Protection of Data from Cipher-Text Only Attack using Key Based Interval Splitting

A. Duraisamy, K. Somasundaram, M. Sathiyamoorthy

Abstract- Modifications of Arithmetic Coding (AC) is to improve the security in two methods are: RAC (Randomized Arithmetic Coding) and KSAC (AC with Key-based interval splitting). For the security, encryption uses AC that is based on the inability of the opponent to distinguish between the encryption of one plaintext from the encryption of another. Chosen plaintext attacks are insecure in RAC, because same key is used to encrypt different messages even random key is used for compress every messages. The new encryption scheme is used for improve security in RAC that is the encryption is performed by a bit wise X-OR of the compressed output with the pseudorandom bit sequence for chosen plaintext attacks. Then encryption scheme is used for improve security in KSAC that is the encryption is performed by a bit wise X-OR of the compressed output with the pseudorandom bit sequence for chosen plaintext attacks.

Keywords-AC, RAC, KSAC, Plaintext, Ciphertext, Plaintext attacks, AES

I. INTRODUCTION

Chosen plaintext for Randomized Arithmetic Code are based on same key used to encode many messages, known that using a same key for many messages leads to insecure encryption scheme .The strongest version of security is chosen-cipher text security where the adversary has access to the encryption and decryption engine (but does not know the secret key) and can decrypt any ciphertext of his choice or encrypt any plaintext to his choice. Every message will be compressed using a new key sequence achieved using a secure pseudorandom sequence generator. Arithmetic coding followed by XOR with a secure Pseudorandom bit sequence leads to an encryption scheme that is chosen-plaintext secure.

The XOR can be incorporated into AC thereby incurring minimal penalty for real-time applications. RAC that uses different key for different messages is not secure under cipher text only attack, but secure in chosen plaintext attack. When both compression and security are sought, one approach is to simply use a traditional arithmetic coder in combination with a well-known encryption method such as the Advanced Encryption Standard (AES).

Manuscript received on March, 2013.

A.Duraisamy, Department of Information Technology, University College of Engineering, Tindivanam (T.N), India.

K.Somasundaram, Department of Information Technology, SNS College of Engineering, Coimbatore (T.N), India.

M.Sathiyamoorthy, Department of Information Technology, University College of Engineering, Tindivanam (T.N), India.

However, while this will certainly meet both goals, it fails to take advantage of the additional design flexibility and potential computational simplifications that are available if the coding and encryption are performed jointly. Providing compression and security simultaneously is important because of increased use of compressed media files in many applications such as the internet, digital cameras, and portable music players.

To achieve both compression and security there are two possible approaches. One is to use traditional compression methods followed by an encryption method using a stream or block cipher. Another approach is to incorporate security by modifying the compression method. The latter approach offers additional design flexibility and hence computational simplifications. This may hence be a good approach to encrypt the large amounts of data in multimedia applications. Modified compression can be used along with selective encryption for real-time multimedia encryption. In selective encryption, only crucial parts of the multimedia data are encrypted.

Indistinguishability against ciphertext-only attacks (or indistinguishability in the presence of an eavesdropper) is the weakest form of security where the adversary can only eavesdrop on ciphertexts. A stronger version of security is chosen-plaintext security where the adversary has access to the encryption engine (but does not know the secret key) and can, therefore, encrypt any message of her choice. Thus the adversary has oracle access to the encryption engine.The pseudorandom bit sequence is derived in advance using Advanced Encryption Standard (AES) in the counter mode, then the first-compress-then-encrypt method results in a performance penalty of only a few two input XOR-gate delays. Chosen-plaintext attacks for both RAC and KSAC are based on the fact that the same key is used to encode many messages. However it is known that using the same key for many messages leads to insecure encryption schemes. We lift this restriction on RAC and KSAC. This implies that every message will be compressed using a new key sequence. This can easily be achieved using a secure pseudorandom sequence generator (e.g., Advanced Encryption Standard (AES) in the counter mode.

