ISSN No. 0976-5697

Volume 1, No. 4, Nov-Dec 2010

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Dynamic Routing of Data with Secure Cryptographic Techniques

V. Soumya*
Computer Science and Engineering
Aurora's Technological Research Institute (ATRI).
Hyderabad, India.
soumya1108@yahoo.co.in.

S. Durgaprasad
Parallel Computing, Aec
Asst. Prof in IT Dept, MRITS.
Hyderabad, India.
sdp_7@yahoo.com

Ma.Jabbar
In Data Mining
Head of the Department, ATRI.
Hyderabad, India.
jabbar.meerja@gmail.com

T. Nagalakshmi
Web Technologies, Atri
Asst. Prof in CSE Dept, ATRI.
Hyderabad, India.
nlakshmi.cse@gmail.com

Abstract: Security has become one of the major issues for data communication over wired and wireless networks. A dynamic routing algorithm has being proposed that could randomize delivery paths for data transmission. The algorithm is easy to implement and compatible with popular routing protocols, such as the Routing Information Protocol in wired networks and Destination-Sequenced Distance Vector protocol in wireless networks, without introducing extra control messages. Our security-enhanced dynamic routing could be used with cryptography-based system designs to further improve the security of data transmission over networks. In this paper, we present a novel encryption-less algorithm to enhance security in transmission of data in networks. The algorithm uses an intuitively simple idea of a jigsaw puzzle to break the transformed data into multiple parts where these parts form the pieces of the puzzle. The algorithm is designed to provide information-theoretic (that is, unconditional) security by the use of a one-time pad like scheme so that no intermediate or unintended node can obtain the entire data. An authentication code is also used to ensure authenticity of every packet.

Keywords: Data protection, Dynamic routing, Key management, One-time pad, Security algorithm, Security-enhanced data transmission.

I. INTRODUCTION

In the past decades, various security-enhanced measures have been proposed to improve the security of data transmission over public networks. Their common objectives are often to defeat various threats over the Internet, including eavesdropping, spoofing, session hijacking, etc. Security of network communications is arguably the most important issue in the world today given the vast amount of valuable information that is passed around in various networks. The high connectivity of the World Wide Web (WWW) has left the world open., Such openness has resulted in various networks being subjected to multifarious attacks from vastly disparate sources, many of which are anonymous and yet to be discovered. The alternative for security-enhanced data transmission is to dynamically route packets between each source and its destination so that the chance for system break-in, due to successful interception of consecutive packets for a session, is slim. The intention of security-enhanced routing is different from the adopting of multiple paths between a source and a destination to increase the throughput of data Therefore, a dynamic routing algorithm is being proposed to provide security-enhanced data delivery without introducing any extra control messages. This security-enhanced dynamic routing could be used with cryptography-based system designs to further improve the security of data transmission over networks. A typical method for security that is used to prevent data from falling into wrong hands is encryption. Some encryption techniques like RSA [14] which use asymmetric keys, involve algebraic multiplications with very large numbers. The cost that has to be paid in implementing encryption in networks is high owing to this computational complexity. While other techniques like the DES [13] which use symmetric keys are less secure computationally than their asymmetric counterparts. Given the amount of computing power that is available, and considering also the growth of distributed computing, it is possible to break into the security offered by many such existing algorithms. So, any alternative to encryption is welcome so long as the level of security is the same or higher. Also, such an alternative should be more efficient in its usage of resources. In practice, in a computer network, data is transferred across nodes in the form of packets of fixed size. Any form of security required is obtained by implementing cryptographic algorithms at the application level on the data as a whole. Then, the enciphered data is packetized at lower levels (in the OSI model) and sent. Any intruder able to obtain all the packets can then obtain the enciphered data by appropriately ordering the data content of each of these packets. Then, efforts can be made to break the cryptographic algorithm used by the sender. In the process of transmission, if it is possible to prevent any information release as to the structure of the data within the packets, an intruder would know neither the nature of the data being transferred nor the ordering of the content from different packets. This is what our algorithm achieves by using a one-time pad like scheme at the source. The objective of this work is to merge security-enhanced dynamic routing algorithm which is based on distributed

routing information that is widely supported in existing wired and wireless networks with the cryptographic algorithm. We aim at the randomization of delivery paths for data transmission to provide considerably small path similarity (i.e., the number of common links between two delivery paths) of two consecutive transmitted packets. The dynamic routing algorithm is easy to implement and compatible with the algorithm that is proposed in this paper.

