International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2016): 79.57 | Impact Factor (2015): 6.391

Data Transfer Based on Between the Node Coverge Area Using Diffe-Hellman Key Algorithm in Adhoc Network

T. Sankar[1], M. Prakash[2]

[1]Assistant Professor, Department of Computer Science, VMKV Arts and Science College, Vinayaka Mission University, Salem

[2]Assistant Professor, Department of Computer Science, VMKV Arts and Science College, Vinayaka Missions University, Salem

Abstract: ***We have used a Distributed Time Sequence Routing protocol (DTSR); the DTSR is used to locate the correct relay node and sink node for data transmission. In our wireless network is considered in to neighbor's node in the network. Using the node data will be sending in to source to destination. To reduce the energy cost, nodes are active only during data transmission and the intersection of node creates a larger merged node. Then we recognize a particular set of adhoc network applications so as to are flexible to this scalability limit. We are also improving the DTSR roaming with both network size and node density. The Diffie-Hellman Key Exchange is one of the more popular and interesting methods of key distribution. It is a public-key cryptographic system whose sole purpose is for distributing keys, whereby it is used to exchange a single piece of information, and where the value obtained is normally used as a session key for a private-key scheme It enables that adhoc nodes can communicate each other securely. The key distribution to adhoc nodes is done by means of two layer process. This paper proposes a key distribution scheme, based on intrusion detection method for using a data transmission from source to destination on the network. It based high level security and more energy efficient data transmission on their network.***

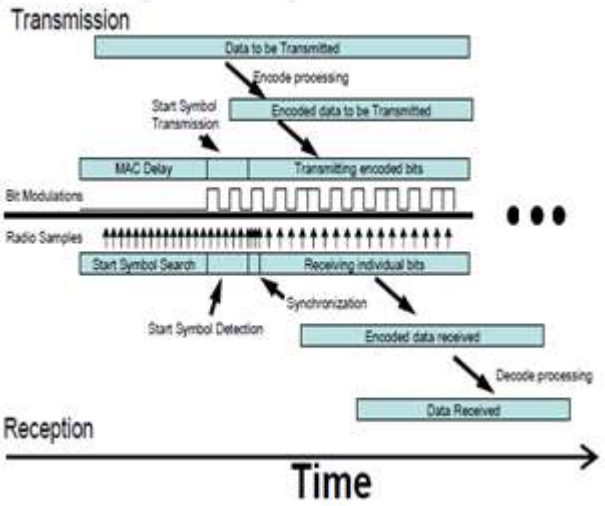
Keywords: ADHOC Network, Diffe-Hellman Key Algorithm, Man

1. Introduction

Adhoc network are an emerging technology with a wide range of potential applications such as environment monitoring, earthquake detection, patient monitoring systems, etc. Adhoc networks are also being deployed for many military applications, such as target tracking, surveillance, and security management. Adhoc network typically consist of small, inexpensive, resource constrained devices that communicate among each other using a multi hop wireless network. Each node, called an Adhoc Node, has one adhoc, embedded processors, limited memory, and low-power radio, and is normally battery operated. Each adhoc node is responsible for sensing a desired event locally and for relaying a remote event sensed by other adhoc nodes so that the event is reported to the end user.

The main characteristics of an adhoc network include

- Power consumption constrains for nodes using batteries or energy harvesting
- Ability to cope with node failures
- Mobility of nodes
- Dynamic network topology
- Communication failures
- Heterogeneity of nodes
- Scalability to large scale of deployment
- Unattended operation
- Power consumption.



Architecture Diagram for Adhoc network

Mobile stations are mounted upon public buses circulating within urban environments on fixed trajectories and near-periodic schedule. Namely, sinks motion is not controllable and their routes do not adapt upon specific adhoc network deployments. Our only assumption is that adhoc is deployed in urban areas in proximity to public transportation vehicle routes. As a fair compromise between a small numbers which results in their rapid energy depletion and a large number which results in reduced data throughput. Finally, SNs are grouped in separate clusters. Raw adhoc data are filtered within individual clusters exploiting their inherent spatial-temporal redundancy. Finally, we assume the unit disk model, which is the most common assumption in adhoc network literature. The underlying assumption in this model is that nodes which are closer than a certain distance can always communicate. However, in practice, a message sent

by a node is received by the receiver with only certain probability even if the distance of the two nodes is smaller than the transmission range.