A. Randomized Arithmetic Code

Chosen plain text for Randomized Arithmetic Code are based on same key used to encode many messages, known that using a same key for many messages leads to insecure encryption scheme every message will be compressed using a new key sequence achieved using a secure pseudorandom sequence generator RAC that uses different key for different messages is not secure under cipher text only attack, But secure in chosen plaintext attack. To prove that RAC that uses different keys for different messages is not secure under ciphertext-only attacks.

Note that we assume that key length is negligible compared to message. A proof similar to the one in this paper could also be used to prove that KSAC is also insecure against ciphertext-only attacks. Indistinguishability against cipher text-only attacks (or in distinguishability in the presence of an eavesdropper) is the weakest form of security where the adversary can only eavesdrop on cipher texts.

Chosen-plaintext attack has been proposed for two methods when the same key is used to encrypt different messages. We first give the definition for security of encryption using AC that is based on the inability of the adversary to distinguish between the encryption of one plaintext from the encryption of another. Using this definition, we prove that RAC is insecure even if the new random key is used to compress every message. Our proof assumes that the only eavesdrop on the ciphertext and cannot request encryptions of chosen-plaintexts**.** A chosen-plaintext attack is the attack model for cryptanalysis which presumes that the attacker has the capability to choose arbitrary plaintexts to be encrypted and obtained the corresponding ciphertexts. The goal of the attack is to gain some further information which reduces the security of the encryption scheme.

In the worst case, a chosen-plaintext attack could reveal the scheme's secret key. For some chosen-plaintext attacks, only a small part of the plaintext needs to be chosen by the attacker: such attacks are known as plaintext injection attacks. This appears, at first glance, to be an unrealistic model; it would certainly be unlikely that an attacker could persuade a human cryptographer to encrypt large amounts of plaintexts of the attacker's choosing. Modern cryptography, on the other hand, is implemented in software or hardware and is used for a diverse range of applications; for many cases, a chosen-plaintext attack is often very feasible. Chosen-plaintext attacks become extremely important in a context of public key cryptography, where the encryption key is the public and attackers can encrypt any plaintext they choose.

B. Arithmetic Coding

Arithmetic coding is a form of entropy encoding used in lossless data compression. Normally, a string of characters such as the words "hello there" is represented using a fixed number of bits per character, as in the ASCII code. When a string is converted to arithmetic encoding, frequently used characters will be stored with fewer bits and not-so-frequently occurring characters will be stored with more bits, resulting in fewer bits used in total. Arithmetic coding differs from other form of the entropy encoding such as Huffman coding in that rather than separating the input into component symbols and replacing each with a code, arithmetic coding encodes the entire message into a single number, a fraction n where $(0.0 \leq n < 1.0)$. In general, each step of the encoding process, except for the very last, is the same; the encoder has basically just three pieces of data to consider:

- The next symbol that needs to be encoded.
- The current interval (at the very start of the encoding process, the interval is set to [0,1), but that will change)
- The probabilities of the model assigns to each of the various symbols that are possible at this stage.

One advantage of arithmetic coding over other similar methods of data compression is the convenience of the adaptation. Adaptation is the changing of a frequency table while processing of a data. The decoded data matches the original data as long as the frequency table in the decoding is replaced in the same way and in the same step as in encoding. The synchronization is, usually, based on a combination of symbols occurring during the encoding and decoding process. Adaptive arithmetic coding significantly improves the compression ratio compared to static methods, it may be as effective as 2 to 3 times better in the result.

C. AES Algorithm

AES is based on a design principle known as the substitution-permutation network, and is fast in both software and hardware. Unlike its predecessor DES, AES doesn't use a Feistel network. AES is a variant of Rijndael which has an fixed block size of 128 bits, and a key size of 128, 192, or 256 bits. By the Rijndael specification *per se* is specified with block and the key sizes that may be any multiple of 32 bits, both with a minimum of 128 and a maximum of 256 bits. AES operates on a 4×4 column-major order matrix of bytes, termed the state, although some versions of Rijndael have a larger block size and have additional column in the state. Most AES calculations are done in the special finite field. The key size used for an AES cipher specifies the number of repetition of transformation rounds that convert the input, called the plaintext, into the final output, called the ciphertext.