The rest of this paper is organized as follows: Section 2 formally defines the related work. In Section 3, we propose a security-enhanced dynamic routing algorithm to randomize the data delivery paths and the cryptographic algorithm that provides more security for data transmission. Section 4 summarizes an analytic study and implementation issues on the proposed algorithm. Section 5 is the conclusion. Section 6 is references.

II. RELATED WORK

Our goal is to merge a distributed dynamic routing algorithm with cryptographic algorithm to improve the security of data transmission. The essential idea in our algorithm is to break the data that is to be transferred into many chunks, which we call parts. These parts when put together form the whole data but only if done so in a particular way, just like in a jigsaw puzzle. The only way of doing so is known to the receiver for whom the data is intended. Any unauthorized node does not have enough information to carry out the right method of obtaining the parts from the packets and joining them, and then knowing it is correct (which is the property of the one-time pad). We have incorporated efficient techniques that enhance the security of the scheme, and at the same time implement the desired features. The concept on which our algorithm hinges heavily is that of the one-time pad. It was first proposed by Vernam in [3]. A formal proof of the perfect-secrecy property of the one-time pad was later provided by Shannon in [4]. Owing to several issues mostly pertaining to key management, the theoretical one-time pad has been tough to implement practically. Numerous attempts have been made but under varying assumptions and conditions.

In [6], it has been argued that unconditional security can be obtained in practice using non-information –theoretically secure methods. This approach maintains that in the practical world, nobody can obtain complete information about a system owing to real-world parameters like noise. Likewise, [7], [8] and [9] provide implementations of one-time pads and unconditional security but under assumptions about the environment and/or adversary. [10] Provides a quantum cryptographic view of one-time pads. Technological and practical limitations constrict the scalability of such methods. Chaotic maps are used to generate random numbers in [11], and these are used to build symmetric encryption schemes including onetime pads. Chaos theoretical methods though providing non-traditional methods of random number generation, are prone to cryptanalysis owing to the still-existent pseudorandom nature, and impose numerous restrictions on data size as is the case with [11]. In this paper, we provide an efficient implementation of the one-time pad without making assumptions or imposing restrictions. In the process, the core issues including key management are addressed and dealt with effectively. Due to its general nature, our algorithm can be deployed in most real-life networks without a fundamental change in the idea.

III. THE ALGORITHMS

The objective of this section is to introduce a distance-vector based algorithm for dynamic routing and the cryptographic algorithm to improve the security of data transmission. The dynamic routing algorithm achieves considerably small path similarity for packet deliveries between a source node and the corresponding destination node. However, the total space requirement would increase to store some extra routing information. The size of a routing table depends on the topology and the node number of a network under discussions. Since the provided distributed dynamic routing algorithm (DDRA) is a distance-vector-based routing protocol for intradomain systems, the number of nodes is limited, and the network topology is hardly fully connected. Hence, the increase of the total space requirement is considerably small.

A. *Distributed Dynamic Routing Algorithm*

The DDRA proposed in this paper consists of two parts:

[a] a randomization process for packet deliveries and
[b] maintenance of the extended routing table.

Randomization Process

Consider the delivery of a packet with the destination t at a node N_i. In order to minimize the probability that packets are eavesdropped over a specific link, a randomization process for packet deliveries shown in Procedure 1 is adopted. In this process, the previous nexthop h_s for the source node s is identified in the first step of the process (line 1). Then, the process randomly picks up a neighboring node in C^{Ni}_t excluding h_s as the nexthop for the current packet transmission. The exclusion of h_s for the nexthop selection avoids transmitting two consecutive packets in the same link, and the randomized pickup prevents attackers from easily predicting routing paths for the coming transmitted packets.

