Broadcast Encryption with Short Cipher Texts using Adhoc on Demand Distance Vector (AODV)

R. Rameswari

Abstract*—Traditional broadcast encryption (BE) schemes allow a sender to securely broadcast to any subset of members but require a trusted party to distribute decryption keys. Group key agreement (GKA) protocols enable a group of members to negotiate a common encryption key via open networks so that only the group members can decrypt the ciphertexts encrypted under the shared encryption key, but a sender cannot exclude any particular member from decrypting the ciphertexts. It proposes Adhoc on demand Distance Vector (AODV).here we want to form a group in a particular distance using distance vector. AODV protocols enable a group of members to negotiate a common encryption key via open networks so that only the group members can decrypt the ciphertexts encrypted under the shared encryption key. But here no require a trusted party to distribute decryption keys. Some features added to DVR algorithm is known as Destination sequence distance vector (DSDV) routing algorithm. This algorithm provides good level of security and performance.

Key words*—Broadcast encryption (BE), group key agreement (GKA),contributory broadcast encryption (ConBE), Adhoc on demand Distance Vector (AODV), Destination sequence distance vector (DSDV)

I. INTRODUCTION

With the increase in technology advancement in communication technologies, there is an increasing demand of versatile cryptographic primitives to protect group communications and computation platforms. These new platforms include instant-messaging tools, collaborative computing, mobile ad hoc networks and social networks. These new applications call for cryptographic primitives allowing asunder to securely encrypt to any subset of the users of the services without relying on a fully trusted dealer. Broadcast encryption (BE) [1] is a well-studied primitive intended for secure group-oriented communications. It allows a sender to securely broadcast to any subset of the group members. Nevertheless, a BE system heavily relies on a fully trusted key server who generates secret decryption keys for the members and can read all the communications to any members. Group key agreement (GKA) is another well-understood cryptographic primitive to secure group-oriented communications. A conventional GKA [2] allows a group of members to establish a common secret key via open networks. However, whenever a sender wants to send a message to a group, he must first join the group and run a GKA protocol to share a secret key with the intended members. More recently, and to overcome this limitation, With the introduction of asymmetric GKA [3], in which only a common group public key is negotiated and each group member holds a different decryption key.

Revised Version Manuscript Received on March 27, 2016.

R. Rameswari, PG scholar, Department of Computer Science and Engineering, SRM University, Kattankulathur (Chennai). India.

However, neither conventional symmetric GKA nor the newly introduced asymmetric GKA allow the sender to unilaterally exclude any particular member from reading the plaintext. Hence, it is essential to find more flexible cryptographic primitives allowing dynamic broadcasts without a fully trusted dealer. This paper investigates a close variation of the above mentioned problem of one-round group key agreement protocols and focuses on “how to establish a confidential channel from scratch for multiple parties in one round”. We provide a short overview of some new ideas to solve this variation. Asymmetric GKA Observe that a major goal of GKAs for most applications is to establish a confidential broadcast channel among the group. We investigate the potentiality to establish this channel in an asymmetric manner in the sense that the group members merely negotiate a common encryption key (accessible to attackers) but hold respective secret decryption keys. We introduce a new class of GKA protocols which we name asymmetric group key agreements (ASGKAs), in contrast to the conventional GKAs. A trivial solution is for each member to publish a public key and withhold the respective secret key, so that the final ciphertext is built as a concatenation of the underlying individual ones. However, this trivial solution is highly inefficient: the ciphertext increases linearly with the group size; furthermore, the sender has to keep all the public keys of the group members and separately encrypt for each member. We are interested in nontrivial solutions that do not suffer from these limitations. Group key agreement (GKA) is another well-understood cryptographic primitive to secure group-oriented communications. A conventional GKA allows a group of members to establish a common secret key via open networks. However, whenever a sender wants to send a message to a group, he must first join the group and run a GKA protocol to share a secret key with the intended members. More recentlyintroduced asymmetric GKA in which only a common group public key is negotiated and each group member holds a different decryption key. However, neither conventional symmetric GKA nor the newly Introduced asymmetric GKA allow the sender to unilaterally exclude any particular member from reading the plaintext1. Hence, it is essential to find more flexible cryptographic primitives allowing dynamic broadcasts without a fully trusted dealer.