As a harmful and easy-to-implement type of attack, a malicious node simply replays all the outgoing routing packets from a valid node to forge the latter node's identity; the malicious node then uses this forged identity to participate in the network routing, thus disrupting thenetwork traffic. Those routing packets, including their original headers, are replayed without any modification. Even if this malicious node cannot directly overhear the valid node's wireless transmission, it can collude with other malicious nodes to receive those routing packets and replay them somewhere far away from the original valid node, which is known as a wormholeattack. Since a node in an adhoc network usually relies solely on the packets received to know about the sender's identity, replaying routing packets allows the malicious node to forge the identity of this valid node. After "stealing" that valid identity, this malicious node is able to misdirect the network traffic.

For instance, it may drop packets received, forward packets to another node not supposed to be in the routing path, or even form a transmission loop through which packets are passed among a few malicious nodes infinitely. It is often difficult to know whether a node forwards received packets correctly even with overhearing techniques. Sinkhole attacks are another kind of attacks that can be launched after stealing a valid identity.

2. Theoretical Background

"Design and implementation of a Trust-aware routing protocol for large Adhoc networks"
The domain of Wireless Adhoc Networks (Adhoc networks) applications is increasing widely over the last few years. As this new type of networking is characterized by severely constrained node resources, limited network resources and the requirement to operate in an ad hoc manner, implementing security functionality to protect against adversary nodes becomes a challenging task. In this present a trust-aware, location-based routing protocol which protects the adhoc network against routing attacks, and also supports large-scale Adhoc networks deployments.

"Trust Evaluation Based Security in Wireless Adhoc Network"
The multi-hop routing in wireless adhoc networks offers little protection against identity deception through replaying routing information. An adversary can exploit this defect to launch various harmful or even devastating attacks against the routing protocols, including sinkhole attacks, wormhole attacks and Sybil attacks. The situation is further aggravated by mobile and harsh network conditions. Traditional cryptographic techniques or efforts at developing trust-aware routing protocols do not effectively address this severe problem. To secure the Adhoc networks against adversaries misdirecting the multi-hop routing, that has been designed and implemented TARF, a robust trust-aware routing framework for dynamic Adhoc networks. Without tight time synchronization or known geographic information, trustworthy, time efficient and energy-efficient route. Most importantly, TARF proves effective against those harmful attacks developed out of identity deception; the resilience of TARF is verified through extensive evaluation with both implementation and empirical experiments on large-scale Adhoc networks under various scenarios including mobile and RF-shielding network conditions.

3. Base Paper Work

In existing system move away without charitable any notice to its helpful nodes. It cause change in network topology, and therefore, it significantly degrade the performance of a steering protocol. Several direction-finding protocol studies are based on node lifetime and link lifetime. The major object here is to evaluate the node time and the link lifetime utilize the lively nature, such as the energy drain rate in addition to the relative mobility opinion rate of nodes. These two presentation metrics are included by route lifetime-prediction algorithm. This algorithm is approved out as follow select the least lively route with the best lifetime for unrelenting data forward. Node Lifetime in addition to link lifetime forecast methods, the exponentially weighted moving average technique is used to approximation the energy use up rate. The handset can measure the symbol strength when it receives the packet from dispatcher in same power level and then it calculates the distance stuck between two nodes by apply the radio spread model.

A mobile ad hoc network consists of many mobile nodes that can communicate with each other directly or through intermediate nodes. Often, hosts in a MANET operate with batteries and can roam freely, and thus, a host may exhaust its power or move away, giving no notice to its neighboring nodes, causing changes in network topology. A key characteristic of these scenarios is the dynamic behavior of the involved communication partners. Communication protocols will have to deal with a frequently changing network topology. However, many applications require stable connections to guarantee a certain degree of Quality of service. In access networks, access point handovers may disrupt the data transfer.