Procedure 1 RANDOMIZEDSELECTOR (s; t; pkt)

[i] Let h_s be the used nexthop for the previous packet delivery for the source node s.
[ii] if h_s€ C^{Ni}_t then
[iii] if $\left| C^{Ni}_t \right| > 1$ then
[iv] Randomly choose a node x from $\{C^{Ni}_t - h_s\}$ as a nexthop, and send the packet pkt to the node x.
[v] $h_s \leftarrow x$, and update the routing table of N_i.
[vi] else
[vii] Send the packet pkt to h_s.
[viii] end if
[ix] else
[x] Randomly choose a node y from C^{Ni}_t as a nexthop, and send the packet pkt to the node y.
[xi] $h_s \leftarrow y$, and update the routing table of N_i.
[xii] end if

Routing Table Maintenance

Let every node in the network be given a routing table and a link table. We assume that the link table of each node is constructed by an existing link discovery protocol, such as the Hello protocol in [1]. On the other hand, the construction and maintenance of routing tables are revised based on the

well-known Bellman-Ford algorithm [2] and described as follows:

Initially, the routing table of each node (e.g., the node N_i) consists of entries $\{(N_j, W_{Ni,Nj}, C^{Ni}_{Nj}= \{N_j\}, H^{Ni}_{Nj} =\Phi)\}$ where N_j€Nbr_i and $W_{Ni,Nj}= w_{Ni,Nj}$. By exchanging distance vectors between neighboring nodes, the routing table of N_i is accordingly updated. Note that the exchanging for distance vectors among neighboring nodes can be based on Predefined interval. The exchanging can also be triggered by the change of link cost or the failure of the link/node. In this paper, we consider cases when N_i receives a distance vector from a neighboring node N_j. Each element of a distance vector received from a neighboring node N_j includes a destination node t and a delivery cost $W_{Nj,t}$ from the node N_j to the destination node t. The algorithm for the maintenance of the routing table of N_i is shown in Procedure 2, and will be described below.

Procedure 2 DVPROCESS {t, $W_{Nj,t}$}

[i] if the destination node t is not in the routing table then
[ii] Add the entry $(t,(w_{Ni,Nj} + W_{Nj,t}), C^{Ni}_t=\{ N_j \}, H^{Nj}_t= \Phi)$
[iii] else if $(w_{Ni,Nj} + W_{Nj,t})< W_{Ni,t}$ then
[iv] $C^{Ni}_t \leftarrow N_j$ and N_j s marked as the minimal-cost nexthop.
[v] $W_{Ni,t}\leftarrow(w_{Ni,Nj} + W_{Nj,t})$
[vi] for each node N_k € Nbr_i except N_j do
[vii] if $W_{Nk,t} < W_{Ni,t}$ then
[viii] $C^{Ni}_t \leftarrow C^{Ni}_t U \{N_k\}$
[ix] end if
[x] end for
[xi] Send $(t,W_{Ni,t})$to each neighboring node N_k € Nbr_i.
[xii] else if $(w_{Ni,Nj} + W_{Nj,t})< W_{Ni,t}$ then
[xiii] if (N_j € C^{Ni}_t) then
[xiv] if N_j was marked as the minimal-cost nexthop then
[xv] $W_{Ni,t} \leftarrow MIN_{Nk € Nbri}(w_{Ni,Nk}+ W_{Nk,t})$
[xvi] $C^{Ni}_t\leftarrow \Phi$
[xvii] for each node N_k € Nbr_i do
[xviii] if $W_{Nk,t} < W_{Ni,t}$ then
[xix] $C^{Ni}_t \leftarrow C^{Ni}_t U \{N_k\}$
[xx] end if
[xxi] end for
[xxii] Send $(t,W_{Ni,t})$ to each neighboring node N_k € Nbr_i.
[xxiii] else if $W_{Ni,t}> W_{Ni,t}$ then
[xxiv] $C^{Ni}_t \leftarrow C^{Ni}_t – \{N_j\}$
[xxv] end if
[xxvi] else if ($N_j \notin C^{Ni}_t$) Λ $W_{Nj,t}< W_{Ni,t}$ then
[xxvii] $C^{Ni}_t \leftarrow C^{Ni}_t U \{N_j\}$
[xxviii] end if
[xxix] end if