II. EXISTING SYSTEM

Group key agreement (GKA) is another well-understood cryptographic primitive to secure group-oriented communications. A conventional GKA allows a group of members to establish a common secret key via open networks. However, whenever a sender wants to send a message to a group, he must first join the group and run a GKA protocol to share a secret key with the intended

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.

members. More recently, and to overcome this limitation, introduced asymmetric GKA, in which only a common group public key is negotiated and each group member holds a different decryption key. However, neither conventional symmetric GKA nor the newly introduced asymmetric GKA allow the sender to unilaterally exclude any particular member from reading the plaintext. Hence, it is essential to find more flexible cryptographic primitives allowing dynamic broadcasts without a fully trusted dealer. The drawbacks of existing system are (1) Need a fully trusted third party to set up the system and (2) Existing GKA protocols cannot handle sender/member changes efficiently.

III. PROPOSED SYSTEM

In the proposed system, we have to create a node in a network using distance vector. If you want to form a group in a particular distance after we want to create a group in a cluster formation. After that communication has been established between the trusted parties for safe communication between the network.

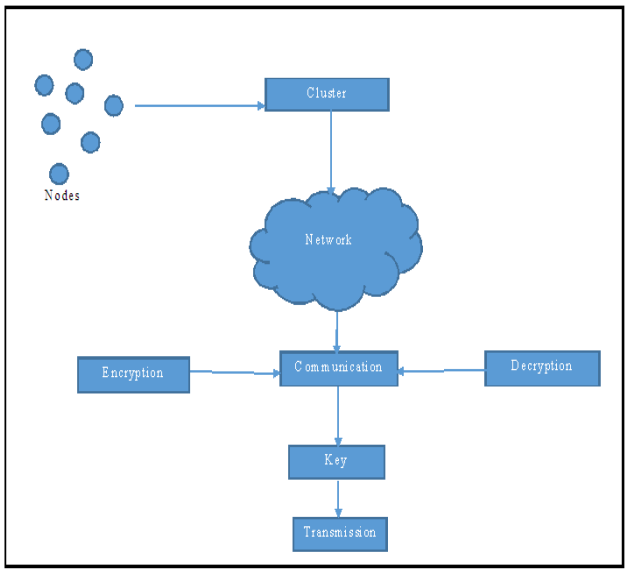

Fig 3.1 block diagram of proposed system.

IV. NETWORK TOPOLOGY CREATION

To be able to run a simulation scenario, a network topology must first be created. In ns2, the topology consists of a collection of nodes and links. Before the topology can be set up, a new simulator object must be created at the beginning of the script. The simulator object has member functions that enable creating the nodes and the links, connecting agents etc. All these basic functions can be found from the class Simulator. When using functions belonging to this class, the command begins with "$ns", since ns was defined to be a handle to the Simulator object

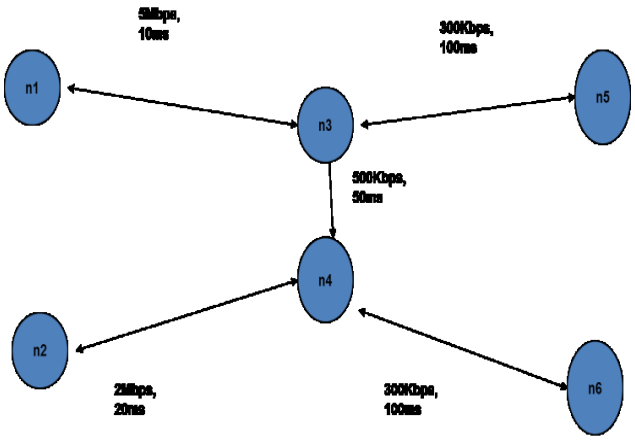

Fig 4.1 creation of network

V. TRANSPORT CONNECTION

The Transmission Control Protocol (TCP) is one of the two original core protocols of the Internet protocol suite (IP), and is so ubiquitous that the entire suite is often called TCP/IP. TCP provides reliable, ordered, error-checked delivery of a stream of octets between programs running on computers connected to an intranet or the public Internet. Applications that do not require the reliability of a TCP connection may instead use the connectionless UDP.

