Real-Time Symmetric Cryptography using Quaternion Julia Set

P. M. Rubesh Anand[1], Gaurav Bajpai[2] and Vidhyacharan Bhaskar[1]

[1] Department of Electronics and Communication Engineering, SRM University, Kattankulathur – 603203, India
[2]Faculty of Engineering, Kigali Institute of Science and Technology, B.P. 3900, Kigali, Rwanda

Summary

In this paper, Quaternion Julia set is used to generate real-time based symmetric keys for cryptography. The number of iterations, complex number and control value are the determining parameters of dynamically varying quaternion Julia image structure. The considered parameters are initialised in the proposed model of symmetric key generation during the establishment of communication between hosts. The model generates variable length, dynamic, one time usable key from quaternion Julia image to encrypt or decrypt data without involving the exchange of key. The time stamp used during the initialization process makes the quaternion Julia image to be different in real-time. The instantaneous key is generated at the hosts independently in a synchronous fashion to enhance the complexity in cryptanalysis. The proposed model has wide range of applications from low confidential to high confidential data transfer in two party and multi-party scenarios.

Key words:

Cryptography, Quaternion Julia set, Symmetric key generation, Real-time encryption

1. Introduction

In cryptosystems, symmetric and asymmetric key algorithms are used. Symmetric or secret key algorithm uses one key whereas asymmetric or public key algorithm uses two different keys [1] [2]. Keys are generated by numerous ways in cryptography [3]. Most of the symmetric key generations are based on the pseudo random number theory [4] [5]. Fractals are also used in the key generation for cryptography [6]. In general, the symmetric key algorithms use unique key to be shared between the communicating hosts for data transfer. The methods used to share the unique key through the communication channel by the hosts are always insecure even though they are encrypted [7] [8]. This makes the symmetric key algorithms more vulnerable during attacks. The possible way to protect from such attacks is by avoiding the secret key exchange through communication channels and negotiating for the selection of common key by the communicating hosts through a secure and suitable technique. This paper proposes a model to generate real-time based key by using quaternion Julia fractal images. Quaternion Julia fractal image is considered for its complex image structure and chaotic behaviour. Small variations in the Julia parameters will result in a drastic change of image structure due to the chaotic nature of the mathematical function. The instantaneous symmetric key is generated by the hosts simultaneously without third party intervention and the exchange of key between hosts is barred to protect from attacks against the key during key exchange. The model produces different symmetric keys during the process of data transfer to enhance the complexity in cryptanalysis.

Further, this paper is organised into six sections. Section 2 covers the introduction to quaternion Julia sets, section 3 describes about the proposed model, section 4 shows the complexity in cryptanalysis, section 5 highlights the advantages along with applications of the proposed model and section 6 presents the conclusion.

2. Quaternion Julia Set

Julia sets are produced by a procedure of repeated iteration [9] [10]. The polynomial used in the process of iteration is quadratic, cubic, quartic or any higher order degree [11]. The parameters required to produce Julia set are *Z, C, n* and *m.* The Julia function is given as:

$$f(Z) = Z^n + C \tag{1}$$

Here, *Z* is a complex number consisting of two components, which are independent of each other called real and imaginary part as in equation (2), *C* is a constant, *n* is the degree of polynomial and *m* denotes the number of iterations.

$$Z = a + ib \tag{2}$$

When the value for *C* is a hyper complex number, there exist four components called Hamiltonian Quaternion [12] [13]. Quaternion Julia set also called as 4D Julia set as they allow four dimensional image structures. Quaternion consists of one real part and three imaginary parts which are denoted as *1, i, j, k* respectively.