The cryptographic algorithm:

In this algorithm, we use the concept of Message Authentication Code (MAC) as suggested in [5] to authenticate messages. For a packet of data, the MAC is calculated as a function of the data contents, the packet sequence number and a secret key known only to the sender and the receiver, and then it is appended to the packet. On receiving a packet, the receiver first computes the MAC using the appropriate parameters, and then performs a check with the MAC attached to the packet. If there is no match, then the receiver knows that the packet has been tampered with. Let the size of a packet in a network be denoted as PS. PS has a value of 1024-bits or 4096-bits typically. The prerequisite of the algorithm is the knowledge for the sender and the receiver only of a number P, exchanged a priori, of size *k**PS, where *k* is a natural number. We can consider P in terms of blocks of size PS each, as $P_1, P_{2,} \ldots\ldots P_k$. Thus P is the number obtained by the concatenation of the P_i blocks for *i* from 1 to *k*, that is, P is $P_1, P_{2,}\ldots\ldots P_k$

In this algorithm, we only consider *k-1* parts of a data at a time, where each part is of any size less than or equal PS-2 (a detailed analysis of the algorithm is presented in the next section). If the entire data is not covered in these *k-1* parts, then the algorithm can be repeated by considering the next *k-1* parts and so on. Also, a random number of size PS is required to be generated for every *k-1* blocks processed. Denote this random number as R. The steps at the sender's end of the algorithm are as follows:

S (D)
[i] Tear the data D into N parts arbitrarily. Consider the first k-1 parts of D. Call them $D_1,D_{2,}.....D_{k-1}$ Prefix and suffix each part by the binary digit '1'.
[ii] Perform the operation $\lvert D_i \rvert$ (XOR) P_i for all i from 1 to k-1. Denote them as $D'_1,D'_{2,}.....D'_{k-1}$.
[iii] Form D'_k as D'_k = R (XOR) P_k .
[iv] Perform transform (P, R).
[v] Generate a new random number and assign it to R.
[vi] Repeat steps 1, 2, 3, 4, 5 for the next k-1 blocks of data, and so on until all N parts are processed.

Now the packets actually transferred are formed from the D'_i blocks, packet sequence number and the MAC (calculated as described earlier). At the receiver's end, the steps are as follows:

R (D')
[i] Perform a check on the MAC for each packet. If satisfied, GOTO next step.
[ii] Order packets according to the packet sequence number.
[iii] For each group of k packets perform the following:
- Perform P_i (XOR) D'_i for all i from 1 to k-1
- Perform P_k (XOR) D'_k and obtain R.
- Remove the leading and trailing '1' of all the values obtained from the previous two steps.
- Perform transform (P, R).
- continue.

The algorithm for the operation transform (P, R) is as follows:

Transform (P,R)
[i] Set $P_i \leftarrow P_i$ (XOR) R for all i from 1 to k-1.
[ii] Set $P_k \leftarrow P_k * R$.

For the transfer of the next data, the following is done. The sender knows the number of parts of the previous data and hence knows the value of N % k, where % denotes the 'modulo' operation. Now, this value subtracted from k provides the P_is unused in the last run of the loop in the algorithm. The next data to be transferred is first broken into parts as before. For the first run of the loop in the algorithm, the first k-1- (N % k) (here, this is the old value of N) parts are only considered and the run executed. For the remaining runs, we consider k-1 parts as before, and the algorithm is continued. This is done for all subsequent data transfers between the sender and the same receiver.