Key Expansion: Round keys are derived from the cipher key using Rijndael's key schedule.

AddRoundKey: Each byte of the state is combined with the round key using bitwise XOR.

SubBytes: A non-linear substitution step where each byte is replaced with an according to a lookup table.

ShiftRows: The transposition step where each row of the state is shifted cyclically a certain number of steps.

MixColumns: A mixing operation which operates on the columns of the state, combining the four bytes in the each column.

II. LITERATURE REVIEW

H.Kim et al [1], adopt an approach in which the intervals associated with each symbol, which are continuous in a traditional arithmetic coder, can be split according to a key known both to the encoder and decoder. Were move the constraint that the intervals corresponding to each symbol be continuous, and instead use a more generalized constraint that the sum of the length soft he one or more intervals associated with each symbol be equal to its probability.

Jiangtao (Gene)Wen et al [2], adopt an approach in which the intervals associated with each symbol, which are continuous in a traditional arithmetic coder, can be split according to a key known both to the encoder and decoder. For example, in a binary system with two symbols A and B and p (A) =2/3 and p(B) =1/3, a traditional partitioning would represent A by the range [0*;* 2/3) andB by the range [2/3*;* 1).

G.Langdon et al [3], describes a joint RAC/XOR encryption paradigm for efficient multimedia data protection is presented in this work. By exploiting the structure of entropy coder, the proposed scheme demands very low computation cost and can be easily implemented.

It is highly robust against various cryptanalytic attacks. The scheme provides good security and adds less over head. Future directions would be to increase the speed of the randomized binary arithmetic coder.J.Wen et al [4], adopt an approach in which the intervals associated with each symbol, which are continuous in a traditional arithmetic coder, can be split according to a key known both to the encoder and decoder. Were move the constraint that the intervals corresponding to each symbol be continuous, and instead use a more generalized constraint that the sum of the lengths of the one or more intervals associated with each symbol be equal to its probability.

N. K. Ratha et al [5], describes an idea of multiple snapshots to be taken in which more than one instance of the same biometric is used for the enrolment and/or recognition. For example, multiple impressions of the same finger, or multiple samples of the voice, or multiple images of the face may be combined. By doing these the error rate during the finger print input decreases drastically.

S.Goldwasser et al [6], given a public-key infrastructure (PKI) and digital signatures, it is possible to construct broadcast protocols tolerating any number of corrupted parties. Almost all existing protocols however, do not distinguish between corrupted parties (who do not follow the protocol), and honest parties whose secret (signing) keys have been compromised (but who continue to behave honestly). We explore conditions under which it is possible to construct broadcast protocols that still provide the usual guarantees (i.e., validity/agreement) to the latter.

D.J.C.Mackay et al [7], Cryptographic techniques are used to secure confidential data from unauthorized access, but these techniques are very sensitive to noise. A single bit change in encrypted data may have catastrophic impact over the decrypted data. This paper addresses the problem of removing bit error in visual data which are encrypted using AES algorithm by block. In order to remove the noise, three statistical analyses are proposed which are based on Global variance, Mean local variance and Sum of squared derivative. These methods exploit local statistics of the visual data and confusion/diffusion properties of the encryption algorithm to correct the errors. Experimental results show that the proposed approaches can be used at the receiving end for the possible solution for error correction in visual data in encrypted domain.