VI. GENERATING TRAFFIC

Random traffic connections of TCP and CBR can be setup between mobile nodes using a traffic-scenario generator script. This traffic generator script is available under ~ns/indep-utils/cmu-scen-gen and is called cbrgen.tcl. It can be used to create CBR and TCP traffics connections between wireless mobile nodes. In order to create a traffic-connection file, we need to define the type of traffic connection (CBR or TCP), the number of nodes and maximum number of connections to be setup between them, a random seed and incase of CBR connections, a rate whose inverse value is used to compute the interval between CBR packets

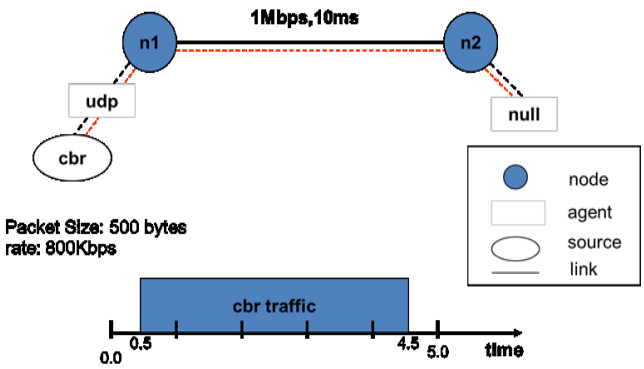

Fig 6.1 generation of traffic

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.

VII. MODULE DESCRIPTION

Module description which consist of 3 modules which are as follows:

1. Network Configuration
2. Cluster formation
3. Communication establishment

1. Network Configuration:

Network formation is how and why networks take particular forms. In many networks, the relation between nodes is determined by the choice of the participating players involved, not by an arbitrary rule. A "strategic" modeling of network requires defining a network's costs and benefits and predicts how individual preferences become outcomes. A strategic network formation requires that individuals create relations that are beneficial and drop those that are not. One of the most well-known examples in this context is the marriage network of sixteen families in Florence, which showed how the Medici family gained power and took control of Florence by creating a high number of inter-marriages with the other families.[1] "Thus, decisions about profitable relations are not a situation of choice, but a situation of strategic interaction – an aspect that is best covered by game theory Network configuration accommodates flexible and modular construction of different node definitions within the same base Node class. For instance, to create a mobile node capable of wireless communication, one no longer needs a specialized node creation command, e.g., [] dsdv-create-mobile-node; instead, one changes default configuration parameters, such as $ns node-config –adhocRoutingdsdv before actually creating the node with the command: $ns node. Together with routing modules, this allows one to combine "arbitrary" routing functionalities within a single node without resorting to multiple inheritance and other fancy object gimmicks. The functions and procedures relevant to the new node APIs may be found in ~ns/tcl/lib/ns-node.tcl. The node configuration interface consists of two parts. The first part deals with node configuration, while the second part actually creates nodes of the specified type.

2. Cluster formation:

Group of independent servers (usually in close proximity to one another) interconnected through a dedicated network to work as one centralized data processing resource. Clusters are capable of performing multiple complex instructions by distributing workload across all connected servers.

Clustering improves the system's availability to users, its aggregate performance, and overall tolerance to faults and component failures. A failed server is automatically shut down and its users are switched instantly to the other server. Nodes which are in a communication distance and frequency range are selected to form a single cluster where the head will be selected through the residual energy of the node. If the node is going to communicate the sender have to check the receiver in a same cluster or not. So the cluster information should be shared to the cluster head by the router for analysis and report of the cluster.

```
static class SRMAgentClass : public TclClass
{
public:
        SRMAgentClass() : TclClass("Agent/SRM") {}
        TclObject* create(int, const char*const*)
        {
                return (new SRMAgent());
        }
} class_srm_agent;
```

3. Communication establishment:

Two-way process of reaching mutual understanding, in which participants not only exchange (encode-decode) information, news, ideas and feelings but also create and share meaning. In general, communication is a means of connecting people or places. In business, it is a key function of management--an organization cannot operate without communication between levels, departments and employees.