IV. IMPLEMENTATION ISSUES

A. Discussion

The essential idea in the algorithm is to split the data into pieces of arbitrary size (rather, the size is not arbitrary since it is bound by 0 and PS-2 but is allowed to take any value between the limits), transmit the pieces securely, and provide a mechanism to unite the pieces at the receiver's end. Towards this, the role of the number P is to provide a structure for the creation of the jigsaw puzzle. This structure masks the pieces as well as protects the data within. The structure is changed periodically with the knowledge of both the sender and the receiver to prevent an adversary gaining information about it. The (XOR) function is reversible and can be easily performed by the receiver since he knows the secret number *P*. However, an adversary without knowing *P* cannot obtain any additional information. This is because of the following.

Given a cipher text C of length PS and a random secret key *P* of length PS, the probability of any particular key is the same. If it is possible to guess the message M such that C = M (XOR) *P*, then it is possible to determine the value of *P*. Since every secret key *P* is equally likely, there is no way of guessing which of the possible messages of length PS or less was sent. In other words, let us assume that the adversary has as much knowledge of the cryptographic mechanism as the receiver does (except, of course, knowledge of the secret key *P*). Now, the only knowledge that an adversary has about both M and *P* is that one is a binary string of length PS while the other is of length less than PS. Then, knowledge of the cipher text C does not provide the adversary with any more information pertaining to either M or *P*. This is the information-theoretic security property of a one-time pad. The value D'_i obeys this one-time pad property since both P_i and D_i are random numbers as far as the adversary is concerned. Thus, it is impossible for an adversary to get any more information given this value. Thus, all of D_i remain completely unknown to the adversary.

Now, after *k* parts are processed, it is not possible to repeat the values P_1, P_2,……. . Else, by manipulations due to the reversible nature of (XOR), some information might be leaked to the adversary. Therefore, the next set of values has to be changed, and towards this a random number of equal size is used. Also, this random number is to be conveyed to the receiver without any adversary knowing its value. Hence, we introduce the random number as the *k*th part. This random number is used to calculate the next P to be used. Since the initial P was a secret for the adversary, and so is the random number, the new value of P is also a secret. Thus, the security of the data transmission is ensured. In our model of the one-time pad, data is of effective size less than (*k-1*)*PS bits and the key 'xor'ed with the data at each stage is the number P_1, P_2,……., P_{k-1}. The random number R is used as an input to a function (namely *tran! sform()*) to generate the key for the next run. The value P_k is used to securely convey the random number R, which is generated by the sender, to the receiver. The security offered by the algorithm is the same as that provided by one-time pad - information-theoretic security. This is evident from the fact that the jigsaw is structured by an embedding of the data in the key using the 'xor' operation. However, unlike the one-time pad, keys used in our algorithm are not completely uncorrelated since the next key is formed as a function of the current key and a random value. Thus, under conditions of information leakage, the security offered by our approach fails while the one-time pad continues to offer the same level of security to the unexposed data. The nature of security under such conditions has not been studied here. For private-key message authentication purposes, another secret key might be needed. However, since any PS-size block of P cannot be guessed from its first use, it is possible to use any of them or parts of it (in case a smaller key is desired) for message authentication by the calculation of MAC. This information is also to be exchanged a priori along with the number P as well as the value of *k*. For different sender-receiver pairs, different secret numbers P are needed. It is necessary to take care to see that this indeed is the case.

B. Implementation Issues

In our algorithm, for the transfer of N parts of data, there are N additions performed. Apart from these operations, the value of P is changed *floor* (N/k) times. At each change, there are k-1 additions and 1 multiplication. Hence, in total our algorithm requires N + *floor* (N/k)*(k-1) additions and *floor* (N/k) multiplications on PS-bit blocks for the transfer of N parts of data. A comparison with other encryption algorithms is valid only when the application of those algorithms is for secure data transmission. From this viewpoint, the statistics presented above compare favorably with respect to encryption algorithms like Advanced Encryption Standard (AES) [12]. In AES, for input block size of 128-bits and key length of 128-bits, there are atleast 11 'xor' operations apart from matrix multiplications, table lookups and vector shifts. Also, our algorithm does not transform the data except for the 'xor' operation. This operation and its inverse can be easily computed. Hence, as compared to encryption algorithms like AES, DES [13] and RSA [14], the data processing time is least for our algorithm. Our algorithm lends itself to parallelism in implementation in software as well as dedicated hardware. The execution of the operation transform (P, R) should follow the processing of k blocks. This sequentially cannot be avoided. However, the processing of the blocks can be done in a parallel manner. In principle, the algorithm can be implemented efficiently using k 'xor' gates, as shown in figure 1. Here, the first three cycles of 'xor's are depicted with the respective inputs and outputs. If number of gates is also a constraint, for best performance, the value of k should be decided accordingly.