P. W. Moo et al [8], consider the problem of resynchronizing simple arithmetic codes. This research lays the groundwork for future analysis of arithmetic codes with high-order context models. In order for the decoder to achieve full resynchronization, the unknown, initial b bits of the code stream must be determined exactly. When the source is approximately IID, the search complexity associated with choosing the correct sequence is at least O ($2^{b/2}$). Therefore, when b is 100 or more, the time complexity required to achieve full resynchronization is prohibitively high. To partially resynchronize, the decoder must determine the coding interval after b bits have been output by the encoder.

T. Lookabaough et al [9], Selective encryption is a technique to save computational complexity or enable interesting new system functionality by only encrypting a portion of a compressed bit stream while still achieving adequate security. Although suggested in the numbers of specific cases, selective encryption could be much more widely used in consumer electronic applications ranging from mobile multimedia terminals through digital cameras were it subjected to a more thorough security analysis. We describe selective encryption and develop a simple scalar quantizer example to demonstrate the power of the concept, list a number of potential consumer electronics applications, and then describe an appropriate method for developing and analyzing selective encryption for particular compression algorithms.

Maneesh Upmanyu et al [10], proposed Biometric authentication has been widely regarded as the most foolproof - or at least the hardest to forge or spoof [1]. Since the early 1980s, systems of identification and authentication based on physical characteristics have been available to enterprise IT. These biometric systems were slow, intrusive and expensive, but because they were mainly used for guarding mainframe access or restricting physical entry to relatively few users, they proved workable in some high-security situations. Twenty years later, computers are much faster and cheaper than ever.

A.K.Jain et al [11], describes the biometric recognition as a pattern recognition system that operates by acquiring biometric data from an individual, extracting a feature set from the acquired data, and comparing this feature set against the template set in the database. He describes that a biometric system basically comprises of four main modules. These modules are sensor module, Feature extraction module, matcher module and System database module.

Katz. J et al [12], a joint RAC/XOR encryption paradigm for efficient multimedia data protection is presented in this work. By exploiting the structure of entropy coder, the proposed scheme demands very low computation cost and can be easily implemented. It is highly robust against various cryptanalytic attacks. The scheme provides good security and adds less over head. Future directions would be to increase the speed of the randomized binary arithmetic coder.

R.L.Rivest et al [13], presented with the novel property that publicly revealing an encryption key does not thereby reveal the corresponding decryption key. This has two important consequences. First is couriers or other secure means are not needed to transmit keys, since a message can be enciphered using an encryption key publicly revealed by the intended recipient. Only decipher the message, since only he knows the corresponding decryption key. The second is a message can be signed" using a privately held decryption key. Anyone can verify the signature using the corresponding publicly revealed encryption key. Signatures cannot be forged, and a sign cannot later deny the validity of this signature. This has been understandable applications in electronic mail and the electronic funds transfer systems. Taher ElGamal et al [14], proposed a public key encryption Algorithm which is better than the conventional RSA algorithm as the security of the RSA depends on the difficulty of factoring large integers.

III. EXISTING SCHEME

Chosen plain text for Randomized Arithmetic Code are based on same key used to encode many messages, known that using a same key for many messages leads to insecure encryption scheme Every message will be compressed using a new key sequence achieved using a secure pseudorandom sequence generator RAC that uses different key for different messages is not secure under cipher text only attack, But secure in chosen plaintext attack. We prove that RAC that uses different keys for different messages is not secure under ciphertext-only attacks.

Note that we assume that key length is negligible compared to message.

A proof similar to the one in this paper could also be used to prove that KSAC is also insecure against ciphertext-only attacks. Indistinguishability against cipher text-only attacks (or in distinguishability in the presence of an eavesdropper) is the weakest form of security where the adversary can only eavesdrop on cipher texts.

Chosen-plaintext attacks have been proposed for the two methods when the same key is used to encrypt different messages. We first give the definition for security of encryption using AC that is based on the inability of adversary to distinguish between the encryption of one plaintext from the encryption of another. Using this definition, we prove that RAC is insecure even if a new random key is used to compress every message. Our proof assumes that the only eavesdrop on the ciphertext and cannot request encryptions of chosen-plaintexts**.**

IV.PROPOSED SYSTEM

The proposed system, improve the security of Key Based Interval Splitting (KSAC) using Advanced Encryption Standard (AES).