If the user wants to send a message, the trial packet will be generated first and sent to the all possible receiver which are next to the sender. If the receiver matches, the route is not needed. If not, the process will be continued till the receiver found. The communication will be done through the tcl protocol.

```
#Setup a TCP connection
settcp [new Agent/TCP/Newreno]
$ns attach-agent $n(5) $tcp
set sink [new Agent/TCPSink/DelAck]
$ns attach-agent $n(1) $sink
$ns connect $tcp $sink
$tcp set fid_ 1
$tcp set window_ 2000
$tcp set packetSize_ 240

#Setup a FTP over TCP connection
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ftp set type_ FTP
$ns at 1.0 "$ftp start"
$ns at 10.8 "$ftp stop"
```

VIII. PROTOCOLS AND PROPOGATION MODEL

8.1 APPLICATION LAYER – FTP

File Transfer Protocol (FTP) is a standard network protocol used to transfer files from one host to another host over a TCP-based network, such as the Internet.FTP is built on a client-server architecture and uses separate control and data connections between the client and the server.[1] FTP users may authenticate themselves using a clear-text sign-in protocol, normally in the form of a username and password, but can connect anonymously if the server is configured to allow it. For secure transmission that hides (encrypts) the username and password, and encrypts the content, FTP is often secured with SSL/TLS ("FTPS"). SSH File Transfer Protocol ("SFTP") is sometimes also used instead, but is technologically different.

8.2 TRANSPORT LAYER – TCP

TCP offers connection between any two applications and it guarantees reliable data delivery. The TCP and IP are the two important protocols and they are named as TCP/IP

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.

protocol suite. TCP is a peer-to-peer, connection oriented protocol, and there is no master-slave relation available. There are two main functions of transport layer in internet namely check summing and multiplexing/demultiplexing of data.

8.3 NETWORK LAYER –DSDV

In ad-hoc networks Destination distance vector was used at first and then On-demand distance vector (AODV) protocol was used. The performance of DVR is not so good, due to count-to-infinity problems. Because every node exchanges with its neighbour table at regular periods. Changes that take place at a node in network slowly propogateand thus it is not a better protocol to apply.

Some features added to DVR algorithm is known as Destination sequence distance vector (DSDV) routing algorithm. DSDV is like traditional distance vector routing technique. It is also called as "Bellman-Ford routing algorithm". The DSDV has few modified procedures to reduce routing loops.

Two features appended with DVR are namely,

***1 .Sequence Numbers*:**

Every routing advertisement should come with a sequence number. These sequence numbers helps to apply the advertisements in a proper order. It helps to avoid looping.

2. Damping:

If the transient changes in network topology prolongs for negligible time then it may not degrade the performance of routing mechanisms. A node waits if changes are unstable and this waiting time will depend on the time taken between first and the best advertisement of routing path to a definite destination node. Important operations of DSDV

a)Each router in the network collect information from all its neighbours.

b) After gathering information node searches for a shortest path to route the packet.

c) A new routing table is generated.

d) Then this router will broadcast this table to its neighbours.Due to this function the other node (neighbours) are triggered to recomputed their respective routing table.

e) This process continues till the routing information becomes stable.

IX. TWO RAY GROUND REFLECTION MODEL

A single line-of-sight path between two mobile nodes is seldom the only means of propagation. The two-ray ground reflection model considers both the direct path and a ground reflection path. It is shown that this model gives more accurate prediction at a long distance than the free space model. The received power at distance is predicted by

$$Pr(d) = PtGtGrht2hr2/d4L \quad \ldots. \quad (1)$$

whereht and hr are the heights of the transmit and receive antennas respectively.

Note that the original equation assumes L=1. To be consistent with the free space model, L is added here. The above equation shows a faster power loss. However, the two-ray model does not give a good result for a short distance due to the oscillation caused by the constructive and destructive combination of the two rays. Instead, the free space model is still used when d is small. Therefore, a cross-over distance dc is calculated in this model. When d<dc, Eqn. (1) is used. When d>dc, Eqn. (2) is used. At the cross-over distance, Eqns. (1) and (2) give the same result. So dc can be calculated as

$$d= (4\pi hthr)/\lambda \quad \ldots. \quad (2)$$

Similarly, the OTcl interface for utilizing the two-ray ground reflection model is as follows.

```
$ns_ node-config -propType
Propagation/TwoRayGround
```