Security of the Jigsaw

The security offered by the algorithm is the same as that provided by one-time pad -information-theoretic security. This is evident from the fact that the jigsaw is structured by an embedding of the data in the key using the XOR operation. However, unlike the one-time pad, keys used in our algorithm are not completely uncorrelated since the next key is formed as a function of the current key and a random value.

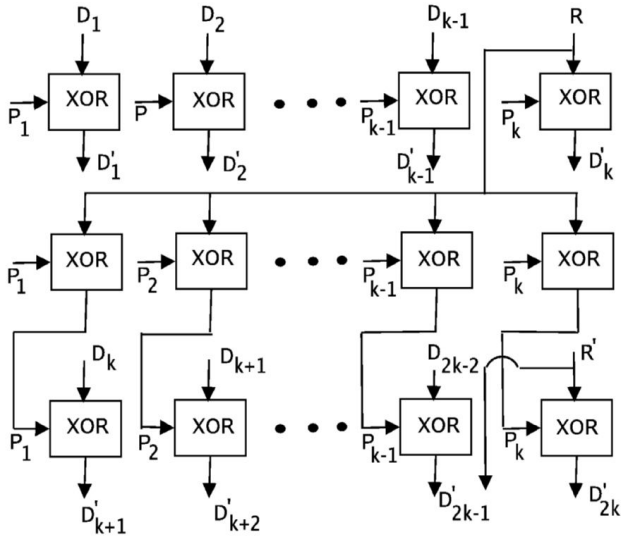

Figure 1: Schematic for parallel implementation of the 'jigsaw' algorithm

Thus, even if some information about the key or the random number of a round or even about the nature of the data is revealed, the security offered by our approach fails while the theoretical one-time pad continues to offer the same level of security to the unexposed data.

V. CONCLUSION

This paper has proposed a security-enhanced dynamic routing algorithm based on distributed routing information widely supported in existing networks. The randomization process provides security for data in the wired or wireless networks. This is the advantage of the distributed dynamic routing algorithm. But to provide more security for the data in the networks, our security-enhanced dynamic routing is merged with cryptography- based system designs to further improve the security of data transmission over networks. In this paper, we adopt a jigsaw approach to secure data transfer in networks. The data to be sent is broken into parts of arbitrary sizes. Enough information is provided efficiently and securely to enable the receiver to solve the jigsaw puzzle. The transfer of the parts is done securely without leaking any information to the adversary regarding the data. The concept of the one-time pad is implemented in the course of the algorithm resulting in information-theoretic security of data transfer. Also, flexibility in the form of a means of control is provided in the algorithm to monitor and check the overhead resulting because of the data expansion due to the arbitrary splitting.

We have illustrated a method of implementing message authentication by private key without the exchange of any more information. We believe that the paradigm of "jigsaw puzzle" as illustrated in this paper is novel and any research towards a sturdier implementation would benefit the community. In this paper, we have presented our implementation of the "jigsaw" paradigm. There could be other implementations that are more efficient than ours.. We have also suggested a technique to make the algorithm secure against cryptanalytic attacks in the eventuality when the nature of the data is revealed. The inclusion of this has been proved to still be substantially more efficient than encryption algorithms. Moreover, our realization of the "jigsaw" paradigm has been designed to support a parallel implementation catering to future technological advancements.

VI. ACKNOWLEDGMENT

Authors would like to thank the anonymous reviewers for their valuable comments which helped to improve the clarity of the paper.