$$C = jc + kd \tag{3}$$

Manuscript received March 5, 2009
Manuscript revised March 20, 2009

A quaternion $q \in H$ is a wider numerical extension of a complex value and consists of four components $a, b, c, d \in R$. The quaternion *q* is given as:

$$q = a + ib + jc + kd \quad (4)$$

where, $$i^2 = j^2 = k^2 = -1 \quad (5)$$

Quaternion multiplication is non-commutative and hence quaternion is not an abelian group. Quaternion Julia set of the quadratic iteration for fixed hyper complex *C* value [14] is given as

$$Z_{n+1} = Z_n{}^2 + C \quad (6)$$

3. Proposed Model

A new way of generating real-time symmetric key used for cryptography is shown in the proposed model. Instead of one key used in symmetric key cryptography, the proposed model uses multiple symmetric keys during the complete session of the data transfer. Fig. 1 shows the detailed structure of the model. Plain text *(M)* is encrypted by the instantaneously generated real-time symmetric key *(K$_i$)* to produce the cipher text *(C$_i$)*. The cipher text is then transmitted from the transmitting host through the communication channel.

$$E_{K_i}(M) = C_i \quad (7)$$

where, $$i = 1,2,3,.....\infty \quad (8)$$

The received cipher text *(C$_i$)* is then decrypted with the real time symmetric key *(K$_i$)* generated at the receiving host independently to get back the plain text *(M)*.

$$D_{K_i}(C_i) = M \quad (9)$$

The operation of the model is divided into four phases as follows:

3.1 Connection Establishment Phase

The transmitting host should establish the connection with the receiving host through SSL handshake. During the connection establishment, the transmitting host should send the date and time stamp which is calculated from its database such that the new time stamp can be calculated by the receiving host for verifying the authentication. The transmitting host should calculate from the new time stamp for the value of sending time stamp. The proposed model assumes that the connection establishment is made in real-time to avoid the time differences in calculation.

Sending time stamp = New time stamp -
Host machine current time stamp (10)

Once the time stamp is shared, both the hosts will perform the next phase simultaneously as shown in Fig. 2.

3.2 Key Generation Phase

Real-time Symmetric key generator uses quaternion Julia set to generate different symmetric keys of variable lengths. The key generator is initialized by time stamp as shown in Fig. 3. When the connection establishment phase is over, the received date and time stamp is given to the key generator where a new time stamp is produced by adding the received time stamp and the current time stamp from the host machine for the purpose of authentication.

New time stamp = Received time stamp +
Host machine current time stamp (11)

