data explore new algorithms. The evolution of encryption of encryptions in the field of cryptography may provide better security than single encryption routine. In this context, the multi-encryption came in picture. This paper deals the multi-encryption approaches of RC6 and XTEA block ciphers. The result and analysis of these approaches produce the security level and the processing speed.**

***Index Terms*— Multiencryption, RC6, XTEA.**

I. Introduction

The need for automated tools for protecting files and other information stored on the computer became evident. The generic name for the collection of tools designed to protect data[1] and to thwart hackers is computer security. Security involving communications and networks is not a simple as it might first appear to the novice. The security mechanisms[2] used to meet the requirements like confidentiality, authentication, non-repudiation and integrity can be done by the cryptographic algorithms.

In general, four types of cryptographic schemes typically used to accomplish these goals: Block Ciphers, Stream Ciphers, hash functions and public-key cryptosystems[3]. Most current cryptographic algorithms are designed for high performance. The structure, functional primitives and storage requirements of cryptographic algorithms relate to their energy consumption. An algorithm's structure indicates how well it lends itself to parallelization and serialization. The multiencryption[4][5][6] and multihasing are increasing the algorithm's security by applying it repeatedly. These can also enable ultra low-power cryptography. When we run a block cipher or hash functions several times in series, will produce a more secure overall cipher or hash function. Multiencryption can also increase the security level without increasing the footprint.

This paper shows the implementation of the block cipher algorithms, RC6[7][8][9] and XTEA[10] in the form of multiencryption approach. The two algorithms were implemented with two different approaches and the results were compared with the DES[11], AES[12]. The implementation is done by JDK1.3.

II. RC6 Block Cipher

RC6 is a fully parameterized family of encryption algorithms. A version of RC6 is more accurately specified as RC-w/r/b where the word size is w bits, encryption consists of a non-negative number of rounds r, and b denotes the length of the encryption key in bytes. The key schedule algorithm is used to generate the set of subkeys. The user supplies a key of b bytes, where $0 \leq b \leq 255$. From this key, 2r+4 words (w bit each) are derived and stored in the array s[0,…,2r+3]. This array is used in both encryption and decryption.

The following procedure shows the encryption steps of RC6 algorithm.

a. RC6 works with four w-bit registers A,B,C and D which contain the initial input.
b. The register value B is added with s[0] and store it in B.
c. The register value D is added with s[1] and store it in D.
d. Repeat the steps for r times
   - *t = (B x (2B+1)) <<< lg w*
   - *u = (D x (2D+1)) <<< lg w*
   - *A = ((A Å t) <<<u) + S[2i]*
   - *C = ((C Å u) <<<t) + S[2i + 1]*
   - Assign the value of B to A, C to B, D to C and A to D
e. Add the register value A with S[2r+2] and store it in A.
f. Add the register value C with S[2r+3] and store it in C.
g. Combine the register values A, B, C and D, which is the cipher text.

The decryption is the reverse process of the encryption

III. XTEA Block Cipher

The Tiny Encryption Algorithm (TEA) is a cryptographic algorithm designed to minimize memory footprint and maximize speed. The TEA was extended to XTEA (eXtended TEA) proposed to fix the two minor weakness. The first enhancement is to adjust the key schedule and the second is to introduce the key material slowly.

The encryption routine of the XTEA is displayed here, the number of cycles as N.

a. The input block (64 bits) is equally divided into two parts as y and z.
b. Assign delta as 0x9E3779B9 and sum as zero.
c. The limit is calculated as the product of delta and N.
d. Repeat the steps until the sum is not equal to the limit.
   - *y = y + (z<<4 ^ z>>5) + z^sum + k[sum&3]*
   - *sum = sum + delta*
   - *z = z + (y<<4 ^ y>>5) + y^sum + k[sum>>3&3]*
e. Combine the ciphertext which is stored in y and z.

Manuscript received June 4, 2009.
W.R. Sam Emmanuel is with the Computer Science Department, Nesamony Memorial Christian College, Marthandam – 629165, Kanyakumari District, Tamil Nadu, INDIA (e-mail: sam_emmanuel@ yahoo.com).

The reverse process will produce the plain text.