In addition, service contexts may need to be transferred to the new access points, introducing additional overhead and delays to the connection. In ad hoc networks, mobile services enable peer-to-peer connections for voice or data traffic. Using stable links is crucial for establishing stable paths between connection peers. Rerouting is especially costly in these networks without infrastructure, since it usually results in (at least partly) flooding the network. The stability of a link is given by its probability to persist for a certain time span, In MANETs, a route consists of multiple links in series, and thus, its lifetime depends on the lifetime of each node, as well as the wireless links between adjacent nodes.

The proposed algorithm consists of the following three phases Route discovery, Data forwarding, and Route maintenance. There are seven main differences between the EDNR and the AODV. First, in the EDNR protocol, every node saves the received signal strength and the received time of the RREQ packet in its local memory and adds this information into the RREP packet header in a piggyback

manner when it receives the RREP for the corresponding RREQ packet to meet the requirement of the connection lifetime-prediction algorithm. Second, node agents need to update their predicted node lifetime during every period. Third the node-lifetime information in the RREP packet is updated when the RREP packet is returned from a destination node to the source node.

The main contribution of this paper is that we combine node lifetime and route lifetime-prediction algorithm, which explores the dynamic nature of mobile nodes the energy drain rate of nodes and the relative mobility estimation rate at which adjacent nodes move apart in a route-discovery period that predicts the lifetime of routes discovered, and then, we select the longest lifetime route for persistent data forwarding when making a route decision. The proposed route lifetime-prediction algorithm is implemented by an exploring dynamic nature routing protocol with large scale environment based on quadrant based dynamic source routing.

MANET is a self-configuring network of mobile nodes connected by wireless links. It is an infrastructure-less system having no designated access points or routers. It follows a distributed architecture and has a dynamic topology in which each node can move randomly in an area of operation. Here each node is free to move independently in any direction, and therefore will change its links to other nodes frequently. Each intermediate node is also a router and forwards traffic sent by other nodes in the path.

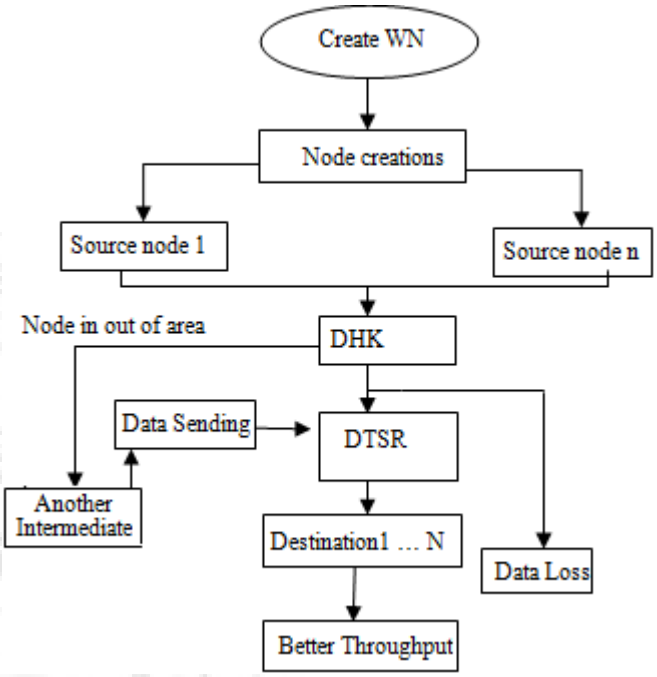
Due to topology changes caused by nodes' mobility in MANET, the routes get disconnected frequently. The existing AODV based routing protocols perform route repair scheme to repair the disconnected route. However, in most cases a source node unnecessarily performs re-route discovery of the whole path even when just one node moves out of the path. Also, if there are no checks on the selection of nodes while performing route discovery there is every possibility that a malicious node may make place in the route leading to a type of attack called the black hole attack. As far as our knowledge goes, there is no such proposed technique in the literature that takes care of both link failure and black hole attack at the same time. Our scheme does a local repair of link failure and also takes care of malicious nodes with the help of a reliability measure while performing route discovery. Simulation results show that our proposed scheme performs better in comparison to a popular existing technique.