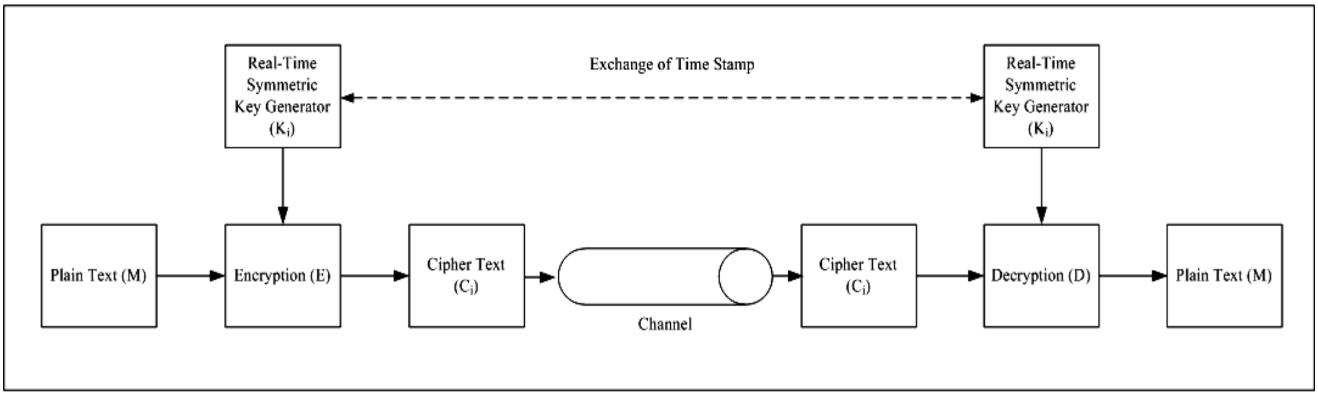


Fig. 1 Proposed Real-time symmetric key cryptography

The new time stamp is available in the database as an authentication of the transmitting host. This is to avoid random time stamp attack for studying the key generation algorithm. Once a new time stamp is calculated, it is checked from the database for initializing the Julia parameters. Attackers cannot repeat the same date and time stamp for hosting attack at later time as the new time stamp uses the current time of the host machine. The initialized Julia parameters generate quaternion Julia image. A 3D plane is considered with angle initialized during the initialization of the Julia parameters. The intersection points of the 3D plane and the Julia image are plotted. Real-time symmetric key of required size is obtained from the plotted image.

3.3 Encryption and Decryption of Data

The symmetric key obtained from the key generator is used for encryption and decryption of data by the hosts respectively. The proposed model uses two types of encryption techniques. First encryption technique is by using XOR operation. This is for the low confidential and high speed data transfer. The encryption and decryption schemes are give in equations (12) and (13):

$$M \oplus K_i = C_i \tag{12}$$

$$C_i \oplus K_i = M \tag{13}$$

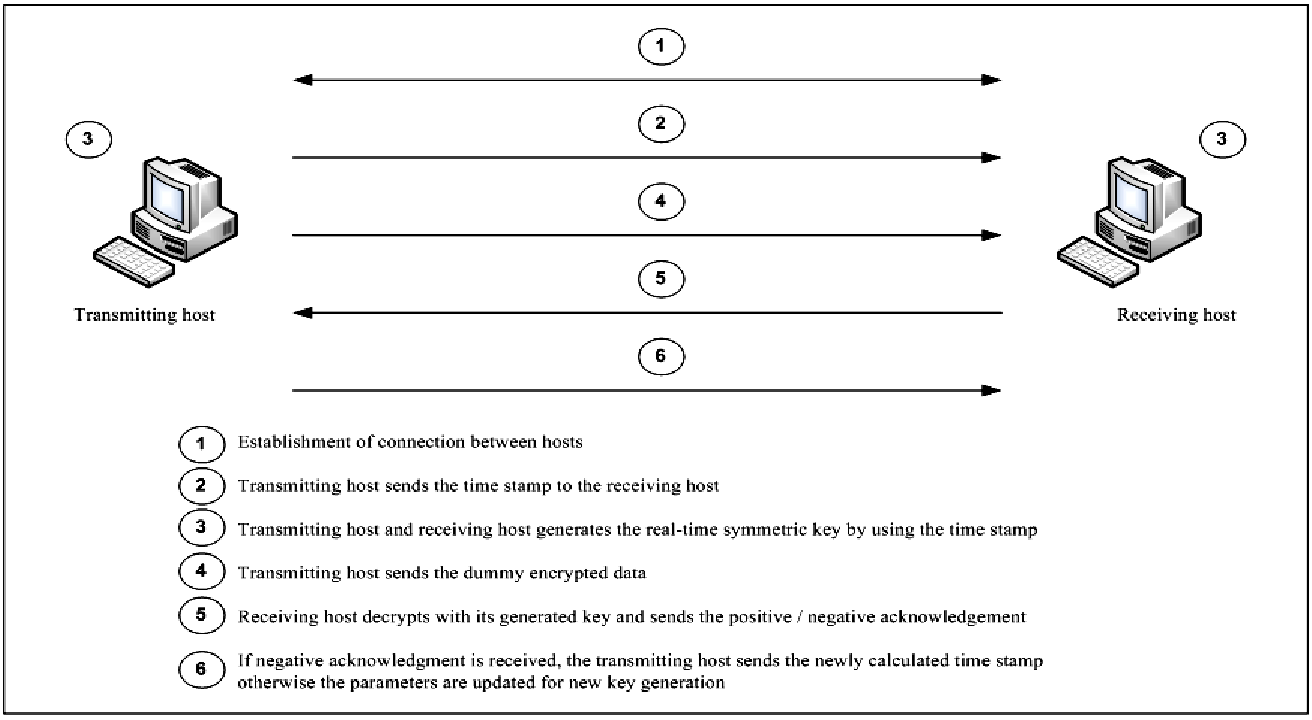
The second encryption technique is for the high confidential data where AES symmetric encryption algorithm [15] is used but with different keys from the real-time symmetric key generator for each block of data to be transferred. The wide range of keys generated from the model is used for one-time pad to enhance the security.