IV. Multiencryption

Multiencryption increases the algorithm's security by applying it repeatedly. The block ciphers consume little power but have a small security margin and run them several times in series, thus obtaining a more secure overall cipher. The most important requirement for a new cryptographic algorithm is scalability. Implementers should be able to scale the algorithm from a bit-serial implementation to a highly parallel implementation depending on the desired maximum power consumption and speed. Multiencryption can enable ultralow-power cryptography.

V. Proposed Approach

The researcher proposed the multiencryption by the serial implementation of the block ciphers XTEA and RC6. Two types of approaches were followed: (a) Multiencryption with XTEA after RC6, (b) Multiencryption with RC6 after XTEA. The two approaches were produced different results for different set of inputs.

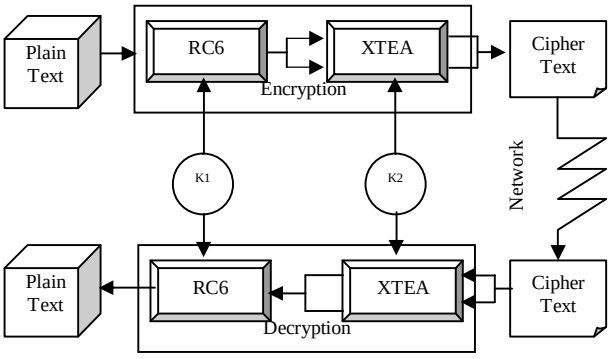


Fig-1 Multiencryption with XTEA after RC6

A. Multiencryption with XTEA after RC6

The following procedure gives full details for converting the plaintext into ciphertext using the algorithms RC6 and XTEA. The XTEA is applied with the ciphertext, which is produced after applying RC6 algorithm. The following procedure gives the encrypted text of this method.

a. The plaintext is divided into number of 128-bits block.
b. Select the 128-bit block one by one and apply RC6 algorithm.
c. Split the 128-bit ciphertext into two 64-bit blocks.
d. Apply XTEA algorithm separately in each 64-bit block.
e. Combine the 64-bit ciphertext blocks into 128-bit block.
f. Repeat the steps (b) to (e) up to end of the plaintext.

The encryption and decryption processes of this approach are shown in Fig-1 with full detail. The RC6 and XTEA algorithms used the keys K1 and K2 for encryption and decryption.

B. Multiencryption with RC6 after XTEA

In this method the XTEA algorithm is used first. The Fig-2 shows the combined approach of the multiencryption with this method. The keys, K1 and K2 are used in XTEA and RC6 respectively. It has the encryption and decryption routine. The steps to convert the plaintext in the form of ciphertext are explained here.

a. The plaintext is divided into number of 64-bits block.
b. Select two 64-bits block and apply XTEA algorithm separately.
c. Combine the two 64-bit ciphertext into single (128-bit) block.
d. Apply RC6 algorithm with the input text as the ciphertext of step (c).
e. The result (128-bit) is the ciphertext of the given plaintext.

Repeat the steps (b) to (e) upto end of the plaintext.

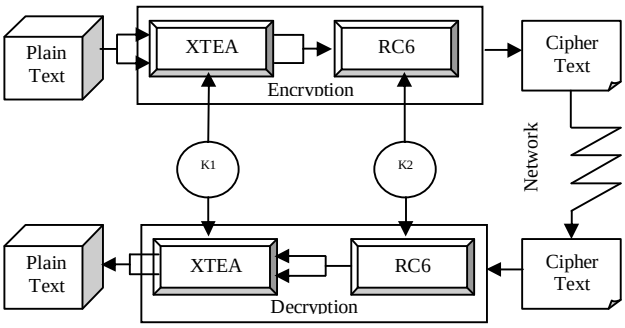


Fig-2 Multiencryption with RC6 after XTEA

VI. Results and Discussion

The performance figures' here is given for an optimized Java implementation of RC6, XTEA and multiencryption. This implementation was compiled with JDK1.3. The performance was measured on 2.40GHz Pentium IV with 256MB of RAM running in Windows XP. Each set of the timing tests described here was executed 5 times, and report the average of the times thereby obtained.

A. Analysis of Correlation Coefficients

In this correlation coefficient analysis, the researcher analyzes the correlation between the bits of plaintext and the corresponding bits of ciphertext. If the correlation coefficient equals to zero, then the plain text and cipher text are totally different. If the correlation coefficient is equal to -1 then the ciphertext is the negative of the plain text. If the correlation coefficient is perfect correlation then the cipher text and the plain text are same.