4. Enhancement

Distributed Time Sequence Routing protocol has been used to send the data efficiently and quickly on to their network. In this algorithm to find out the correct node locate route as well as direct path in the network base on the time. DTSR protocol is to transfer the data in to without any modification. Availability parameters mean connectivity and functionality in the network management layer. Connectivity is the physical connectivity of network elements. Loss is the fraction of packets lost in transit from sender to target during a specific time interval, expressed in percentages. Have to improve the network throughput, Network delivery ratio, and availability, data loss. Consequently, the Diffie-Hellman algorithm should be used with a form of authentication such as certificates to ensure that symmetric keys are established between nodes.

To focus on the frequent case in which two people get together (e.g., at a meeting, or in the street) and make use of their devices to communicate with each other, or at least to exchange their (electronic) business cards. Clearly, the communication between these devices must be properly secured. Very often, the two users will want the security between their devices to be peer-to-peer, thus operating independently from any authority.

5. Architecture Diagram



Keying Process

In practice, this means that the mobile devices must run a protocol to authenticate each other and to protect the data they exchange (to ensure confidentiality and integrity); the latter operation typically requires setting up a symmetric shared key. This key can be used to secure both immediate communications and communications that take place afterwards. It is a common belief that peer-to-peer security is more difficult to achieve than traditional security based on a central authority; moreover, wireless communication and mobility are considered to be at odds with security. Indeed, jamming or eaves dropping is easier on a wireless link than on a wire done, notably because such mischief can be perpetrated without physical access or contact; likewise, a mobile device is more vulnerable to impersonation and to denial-of-service attacks. To compensate for the much higher vulnerability of radio channels, in some solutions users are required to type a password in both devices; in other solutions, they simply have to compare strings of words (the longer the string, the higher the security.

In contrast to this widespread belief, physical presence is the best way to increase mutual trust and to exchange information in a secure way. Indeed, authentication is

straightforward, as users can visually recognize each other(if they meet for the first time, they can be introduced to each other by a common friend whom they trust; or they can check each other's ID). In order to establish a shared key, they can make use of a location limited channel (e.g., physical contactor infrared between their two devices. The man-in-the middle attack is considered to be infeasible in these conditions.

6. Module's Description

Synchronization of Multiple Nodes
Adhoc networks most often have a much more complicated topology than the simple examples and not all adhoc nodes can communicate with each other directly. Thus, multi-hop synchronization is required, which adds an additional layer of complexity. Clearly, this could be avoided by using an overlay network which provides virtual, single-hop communication from every adhoc node to a single master node.

Diffie-Hellman Key Algorithm
It should be complemented with an authentication mechanism. In this approach for key distribution in security factors with respect fact that solving attacking problem is very challenging and that the shared key is never itself transmitted over the channel.

Distributed Time Synchronization Routing Protocol
DTSR is a reactive time synchronization protocol, which can be used to obtain times of event detections at multiple observers in the local time of the sink node(s).

Graph Design Based Result
Graph is a necessary part of display a result, so we plot a graph to demonstrate a various result comparison with packets, throughput, energy efficient and etc

Problems in Wireless Communications
Some of the problems related to wireless communication are multipath propagation, path loss, interference, and limited frequency spectrum. Multipath Propagation is, when a signal travels from its source to destination, in between there are obstacles which make the signal propagate in paths beyond the direct line of sight due to reflections, refraction and diffraction and scattering. Path loss is the attenuation of the transmitted signal strength as it propagates away from the sender. Path los scan be determined as the ratio between the powers of the transmitted signal to the receiver signal. This is mainly dependent on a number of factors such as radio frequency and the nature of the terrain. It is sometimes important to estimate the path loss in wireless communication networks. Due to the radio frequency and the nature of the terrain are not same everywhere, it is hard to estimate the path loss during communication. During communication a number of signals in the atmosphere may interfere with each other resulting in the destruction of the original signal. Limited Frequency Spectrum is where, frequency bands are shared by many wireless technologies and not by one single wireless technology.