3.4 Acknowledgment Phase

In order to change the key, the model avoids sending time stamps every time. Instead, the receiving host will send a positive or negative acknowledgment for each block in case of block cipher and per session for stream cipher. At the initial stage, the new time stamp generated is tested with a dummy data block and once the attempt is successful, the receiving host will send a positive acknowledgment. When a positive acknowledgment is received, both the hosts simultaneously update the new time stamp by adding the number of iterations used in the last Julia set with the previous new time stamp.

New time stamp = Previous new time stamp +
Number of iterations in the last Julia set (14)

If a receiving host is unable to decrypt the dummy data block, it will send the negative acknowledgment. Once the negative acknowledgment is received by the transmitting host, it will calculate a fresh time stamp to start with a new session.



1 Establishment of connection between hosts

2 Transmitting host sends the time stamp to the receiving host

3 Transmitting host and receiving host generates the real-time symmetric key by using the time stamp

4 Transmitting host sends the dummy encrypted data

5 Receiving host decrypts with its generated key and sends the positive / negative acknowledgement

6 If negative acknowledgment is received, the transmitting host sends the newly calculated time stamp otherwise the parameters are updated for new key generation

Fig. 2 Time sequence operation of the proposed model

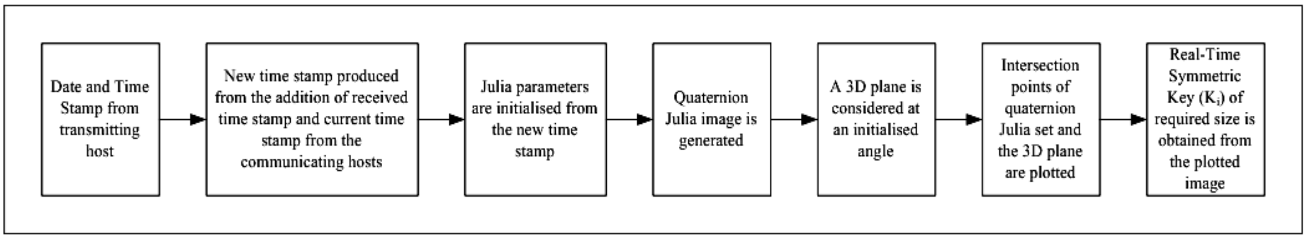

Date and Time Stamp from transmitting host
New time stamp produced from the addition of received time stamp and current time stamp from the communicating hosts
Julia parameters are initialised from the new time stamp
Quaternion Julia image is generated
A 3D plane is considered at an initialised angle
Intersection points of quaternion Julia set and the 3D plane are plotted
Real-Time Symmetric Key (K_i) of required size is obtained from the plotted image

Fig. 3 Real-time Symmetric key generator

3.5 Quaternion Julia Images

Quad 1.20 is a 3D fractal generator [16]. We have used the software to render the 3D image structure of the 4D Quaternion Julia set by making one of the dimension to be constant [17]. Additionally 3D intersection plane is defined for slicing the image. A part of the proposed model is tested by generating various quaternion Julia images and slicing them through 3D planes. Some of the results are given below in Fig. 4, Fig. 5, Fig. 6 and Fig. 7.