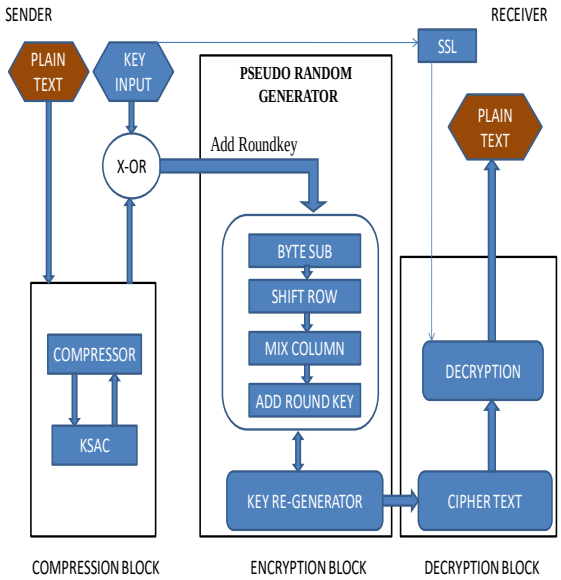

Fig. 1 Proposed System Architecture

A. Compression Block

The plain text is the input to the compressor. This plain text gets compressing by key based interval splitting arithmetic coding. The actual compression is performed by the interval splitting arithmetic coder. In an interval splitting AC, the intervals associated with each symbol, which are continuous in a traditional arithmetic coder, can be split according to a key known both to the encoder and decode data compression, source coding, or bit-rate reduction involves encoding information using fewer bits than the original representation. Compression can be either lossy or lossless. Lossless compression reduces bits by identifying and eliminating statistical redundancy. No information is lost in lossless compression. Lossy compression reduces bits by identifying marginally important information and removing it. The process of reducing the size of a data file is popularly referred to as data compression, although it's formal name is source coding.

Compression is the useful because it helps reduce resources usage, such as data storage space or transmissions capacity. Because the compressed data must be decompressed to use, this extra processing impose computational or other costs through decompression, the situation is far from being a free lunch. Data compression is the subject to a space-time complexity trade-off. For instances, a compression scheme for video may require expensive hardware for the video to be decompressed fast enough to be viewed as it is being decompressed, and the option to decompress the video in the full before watching it may be inconvenient or require additional storage. The design of data compression schemes involve trade-offs among the various factors, including the degree of compression, the amount of distortion introduced the computational resources required to compress and uncompress the data.

B. Encryption Block

In cryptography, encryption is the process of encoding messages in such a way that eavesdroppers or hackers cannot read it, but that authorized parties can. In an encryption schemes, the message or information is encrypted using an encryption algorithm, turning it into an unreadable ciphertext. This is usually done with the use of an encryption key, which specifies how the message is to be encoded. Any adversary that can see the ciphertext should not be able to determine anything about the original message. An authorized party, however, is able to decode the ciphertext using a decryption algorithm that usually requires a secret decryption key, that adversary does not have the access to. For technical reasons, an encryption scheme usually needs a key-generation algorithm to randomly produce keys.

There are two basic types of the encryption schemes: Symmetric-key and public-key encryption. In symmetric-key schemes, the encryption and decryption keys are the same. Thus communicating parties must agree on a secret key before they wish to communicate. In public-key schemes, the encryption key is published for anyone to use and encrypt messages. However, only the receiving party has access to the decryption key and is capable of reading the encrypted messages. Public-key encryption is a relatively recent invention: historically, all encryption schemes have been symmetric-key schemes. The input key sequence is given to pseudorandom generator. The pseudorandom generator operations are bytesub, shiftrow, mixcolumn and addroundkey. Each of the bytes in the matrix changed to another byte in byte substitution. The rows of the matrix are shifted cylindrically to the left. The output of the shift row is multiplied by binary matrix. The add round key is derived by XOR with mixcolumn. Key re-generator gives the key for each key based interval splitting in arithmetic coding. Then it gives as input to the pseudo random generator until the given input splitting finished.