The correlation coefficient is measured by the following formulae.

Correlation coefficient, $r_{xy} = \frac{\mathrm{cov}(x,y)}{\sigma_x \sigma_y}$ (1)

Standard Deviation of x, $\sigma_x = \sqrt{\frac{\sum (x-\bar{x})^2}{N}}$ (2)

Standard Deviation of y, $\sigma_y = \sqrt{\frac{\sum (y-\bar{y})^2}{N}}$ (3)

Covariance, $\mathrm{cov}(x,y) = \frac{\sum (x-\bar{x})(y-\bar{y})}{N}$ and (4)

Mean $\bar{x} = \frac{\sum x_i}{N}$ (5)

Table-I Correlation between plaintext and ciphertext

Algorithm Used	Text file content		
	Alphabets	Digits	Alphanumeric
RC6	-0.0024	0.0024	-0.0019
XTEA	-0.0323	0.0651	0.0134
DES	-0.0482	0.0979	0.0155
AES	-0.0172	0.0293	-0.0341
Multiencryption (XTEA after RC6)	0.0166	-0.0695	0.0019
Multiencryption (RC6 after XTEA)	0.0226	-0.0665	0.0221

The analysis carried out with three different type of data files which contains alphabets only, digits only and alphanumeric text respectively, which shown in Table-I. Here, the two multiencryption approaches produce the negative of the plain text if the plain text is full of digits. For the alphanumeric values the plain text will produce totally different cipher text, when the 'multiencryption of XTEA after RC6' has been used.

B. Analysis using Histograms

One of the important factors in examining the encrypted text is the visual inspection where the highly disappeared features of the text the better the encryption algorithm.

To prevent the leakage of information to an opponent, it is advantageous if the cipherimage bears little or no statistical similarity to the plaintext. The histogram illustrates how the characters in a textfile are distributed by graphing the number of characters at each ascii number level. Here the researcher calculated and analyzed the histograms of the several encrypted as well as original files that have widely different content.

The Fig-3 and Fig-4 shows the histograms of the plaintext and the ciphertext when we used alphabets, digits and alphanumeric. When we use XTEA first it provides better results than the application of RC6 first. This is clearly described in the Fig-3. The fig-4 shows the graph to represent the ascii numbers' count for each character levels. The comparison of the diagrammatic representations will provide the pitfalls of the implementation. Some of the histogram contains large spikes. These spikes correspond to the ascii values that appear more often in the corresponding text file.

When the variations of the ascii characters are equally distributed then we can get better result. The variations of gray levels are equally distributed in the AES algorithm when we compare all other three algorithms (RC6, DES and XTEA). The multiencryption provides best result when the XTEA used first. The histogram of the ciphertext for multiencryption with XTEA first is more uniform, significantly different from that of the plaintext and bears no statistical resemblance to the plaintext. It is clear that the histogram of the encrypted text is fairly uniform and significantly different from the respective histograms of the plaintext and hence does not provide any clue to employ any statistical attack on the proposed multiencryption procedure. The diagrammatic representation shows the histograms of Alphabets, digits and alphanumeric text.

To estimate the quality of the process of multiencryption, it is necessary to study the evolution of the entropy.

Entropy, $H = -\sum_{i=0}^{2^R-1} n_i \log_2^{P(n_i)}$, where n_i is the ascii value of character, $P(n_i)$ the probability to find this character and R the number of bits per character. Entropy allows to have an idea of the redistribution of pixels and the number necessary for transmission by network.

C. Analysis Based on Encryption and Decryption Speed

The encryption and decryption process speed were calculated separately using the text file with the file size 24KB. It has 1398 blocks, each of which having 128-bits. Most of the algorithms produced less decryption time comparing to encryption, at the same time the conversion rate of the decryption is more.

Table-II Conversion Speed

Algorithm Used	Encryption/Decryption (E/D)	Conversion Time (ms)	Conversion Rate	
			(KB/ sec)	(128bits Blocks/ Sec)
RC6	E	1500	16.00	0932
	D	1234	19.45	1133
XTEA	E	1719	13.96	0813
	D	1704	14.08	0820
DES	E	1469	16.34	0952
	D	1484	16.17	0942
AES	E	1390	17.27	1006
	D	1375	17.45	1017
Multiencryption RC6 with XTEA	E	2218	10.82	0630
	D	2172	11.05	0644
Multiencryption XTEA with RC6	E	2422	09.91	0577
	D	2266	10.59	0617

When the researcher compare the conversion rate between multiencryption and normal encryption, the multiencryption produced less conversion rate. For the encryption, the multiencryption conversion rate is 9.18% more if the RC6 algorithm used first while for the decryption 4.34%. This is described in the Table-II.