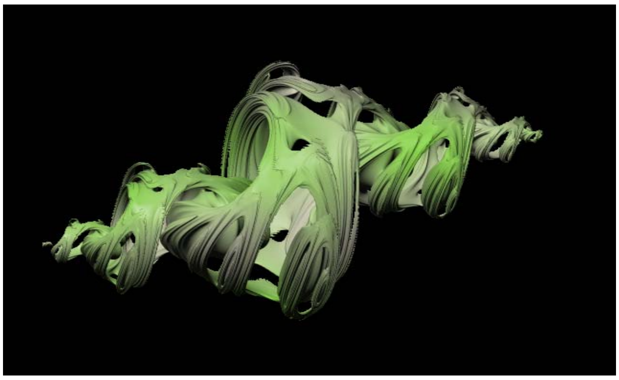


Fig. 4 Quaternion Julia set associated with
$C = 0.727 + -0.256i + 0.4j + 0.02k$
without intersection plane, iterations = 14

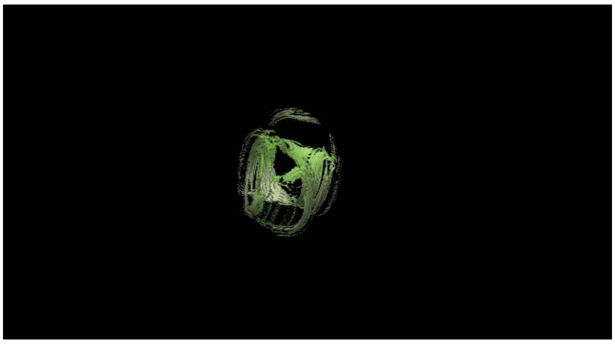


Fig. 5 Intersection of Quaternion Julia set associated with
$C = 0.727 + -0.256i + 0.4j + 0.02k$
with intersection plane $w = 0.542$, *Q-space normal vector=0.2+0.025i+0.1j*, iterations = 14

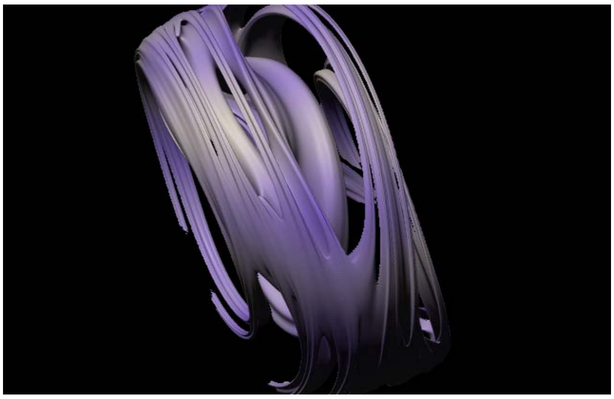


Fig. 6 Intersection of Quaternion Julia set associated with
$C = -0.5 + 0.6i + 0.4j + 0.5k$
with intersection plane $w = 0.09457$, *Q-space normal vector=i+j*, iterations = 10

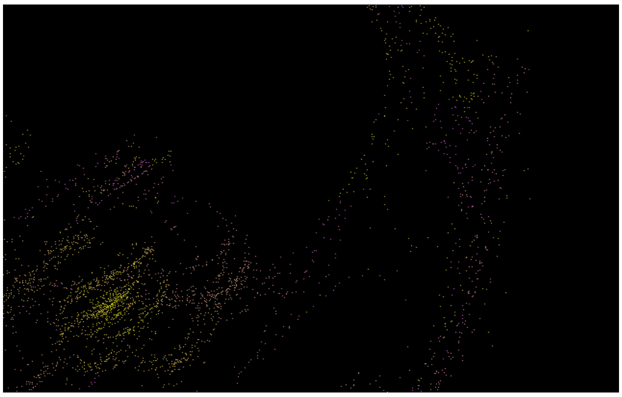


Fig. 7 Intersection of Quaternion Julia set associated with
$C = 0.7 + 0.5i + 0.2j + 0k$
with intersection plane $w = 0.00012$, *Q-space normal vector=0.2+0.025i+0.1j*, iterations = 50

A small variation in the Julia parameter shows an entirely different Julia fractal images. These dynamically varying images after slicing represent the instantaneous symmetric keys.