Alternatively, the user can use

```
set prop [new Propagation/TwoRayGround]
$ns_ node-config -propInstance $prop
```

X. SIMULATION RESULTS

The simulation are done using Cygwin NS2 tool for broadcast encryption. To be able to run a simulation scenario, a network topology must first be created. In ns2, the topology consists of a collection of nodes and links. Before the topology can be set up, a new simulator object must be created at the beginning of the script.It can be used to create CBR and TCP traffics connections between wireless mobile nodes. In order to create a traffic-connections. Network configuration accommodates flexible and modular construction of different node definitions within the same base Node class. For instance, to create a mobile node capable of wireless communication networks for transport. In ad-hoc networks Destination distance vector was used at first and then On-demand distance vector (AODV) protocol was used. The simulation are done for creating a network, establishment of transport and cluster formation are shown below

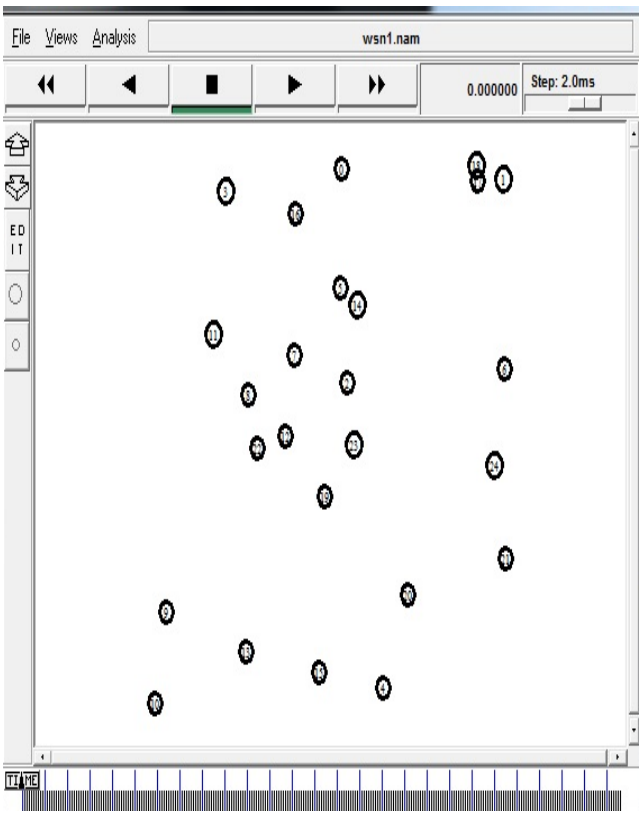

Fig 10.1 simulation of network node creation.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.