VII. CONCLUSION

This paper introduces the impact of multiencryption in data security by RC6 and XTEA block ciphers. The testing, verification, efficiency analysis and security evaluation of multiencryption is done by three types of text files having only alphabets, only digits and alphanumeric text. The quality evaluation and the comparisons were done by simulation programs. The statistical analysis like correlation coefficients between plain text and cipher text, histogram analysis and the efficiency analysis gives the accuracy of the results. The combined approach of the operations in the RC6 and XTEA block ciphers will improve the efficiency of the conversion rate. This may provide best algorithm with better security and conversion rate.

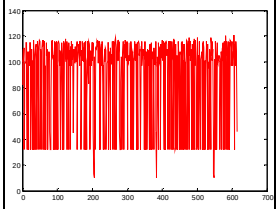
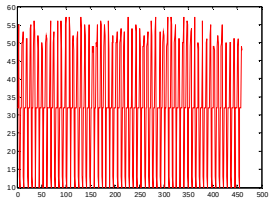
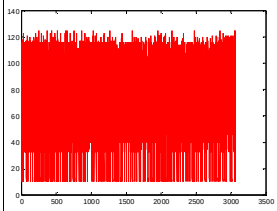
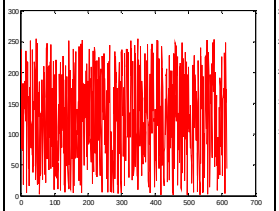
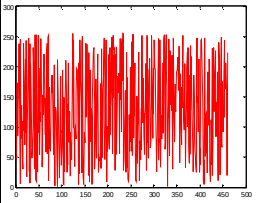
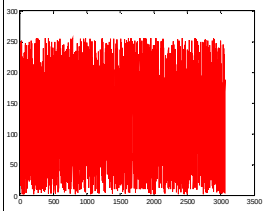
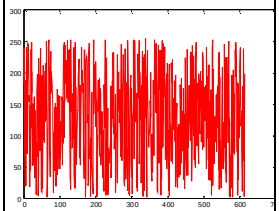
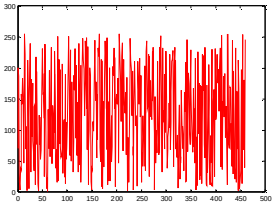
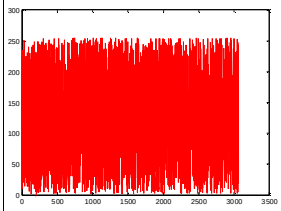
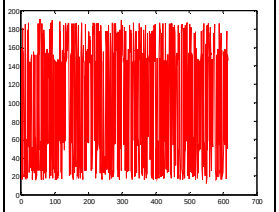
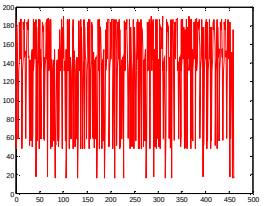
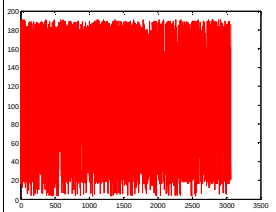
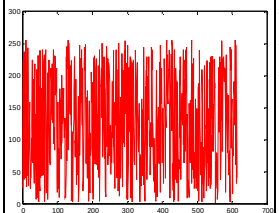
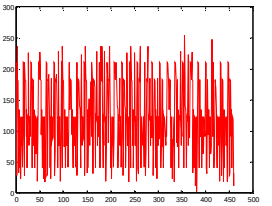
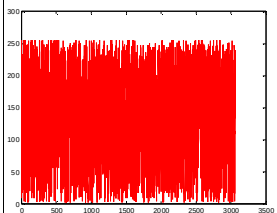
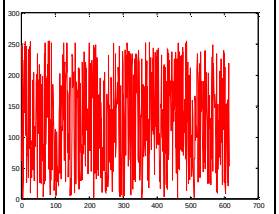
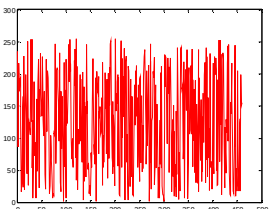
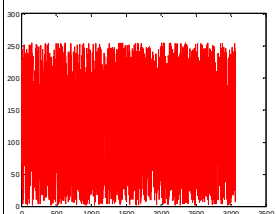
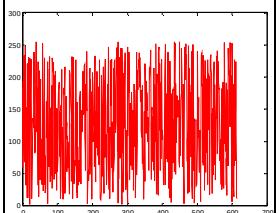
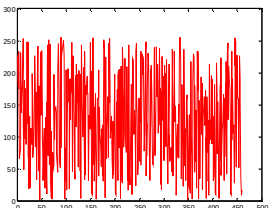
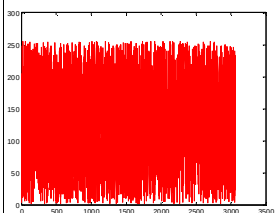
	Alphabets	Digits	Alphanumeric
Input			
RC6			
XTEA			
DES			
AES			
Multiencryption (XTEA after RC6)			
Multiencryption (RC6 after XTEA with)			