C .Decryption Block

When receiver receives the encrypted message, called cipher text he changes it back to plain text using a decryption key. Chosen plaintext means hacker gains temporary access to the encryption machine. Receiver can encrypt a large number of suitably chosen plaintext and try

to use the resulting cipher text to deduce the key. Chosen cipher text strings of symbols and try to use the results to deduce the key. No information is lost in lossless compression.

Lossy compression reduces the bits by identifying marginally important information and removing it. Finally the original file which was send by the sender is decrypted and decompressed. Each type of data-compression algorithm minimizes redundant data in a unique manner. For example, the Huffman encoding algorithm assigns the code to characters in a file based on how frequently those characters occur.

V.CONCLUSION

The proposed system provides the security on encrypted secret message by Key based interval splitting (KSAC) using Advance Encryption Standard (AES). When sending the message from sender to receiver first the message compressed then encrypted. For encryption AES is used. The compressed encrypted message sends to the receiver. The 128 bit key also sends to the receiver. Receiver receives the both key and encrypted message. Then he decrypts the original message which was s Randomized Arithmetic coding is used to encrypt the message. In our proposed system we used Key based interval Splitting by the algorithm Advance Encryption Standard. It encrypts by the 128 bit key. For encryption we can use 256 and other combination of keys which is compatible for Advance encryption standard. And also we can change AES algorithm for encryption in future. The decryption method depends on the encryption method used.

In existing system Randomized Arithmetic coding is used to encrypt the message. In our proposed system we used Key based interval Splitting by the algorithm Advance Encryption Standard. It encrypts by the 128 bit key. For encryption we can use 256 and other combination of keys which is compatible for Advance encryption standard. And also we can change AES algorithm for encryption in future. The decryption method depends on the encryption method used.