4. Complexity in Cryptanalysis

The main highlight of the proposed model is that the keys are not exchanged through the communication channel. This gives a very little information for security to be breached. The only way to get the plain text from the cipher text is by brute-force attack [18]. As the keys are changing in real-time for each block of data, the process of cryptanalysis will be a time and energy consuming. The real-time symmetric key in the proposed model undergoes three degrees of randomness along with the property of one-time pad. This makes the cryptanalysis a difficult process [19] compared to the life time of the message used. Due to the three degrees of randomness, the key prediction and cryptanalysis is a complex task. The three degrees of randomness exist in the generated key are as follows:

4.1 Quaternion Julia Set

Due to the chaotic behavior, a minor variation in the quaternion Julia parameters produces an entirely different image fractal. As there are infinite combinations of these parameters, the proposed model results with a nearly infinite number of new quaternion Julia images between intervals of time. The generation of quaternion Julia set is an irreversible process making it impossible to predict the parameters from the image structure and vice versa.

4.2 Slicing by 3D plane

The second degree of randomness is made by slicing the generated quaternion Julia image structure. The proposed model considers only the intersection points of quaternion Julia image with the 3D plane at a random angle to overcome the problem of self-similarity in the structure of the Julia fractal images. This makes the same quaternion Julia image with the 3D plane of different slicing angle results in entirely different points of intersection.

4.3 Variable key size

The third degree of randomness is obtained by selecting some random points from the full set of intersection points depending upon the size of key needed. The similar way of point's selection is employed at the encryption and decryption ends. The generated variable length key has high degree of anonymity for the process of encryption/decryption by stream cipher and block cipher of variable block size.

5. Advantages and Applications

Any efficient cryptography should satisfy the following properties [19] [20] and the proposed model is addressing them with positive remarks. Table 1 shows the comparison between general cryptographic algorithms and the proposed model [21]. The proposed model supports the four main properties of efficient cryptography.

Table 1: Comparison of different cryptographic algorithms

Cryptographic Algorithms / Property	Symmetric Encryption	Public-key Encryption	Digital Signature	Message Authentication Codes	**Proposed Model**
Confidentiality	Yes	Yes	No	No	**Yes**
Authentication	No	No	Yes	Yes	**Yes**
Integrity	No	No	Yes	Yes	**Yes**
Non-repudiation	No	No	Yes	Yes	**Yes**

5.1 Confidentiality

In the proposed model, each block of data is encrypted with different keys of variable length for one-time pad. The message of more blocks uses dynamic key for each block making the entire data transfer a highly confidential.

5.2 Authentication

The proposed model works in real-time using time-stamps that are calculated from the database. The receiving host verifies with its database for authentication. If a cryptanalyst performs the random time-stamp attack, the dynamic variation in the encryption/decryption key, acknowledgment process for each block of data can identify the unauthenticated users.

5.3 Integrity

Though there is a problem of modifying the data in symmetric key encryptions, the number of keys used in the proposed model is huge to make a cross verify of the received data. If an attacker knows the encryption scheme, it is not possible to generate every quaternion Julia image used in the encryption process without the knowledge of the parameter values that is available in the database. The data to be transmitted is compressed and fragmented into pieces with individual hash codes protect the data from being modified.

5.4 Non-repudiation

The sender cannot deny the sending of message as the model uses real-time cryptography along with the time stamp from the sender. The handshake process during the connection establishment verifies the sender's authentication through the time-stamp. As the time stamp is not selected in random at the transmitting host, the problem of repudiation is solved. The initialization processes are hidden from the users and a log is stored at a secure place for each attempt in establishing connection serves a powerful non-repudiation of the transmitting host.