Fig-3 Histogram for the Encrypted Text

	Alphabets	Digits	Alphanumeric

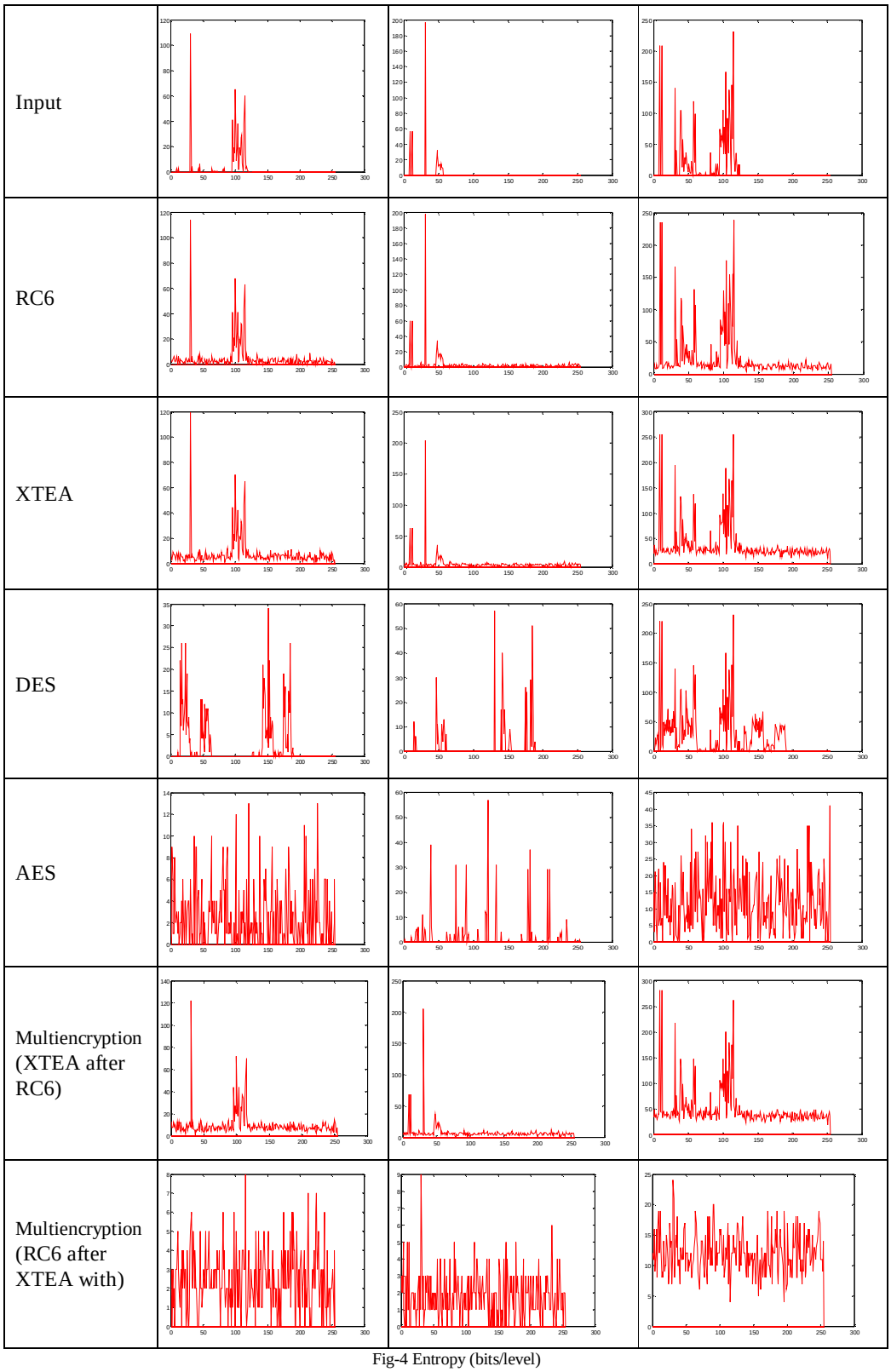


Fig-4 Entropy (bits/level)

REFERENCES

[1] Osama S. Farag Allah, Abdul Hamid M. Ragib, and Nabil A. Ismali, "Enhancements and Implementation of RC6 Block cipher for data security", IEEE Catalog Number: 01CH37239, 2001.

[2] Bruce Schneier, "Applied Cryptography – Protocols, algorithms, and source code in C", 2nd ed., John Wiley & Sons Inc., New York., 1996.

[3] William Stallings, "Cryptography and Network Security Principles and Practices", 3rd Ed., Pearson Education, 2004.

[4] Jens-Peter Kaps, Gunnar Gaubatz and Berk Sunar, "Cryptography on a Speck of Dust", IEEE Computer Society, 2007, pp.38-44.

[5] R. Zhang, G. Hanaoka, J. Shikata, and H. Imai, "On the security of multi-layered encryption or CCAsecurity+CCA- security=CCA-security?", SCIS'03, 2003.

[6] Tomasz Müldner, Jan Krzysztof Miziołek and Gregory Leighton, "Using Multi-Encryption to Provide Secure and Controlled Access to XML Documents", Extreme Markup Languages, 2006, pp.1-11.
[7] R.L.Rivest, M.J.B.Robshaw, R.Sidney and Y.L.Yin. (1998). The RC6TM Block Cipher. Available: http://www.rsasecurity.com /rsalabs/rc6/.
[8] RSA security, "The RC6 Block cipher", Cryptographic Technique Specifications.
[9] RSA security, "The RC6 Block cipher", Self Evaluation Report.
[10] D.Moon, K.Hwang, W.Lee, S.Lee and J.Lim, "Impossible differential cryptanalysis of reduced round XTEA and TEA", Proceedings of the International workshop on Fast Software Encryption, 2002.