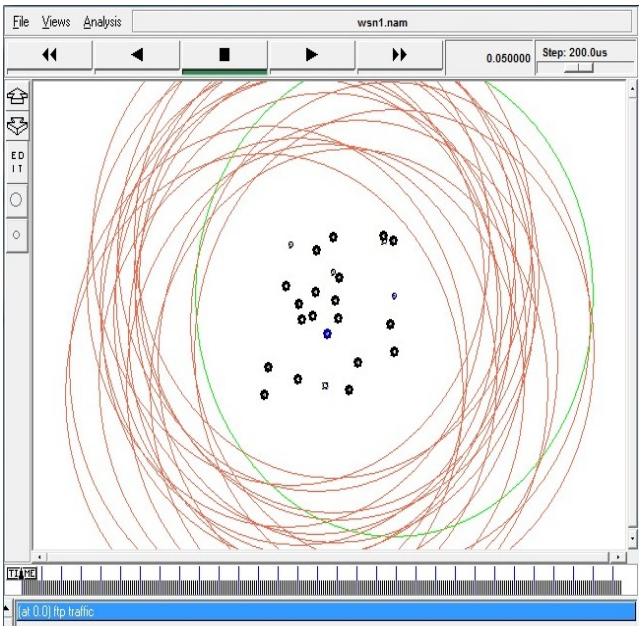

Fig10.2 simulation of cluster formation

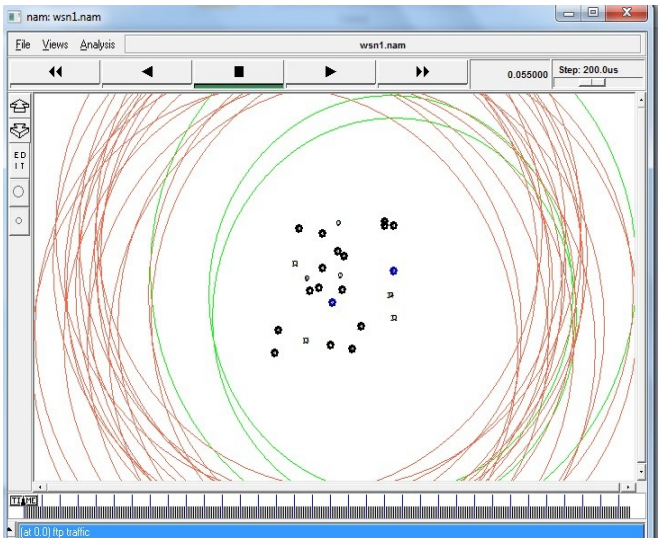

Fig10.3 simulation of communication establishment

XI. CONCLUSION

In this paper, the efforts are taken toAdhoc on demand Distance Vector (AODV) topology. Here we want to form a group in a particular distance using distance vector. AODV protocols enable a group of members to negotiate a common encryption key via open networks so that only the group members can decrypt the ciphertexts encrypted under the shared encryption key and the system does not require a trusted key server. Neither the change of the sender nor the dynamic choice of the intended receivers require extra roundsto negotiate group encryption/decryption keys.

REFERENCES

1. Bellare, M., Canetti, R., Krawczyk, H.: A Modular Approach to the Design and Analysis of Authentication and Key Exchange. In: STOC'98, pp. 419-428. ACM Press (1998)
2. Boneh, D., Boyen, X., Goh, E.J.: Hierarchical Identity Based Encryption withConstant Size Ciphertext. In: Cramer, R. (Ed.) Eurocrypt'05. LNCS 3494, pp.440-456. Springer, Heidelberg (2005)
3. Boneh, D., Franklin, M.: Identity Based Encryption from the Weil Pairing. SIAM J. of Computing 32(3): 586-615 (2003)
4. S. Yu, C. Wang, K. Ren, and W. Lou, "Attribute Based Data Sharing with Attribute Revocation," Proc. Fifth ACM Symp.Information, Computer and Comm. Security (ASIACCS '10), 2010.
5. S. Narayan, M. Gagne´, and R. Safavi-Naini, "Privacy Preserving EHR System Using Attribute-Based Infrastructure", Proc. ACM Cloud Computing Security Workshop (CCSW '10), pp. 47-52, 2010.
6. J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-Policy Attribute-Based Encryption", Proc. IEEE Symp. Security and Privacy (SP '07), pp. 321-334, 2007.
7. J.A. Akinyele, C.U. Lehmann, M.D. Green, M.W. Pagano, Z.N.J. Peterson, and A.D. Rubin, "Self-Protecting Electronic Medical Records Using Attribute-Based Encryption", Cryptology ePrint Archive, Report 2010/565, http://eprint.iacr.org/, 2010.
8. M. Chase and S.S. Chow, "Improving Privacy and Security in Multi-Authority Attribute-Based Encryption", Proc. 16th ACM Conf. Computer and Comm. Security (CCS '09), pp. 121-130, 2009.
9. X. Liang, R. Lu, X. Lin, and X.S. Shen, "Ciphertext Policy Attribute Based Encryption with Efficient Revocation", technical report, Univ. of Waterloo, 2010.
10. J. Hur and D.K. Noh, "Attribute-Based Access Control with Efficient Revocation in Data Outsourcing Systems", IEEE Trans. Parallel and Distributed Systems, vol. 22, no. 7, pp. 1214-1221, July2011.

R. Rameswari, received B.E degree in Computer science and engineering in Anna University, Tiruchirapalli. She pursuing his M.Tech Degree in Computer science and engineering in SRM University, Kattankulathur campus, Chennai-100. She worked as software programmer in Mobius knowledge service, Chennai. Now she is working as member technical staff in HCL technologies, Sollinganallur, Chennai. His research interests include computer networks, operating systems.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.