NETWORK SIMULATOR 2.28 (NS2)
Ns-2 is a packet-level simulator and essentially a centric discrete event scheduler to schedule the events such as packet and timer expiration. Centric event scheduler cannot accurately emulate "events handled at the same time" in real world, that is, events are handled one by one. This is not a serious problem in most network simulations, because the events here are often transitory. Beyond the event scheduler, ns-2 implements a variety of network components and protocols. Notably, the wireless extension, derived from CMU Monarch Project, has 2 assumptions simplifying the physical world: Nodes do not move significantly over the length of time they transmit or receive a packet. This assumption holds only for mobile nodes of high-rate and low-speed. Consider a node with the sending rate of 10Kbps and moving speed of 10m/s, during its receiving a packet of 1500B, the node moves 12m. Thus, the surrounding can change significantly and cause reception failure. Node velocity is insignificant compared to the speed of light. In particular, none of the provided propagation models include Doppler effects, although they could.

Structure of NS-2
- Create the event scheduler
- Turn on tracing
- Create network
- Setup routing
- Insert errors
- Create transport connection
- Create traffic
- Transmit application-level data

7. How to Start TCL Scripts

We can write were Tcl scripts in any text editor. First of all, we need to create a simulator object. This is done with the command set ns [new Simulator] Now we open a file for writing that is going to be used for the nam trace data. Set ns [open out.nam w] $ns nam trace-all $ns The first line opens the file 'out.nam' for writing and gives it the file handle 'ns'. In the second line we tell the simulator object that we created above to write all simulation data that is going to be relevant for nam into this file. The next step is to add a 'finish' procedure that closes the trace file and starts nam.

Network Components
The root of the hierarchy is the Tcl Object class that is the super class of all OTcl library objects (scheduler, network components, timers and the other objects including NAM related ones). As an ancestor class of Tcl Object, Ns Object class is the super class of all basic network component objects that handle packets, which may compose compound network objects such as nodes and links. The basic network components are further divided into two subclasses, Connector and Classifier, based on the number of the possible output data paths. The basic network objects that have only one output data path are under the Connector class, and switching objects that have possible multiple output data paths are under the Classifier class.

Packet
A NS packet is composed of a stack of headers, and an optional data space. A packet header format is initialized when a Simulator object is created, where a stack of all registered (or possibly useable) headers, such as the common header that is commonly used by any objects as needed, IP header, TCP header, RTP header (UDP uses RTP header) and trace header, is defined, and the offset of each header in the stack is recorded. What this means is that whether or not a specific header is used, a stack composed of all registered headers is created when a packet is allocated by an agent, and a network object can access any header in the stack of a packet it processes using the corresponding offset value.

Starting NAM
NAM is a Tcl/TK based animation tool for viewing network simulation traces and real world packet trace data. The first step to use NAM is to produce the trace file. The trace file should contain topology information, e.g., nodes, links, as well as packet traces. Usually, the trace file is generated by ns2. During ns2 emulation, user can produce topology configurations, layout information, and packet traces using tracing events in ns2. When the trace file is generated, it is ready to be animated by NAM. Upon startup, NAM will read the trace file, create topology, pop up a window, do layout if necessary and then pause at the time of the first packet in the trace file. Through its user interface, NAM provides control over many aspects of animation.

Link
A link is another major compound object in NS. When a user creates a link using a duplex-link member function of a Simulator object, two simplex links in both directions are created. One thing to note is that an output queue of a node is actually implemented as a part of simplex link object. Packets dequeued from a queue are passed to the Delay object that simulates the link delay, and packets dropped at a queue are sent to a Null Agent and are freed there. Finally, the TTL object calculates Time to live parameters for each packet received and updates the TTL field of the packet.