5.5 Applications

The proposed model can be used for low confidential - high speed data transfer and high confidential - low speed data transfer by selecting the appropriate encryption key size and algorithm. This model supports data transfer between two party and multi-party. Starting from any secret number like credit card numbers to a very large data file like video on demand can be secretly transferred. In the multi-party data transfer scenario, the communicating hosts should exchange individual time stamps for each other party and can work simultaneously in data transfer through different ports. The proposed model can be securely implemented as software or dedicated customized hardware in the communicating hosts without giving way for compromising with the attackers.

6. Conclusion

Real-time cryptography is much demanded for network and information security. This proposed mathematical model has the advantage of generating instantaneous real-time symmetric keys by the hosts simultaneously and are not shared in the public channel. The dynamically varying keys hold the unpredictability nature making the data transfer secure. The quaternion Julia set has the chaotic nature which gives entirely different image structure for a small variation in the parameters. The proposed model works with complex image structure of quaternion Julia set instead of pseudo random number generation and the three degree of randomness in the generated key makes it difficult to predict the sequence. The proposed model has applications from low confidential to high confidential data transfer in two party and multi-party scenarios.

References

[1] Whitfield Diffie and Martin E. Hellman, “New directions in cryptography”, IEEE Transactions on Information Theory, Vol. 22, Issue 6, pp. 644-654, Nov. 1976.

[2] R. Rivest, A. Shamir, L. Adleman. “A Method for Obtaining Digital Signatures and Public-Key Cryptosystems”, Communications of the ACM, Vol. 21 (2), pp. 120–126, 1978.

[3] R. Blom, “An Optimal Class of Symmetric Key Generation System”, Advances in Cryptology - Eurocrypt’84, LNCS Vol. 209, pp. 335-338, 1985.

[4] K. Zeng, C.H. Yang, D.Y. Wei and T.R.N. Rao, “Pseudorandom bit generators in stream-cipher cryptography”, IEEE Computer, Vol. 24, Issue 2, pp. 8-17, February 1991.

[5] L. Blum, M. Blum and M. Shub. “A simple unpredictable pseudorandom number generator”, SIAM Journal on Computing, 15(2):364-383, 1986.

[6] Mohammad Ahmad Alia and Azman Bin Samsudin, “New Key Exchange Protocol Based on Mandelbrot and Julia Fractal Sets”, International Journal of Computer Science and Network Security, Vol. 7, No. 2, February 2007.

[7] Sung-Ming Yen, "Cryptanalysis of an authentication and key distribution protocol", IEEE Communications Letters, Vol. 3, No. 1, pp. 7-8, January 1999.

[8] J. Kelsey, B. Schneier, D. Wagner, “Key-Schedule Cryptanalysis of IDEA, G-DES, GOST, SAFER, and Triple-DES”, Lecture Notes in Computer Science, Issue 1109, pp. 237-251, Springer Verlag, 1996.

[9] Gaston Maurice Julia, “Memoir on iterations of rational functions”, In the Journal de Mathématiques pures et appliquées – 4th tome, 1918 (83th volume of the collection), published by Gauthier-Villars editor, Translated in English by Alessandro Rosa, May 15th, 2001.

[10] Douady A., "Julia Sets and the Mandelbrot Set", In the Beauty of Fractals: Images of Complex Dynamical Systems (Ed. H.-O. Peitgen and D. H. Richter). Berlin: Springer-Verlag, pp. 161, 1986.

[11] Hamilton W. R., “Elements of Quaternions”, Third Edition, Volume 1-2, Chelsea Publishing Company, New York, 1969.

[12] John Milnor, “Dynamics in one complex variable: introductory lectures”, Braunschweig : Vieweg, 2000.