[11] National Institute of Standards and Technology (NIST), "Data Encryption Standard (DES)", Federal Information Processing Standards Publication 46-3 (FIPS PUB 46-3), 1999.
[12] J.Daeman and V.Rijmen. (1999). AES Proposal : Rijndael. Available: http://www.esat.kuleuven.ac.be/rijmen/rijndael/ rijndaeldocV2.zip

W.R. Sam Emmanuel, from INDIA, DOB:05-06-1975. In 1995 he received his BSc. degree in Mathematics from MS university, Tirunelveli. He has awarded master's degree in Computer Applications from Bharathidhason university, Tiruchi in 1998. He also received master's degree of library science from Alagappa University, Karaikudi, India in 2006. He has two MPhil degrees in the fields of Computer Science and Library Science received in 2002 and 2007 respectively. He is currently pursuing the Ph.D. degree in Vinayaka Mission's University, Salem, India.

He is working as a SENIOR LECTURER at the Computer Science Department of Nesamony Memorial Christian College, Marthandam, Tamil Nadu, India from the year 2000. He has more than 10 years teaching experience in the field of Computer Science. He has several publications in national and international journals. He has published the book "Data Encryption Algorithms" (India: Tony's Publications, 2007). His research interests include data encryption, image encryption, video encryption, compression, multimedia security, Cryptography, Security of e-resources.

Mr. Emmanuel is a member of "Computer Society of India", member of "Indian Society for Technical Education", and also the member of "International Association of Computer Science and Information Technology".