Starting NS
NS starts with the command ns (assuming that we are in the directory with the ns executable, or that were path points to that directory), where is the name of a Tcl script file which defines the simulation scenario (i.e. the topology and the events). We could also just start ns without any arguments and enter the Tcl commands in the Tcl shell, but that is definitely less comfortable. Everything else depends on the Tcl script. The script might create some output, it might write a trace file or it might start nam to visualize the simulation.

Protocols
A Mobile Ad hoc Network (MANET) is a kind of wireless ad-hoc network, and is a self-configuring network of mobile routers (and associated hosts) connected by wireless links – the union of which forms an arbitrary topology. The routers are free to move randomly and organize themselves arbitrarily, thus the network's wireless topology may change rapidly and unpredictably. Such a network may operate in a standalone fashion, or may be connected to the larger Internet. There are various routing protocols available for MANETs. The most popular ones are DSR, AODV and DSDV. In this thesis, an attempt has been made to compare these three protocols on the performance basis under different environments. The comparison has been done under two protocols namely UDP and TCP. The tools used for the simulation are NS2 which is the main simulator, NAM (Network Animator) and Tracegraph which is used for preparing the graphs from the trace files. The results presented in this thesis work clearly indicate that the different protocols behave differently under different environments. The results also illustrate the important characteristics of different protocols based on their performance and thus suggest some improvements in the respective protocols. Protocol names: MANET, AODV, DSR, DSDV, NS2, NAM, UDP, TCP, Trace graph.

System Testing
System testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently before live operation commences. Testing is vital to the success of the system. An elaborate testing of data is prepared and the system is tested using this test data. Then testing errors are noted and the corrections are made for each user. The users are trained to operate the developed system. Both hardware and software securities are made to run the developed system successfully in future.

Testing Steps
- Unit Testing
- Integration Testing
- Validation Testing
- Output Testing
- User Acceptance Testing

Unit Testing
Unit testing focuses verification efforts on the smallest unit of software design and the module. This is also known as "Module Testing". The modules are tested separately. This testing is carried out during programming stage itself. In this testing step each Module is found to be working satisfactorily as regard to the expected output from the module.

Integration Testing
Integration testing is a systematic technique for constructing tests to uncover errors associated within the interface. In this project, all the modules are combined, and then the entire Program is tested as a whole. Thus in the integration testing step, all the errors uncovered are corrected for the next testing steps.

Validation Testing
Validation testing is the testing where requirements are established as a part of software requirement analysis is validated against the software that has been constructed. This test provides the final assurance that the software meets all functional, behavioral and performance requirements. The errors, which are uncovered during integration testing, are corrected during this phase.

Output Testing

After performing the validation testing, the next step is output testing of the proposed system since no system could be useful if it does not produce the required output in the specific format. The Output generated or displayed by the system under consideration is tested asking the users about the format required by then. Here, the output is considered into two ways: one is on the screen and the other is in a printed format.

The output format on the screen is found to be correct as the format designed according to the user needs .For the hard copy also; the output comes out as specified by the user. Hence output testing doesn't result in any connection in the system.

User Acceptance Testing

The testing of the software began along with coding. Since the design was fully object-oriented, first the interfaces were developed and tested. Then unit testing was done for every module in the software for various inputs, such that each line of code is at least once executed User acceptance of a system is the key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with the prospective system users at time of developing and making for proxy server.

8. System Implementation

The System implementation phase consists of the following steps:

- Testing the developed software with sample data.
- Correction of any errors if identified.
- Creating the files of the system with actual data.
- Making necessary changes to the system to find out errors.
- Training of user personnel.

The system has been tested with sample data, changes are made to the user requirements and run in parallel with the existing system to find out the discrepancies. The user has also been appraised how to run the system during the training period.

This phase is primarily concerned with user training, site preparation and file conversions. During the final testing, user acceptance is tested, followed by user training. Depending in the nature of the extensive user training may be required.

After development and testing has been completed, implementation of the information system can begin. During system implementation, the project team should be brought back to full strength. During software development stage, project teams end to play passive role as the technical steps of program development and testing evolve. However, broad organizational representation, accomplished through the project team, is required to complete the system development cycle has offer very efficient yet simple implementation techniques for development of the project.