VII. REFERENCES

[1] D. L. Mills, *DCN Local-Network Protocols*, Request for comments (RFC 891), Dec. 1983.
[2] T.H. Cormen, C.E. Leiserson, and R.L. Rivest, *Introduction to Algorithms*. MIT Press, 1990.
[3] Gilbert S. Vernam, "Cipher printing telegraph systems for secret wire and radio telegraphic communications", *J. American Institute for Electrical Engineers*, 55:109-115, 1926.
[4] Claude E. Shannon, "Communication Theory of Secrecy Systems", *Bell Systems Technical J.*, 28:656-715, 1949.
[5] Ronald L. Rivest, "Chaffing and Winnowing: Confidentiality without encryption", Apr 1998, http://theory.lcs.mit.edu/ rivest/chaffing.txt
[6] Ueli Maurer, "Information-Theoretic Cryptography (Extended Abstract)", *Proc. 19th Annual International Cryptology Conference,* LNCS, 1666:47-64, 1999.
[7] Christian Cachin, and Ueli Maurer, "Unconditional Security Against Memory-Bounded Adversaries", *Advances in Cryptology: CRYPTO* '97, LNCS, 1294: 292-306, 1997.
[8] Ueli Maurer, and Stefan Wolf, "Unconditionally secure key agreement and the intrinsic conditional information", *IEEE Transactions on Information Theory*, 45- 2:499-514, 1999.
[9] Manuel Gunter, Marc Brogle, and Torsten Braun, "Secure Communication: a New Application for Active Networks", *International Conference on Networking (ICN)*, 2001.
[10] G.Brassard and C.Crepeau, "25 years of Quantum cryptography", *SIGACT News*, 27-3:13-24, 1996.
[11] Jiri Fridrich, "Symmetric Ciphers Based On Two-Dimensional Chaotic Maps", *Intl. J. of Bifurcation and Chaos (IJBC) in Applied Sciences and Engineering*, 8-6, 1998.
[12] Federal Information Processing Standards Publication 197, *Advanced Encryption Standard*, Nat'l Institute of Standards and Technology, 2001.
[13] Federal Information Processing Standards 46, *Data Encryption Standards*, Nat'l Bureau of Standards, 1977.
[14] Ronald L. Rivest, Adi Shamir, and Leonard Adleman, "A Method for Obtaining Digital Signatures and Public Key Cryptosystems", *Communications of the ACM*, 21-2:120-126, 1978.

AUTHORS

V. SOUMYA received her B.Tech degree in Computer Science and Engineering from Progressive engineering college, JNTU, Hyderabad in the year 2008 and at present she is pursuing M.Tech from the year 2008 and about to complete her M.Tech in Computer Science and Engineering at Aurora's Technological Research Institute, JNTU, Hyderabad.

S. DURGAPRASAD received his B.Tech degree in Computer Science and Engineering in the year 2005 from Newton's Institute of Engineering, Macherla , JNT University, Hyderabad and also pursuing his M.Tech in PARALLEL COMPUTING from 2008 and about to complete his M.Tech at Aurora's Engineering college, JNT University, Hyderabad. At present working as Assistant Professor in IT department at MallaReddy Institute of Technologies, Hyderabad. His research interest includes parallel computing and Security.

MA. JABBAR obtained his Bachelors in Computer Science Engineering from SRTM University Nanded and Masters from JNTUK. He is doing his research in Data Mining. He is having more than 10 years of Teaching Experience .Currently he is the Head of CSE Dept. at Aurora's Technological and Research institute Hyderabad. His research interests include data mining and advanced database systems.

T. NAGA LAKSHMI received her B.Tech degree in Computer Science and Information Technology in the year 2005 from GURU NANAK ENGINEERING COLLEGE, JNT University, Hyderabad and also received her M.Tech degree in WEB TECHNOLOGIES 2006-2008 from Aurora's Technological and Research Institute, JNT University, Hyderabad. At present working as Assistant Professor in Aurora Technological and Research Institute. Her research interest includes Data Mining, Web Mining, and Security.