[13] Zhang F., “Quaternions and Matrices of Quaternions”, Linear Algebra and Its Applications, Vol. 251, Issue 1-3, January 1997.

[14] Yumei Dang, Louis H. Kauffman, Daniel Sandin**,** “Hypercomplex Iterations: Distance Estimation and Higher Dimensional Fractals**”** (Series on Knots & Everything)**,** World Scientific Publishing Co. Pte. Ltd.**,** 2002.

[15] Advanced Encryption Standard (AES), Federal Information Processing Standards Publication 197, Available [Online]: http://www.nsa.gov/ia/programs/suiteb_cryptography/index.shtml, November 26, 2001.

[16] Quat 1.20, “A 3D Fractal Generator”, Available [Online]: http://www.physcip.uni-stuttgart.de/phy11733/quat_e.html

[17] Hart, John C., George K. Francis, and Louis H. Kauffman, "Visualizing Quaternion Rotation", ACM Transactions on Graphics, Vol. 13, No. 3, pp. 256-276, 1994.

[18] ECRYPT Yearly Report on Algorithms and Key sizes, (2007-2008), Available [Online]: http://www.ecrypt.eu.org/ecrypt1/documents/D.SPA.28-1.1.pdf, July 2008.

[19] Bruce Schneier, “Applied Cryptography: Protocols, Algorithms, and Source Code in C”, Second Edition, John Wiley & Sons, 1996.

[20] William Stallings, “Cryptography and Network Security: Principles and Practices”, Fourth Edition, Prentice Hall, 2006.

[21] B.S. Kaliski, “A Survey of Encryption Standards”, IEEE Micro, Vol. 13, Issue 6, pp. 74-81, December 1993.

P. M. Rubesh Anand received the B.E. degree in Electronics and Communication Engineering from Periyar University, India in 2002, and M.Tech. degree in Advanced Communication Systems from SASTRA University, India in 2004. Since 2005, he is working as a lecturer in the Faculty of Engineering, Kigali Institute of Science and Technology, Kigali, Rwanda. Currently, he is pursuing his Ph.D. research at SRM University, Kattankulathur, India. His research interests include communication networks, cryptography and network security.

Gaurav Bajpai received the B.Tech. degree in Computer Science & Engineering from SRMSCET Rohilkhand University, India in 2000, M.Tech. degree in Software Engineering from Motilal Nehru National Institute of Technology, Allahabad, India and Ph.D. degree from Uttar Pradesh Technical University, Lucknow, India in 2006. He was an assistant Professor in the Department of Computer Science and Business Administration, Academy of Medical Sciences and Technology, Khartoum, Sudan from April 2006 to March 2007. Since March 2007, he is working as a Senior Lecturer in the Department of Computer Engineering and Information Technology, Faculty of Engineering, Kigali Institute of Science and Technology, Rwanda. His research interests include software engineering, network routing, network hardware security and bio-medical engineering. He has published more than 30 International Journal and conference papers.

Vidhyacharan Bhaskar received the B.Sc. degree in Mathematics from D.G. Vaishnav College, Chennai, India in 1992, M.E. degree in Electrical & Communication Engineering from the Indian Institute of Science, Bangalore in 1997, and the M.S.E. and Ph.D. degrees in Electrical Engineering from the University of Alabama in Huntsville in 2000 and 2002 respectively. During 2002-2003, he was a post-doc fellow with the Communications research group at the University of Toronto. From Sep. 2003 to Dec. 2006, he was an Associate Professor in the Département Génie des systèmes d'information et de Télécommunication at the Université de Technologie de Troyes, France. Since January 2007, he is a Professor and Associate Dean of the School of Electronics and Communications Engineering at S.R.M. University, Kattankulathur, India. His research interests include wireless communications, signal processing, error control coding and queuing theory. He has published 25 International Journal articles as first author, presented 9 International Conferences, and co-authored a book on MATLAB.