9. Conclusion

Then the information based metric, entropy, is applied for final filtering of suspicious flow. Trust value for a client is assigned by the server based on the access pattern of the client and updated every time when the client contacts the server. Diffie-Hellman Key swap is one of the more well-liked and interesting method of key sharing. It is a public-key cryptographic system whose sole reason is for distributing keys, whereby it is used to swap over a single piece of information, and anywhere the value obtained is in general used as a sitting key for a private-key system It enables that adhoc nodes can converse each other securely.

In our work we have using in attendance are still several issues regarding the key method on environment that merit further research as the obtainable network may connect multiple stub networks. Which could make a single IP address to appear and have multiple valid hop-counts at the same time which further require enchantment in the proposed algorithm Multipath routing to check the credential of the sender for legitimate packets. Diffe helman key method is a growing threat across Internet, disrupting access to information and services. Now days, these attacks are targeting the application layer. Attackers are employing techniques that are very difficult to detect and mitigate. This paper proposes a hybrid detection scheme based on the trust information and information theory based metrics. Initial filtering is based onthe trust value scored by the node. The proposed strategy is effective and efficiently scalable that has several advantages like memory non intensive, minimum overhead in terms of resources and time, and independent of traffic pattern.

The request from the client always includes this trust value to identify itself to the server. The Web user browsing behavior rate, page viewing time and sequence of the requested objects of the client is captured from the system log during non-attack cases. Based on the observation, Entropy of requests per session is calculated and used for rate limiting. A scheduler is included to schedule the session based on the trust value of the user and the system workload.

10. Future Work

Our future work implements the security level based data transmission on network. It carries an expandable range of information resources and services which lead to bulk exchange of traffic over the collision every day. This excessive popularity creates some troubles in the networks. Among them, Node and Diffe Hellman key are the two major events. Web services needs stability and security from these two concerns. There are some methods that can discriminate DDoS attack from node and trace the sources of the attack in huge volume of network traffic. However, it is difficult to detect the exact sources of attacks in network traffic when flicker crowd event is also present. Due to the alikeness of these two anomalies, attacker can easily mimic the malicious flow into legitimate traffic patterns and defense system cannot detect real sources of attack on time. Also to implement the Entropy variation is a theoretic concept which is a measure of changes in concentration of distribution of flows at a router for a given time duration. In

future work simulation criteria for considering the resolution of specified objectives and their problem reports simultaneously, that is, the behavior of routing protocols in wireless sensor network by considering the realistic attack traces. The three metrics of Packet delivery ratio, End to end Delay and Throughput are evaluated using AODV protocol in three density regions of low density, medium density and high density in network scene as well as in node point.

References

[1] Qiang Tang, Liqun Chen, "Weaknesses in two group Diffie-Hellman key exchange protocols" 2006
[2] John Paul Walters, Zhengqiang Liang, "Wireless Sensor Network Security: A Survey" 2006
[3] Aniket Kate, Greg Zaverucha, and Urs Hengartner, "Anonymity and Security in Delay Tolerant Networks" 2008
[4] Mario ˇCagalj, Srdjan ˇCapkun and Jean-Pierre Hubaux, "Key agreement in peer-to-peer wireless networks"
[5] Jean-Fran¸cois Raymond and Anton Stiglic, "Security Issues in the Diffie-Hellman Key Agreement Protocol" 2008
[6] Sye Loong Keoh, Emil Lupu and Morris Sloman, "Securing Body Sensor Networks: Sensor Association and Key Management" 2010
[7] Yongdae Kim_, Adrian Perrig_, and Gene Tsudik, "Tree-based Group Key Agreement" 2011
[8] Victor C. Zandy and Barton P. Miller, "Reliable Network Connections" 2010
[9] Wenliang Du, Jing Deng, Yunghsiang S. Han, "A Key Pre-distribution Scheme for Sensor Networks Using Deployment Knowledge" 2004
[10] Tony Chung and Utz Roedig, "DHB-KEY: An Efficient Key Distribution Scheme for Wireless Sensor Networks" 2008