REFERENCES

1. Kim, H. Wen, J. and Villasenor, J.D. (2007) "Secure arithmetic coding",in IEEE Transaction Signal Process,volume.55, no.5, pp.2263–2272.
2. Jiangtao(Gene)Wen,Irvine,S.A.andRinsma-Melchert (1995) "On the in security of arithmetic coding", in Computer Security, volume14, pp.167–180.
3. Langdon, G. and Rissanen, J. (1981) "Compression of black-white images with arithmetic coding",IEEE Trans. Commun., vol. COM-29, no. 6, pp.858–867.
4. Wen, J. Kim,H. and Villasenor, J.D. (2006) "Binary arithmetic coding with key-based interval splitting", in IEEE Signal Processing Lett,volume13, no.2, pp. 69–72.
5. N. K. Ratha, J. H. Connell, and R. M. Bolle, "Enhancing security and privacy in biometrics-based authentication systems," IBM Syst. J., vol.40, no. 3, pp. 614–634, Mar. 2001.
6. Goldwasser, S. and Bellare, M. (1996-2008) "Lecture Notes on Cryptography, Lecture Notes for a Summer Course on Cryptography", Cambridge, MA:MIT.
7. MacKay,D.J.C. (2003) "Information Theory", Inference, and Learning Algorithms Cambridge, U.K.: Cambridge Univ. Press.
8. Moo, P.W. and Wu, X. (1999) "Resynchronization properties of arithmetic coding", in Proc. Data Compression Conf., Snowbird, UT, p. 540.
9. Lookabaough, T. and Sicker, D.C. (2004) "Selective encryption for consumer applications", IEEE Commun. Mag., vol. 42, no. 5, pp. 124–129.
10. Maneesh Upmanyu, Anoop M. Namboodiri, Kannan Srinathan, and C. V. Jawahar, "Blind Authentication: A Secure Crypto-Biometric Verification Protocol", IEEE Transaction on Biometric, Vol. 5, No. 2, June. 2009.
11. A. K. Jain, A. Ross, and S. Prabhakar, "An introduction to biometric recognition," IEEE Trans. Circuits Syst. Video Technol., vol. 14, no. 1,pp. 4–20, Jan. 2004.
12. Katz, J. and Lindell, Y. (2008) 'Introduction to Modern Cryptography', London, U.K.: Chapman & Hall/CRC.
13. R. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," Commun. ACM, vol. 21, no.2, pp. 120–126, 1978.
14. Taher ElGamal "A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms" IEEE transactions on information theory, VOL. IT-31, NO. 4, JULY 1985.
15. C. Shi, S.-Y.Wang, and B. Bhargava, "MPEG video encryption in realtime using secret key cryptography," in Proc. 1999 Int. Conf. Parallel and Distributed Processing Techniques and Applications (PDPTA'99), Las Vegas, NV, Jun./Jul. 28–1, 1999.
16. H. Cheng and X. Li, "Partial encryption of compressed images and video," IEEE Trans. On Signal Process., vol. 48, no. 8, pp. 2439–2451, Aug. 2000.
17. M. Van Droogenbroeck and R. Benedett, "Techniques for a selective encryption of uncompressed and compressed images," in Proc. Advanced Concepts for Intelligent Vision Systems (ACIVS) 2002, Ghent, Belgium, Sep. 9–11, 2002.
18. A. Pommer and A. Uhl, "Selective encryption of wavelet-packet encoded image data," Multimedia Syst. J., vol. 9, no. 3, pp. 279–287, 2003.
19. W. Zheng and S. Lei, "Efficient frequency domain selective scrambling of digital video," IEEE Trans. Multimedia, vol. 5, pp. 118–129, 2003.
20. D. Kundur and K. Karthik, "Video fingerprinting and encryption principles for digital rights management," Proc. IEEE, vol. 92, no. 6, pp. 918–932, Jun. 2004.
21. T. Lookabaough and D. C. Sicker, "Selective encryption for consumer applications," IEEE Commun. Mag., vol. 42, no. 5, pp. 124–129, May 2004.
22. D. S. Taubman and M. W. Marcellin, JPEG2000: Image Compression Fundamentals, Standards and Practice. Norwell, MA: Kluwer Academic, 2002.

AUTHORS PROFILE

Mr.A.Duraisamy received his M.E degree in Computer Science and Engineering from College of Engineering Guindy, Anna University Main Campus Chennai, India in 2010 and B.E degree in Computer Science and Engineering from Anna University, Chennai, India in 2006. He is currently working as a Teaching Fellow in University College of Engineering, Tindivanam, Tamilnadu, India. His areas of interest are: Web Application Security, Image Processing, Networking, Bio-Metrics, Cloud Computing and Peer to Peer Networking. He has published Four Papers in International Journals, Three National Conferences and Life Member in Indian Society for Technical Education.

Mr.M.Sathiyamoorthy received his M.E degree in Computer Science and Engineering from College of Engineering Guindy, Anna University Main Campus Chennai, India in 2007 and B.E degree in Computer Science and Engineering from Madras University, Chennai, India in 2003. He is currently working as a Teaching Fellow in University College of Engineering, Tindivanam, Tamilnadu, India. His areas of interest are: Service Oriented Architecture, Web Services, Web Application Security, Cryptography and Network Security and peer to peer Networking. He has published Two Papers in International Journal. He worked in Tata Consultancy Services Ltd, Chennai as Assistant Systems Engineer for 2.5 years.

Mr.K.Somasundaram received his B.Tech degree in Information Technology from University College of Engineering, Tindivanam in 2012 and presently pursuing M.Tech degree in Information Technology in SNS College of Engineering, Coimbatore. His research interests are Network Security, Cloud Computing and Data Mining.