International Journal of Computer Science and Security (IJCSS)

ISSN : 1985-1553

VOLUME 4, ISSUE 6

PUBLICATION FREQUENCY: 6 ISSUES PER YEAR

Copyrights © 2011 Computer Science Journals. All rights reserved.

International Journal of Computer Science and Security (IJCSS)

Volume 4, Issue 6, 2011

Edited By
Computer Science Journals
www.cscjournals.org

Editor in Chief Dr. Haralambos Mouratidis

International Journal of Computer Science and Security (IJCSS)

Book: 2011 Volume 4, Issue 6
Publishing Date: 08-02-2011
Proceedings
ISSN (Online): 1985-1553

This work is subjected to copyright. All rights are reserved whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illusions, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication of parts thereof is permitted only under the provision of the copyright law 1965, in its current version, and permission of use must always be obtained from CSC Publishers. Violations are liable to prosecution under the copyright law.

IJCSS Journal is a part of CSC Publishers
http://www.cscjournals.org

© IJCSS Journal
Published in Malaysia

Typesetting: Camera-ready by author, data conversation by CSC Publishing Services – CSC Journals, Malaysia

CSC Publishers

Editorial Preface

This is fifth issue of volume four of the International Journal of Computer Science and Security (IJCSS). IJCSS is an International refereed journal for publication of current research in computer science and computer security technologies. IJCSS publishes research papers dealing primarily with the technological aspects of computer science in general and computer security in particular. Publications of IJCSS are beneficial for researchers, academics, scholars, advanced students, practitioners, and those seeking an update on current experience, state of the art research theories and future prospects in relation to computer science in general but specific to computer security studies. Some important topics cover by IJCSS are databases, electronic commerce, multimedia, bioinformatics, signal processing, image processing, access control, computer security, cryptography, communications and data security, etc.

This journal publishes new dissertations and state of the art research to target its readership that not only includes researchers, industrialists and scientist but also advanced students and practitioners. The aim of IJCSS is to publish research which is not only technically proficient, but contains innovation or information for our international readers. In order to position IJCSS as one of the top International journal in computer science and security, a group of highly valuable and senior International scholars are serving its Editorial Board who ensures that each issue must publish qualitative research articles from International research communities relevant to Computer science and security fields.

IJCSS editors understand that how much it is important for authors and researchers to have their work published with a minimum delay after submission of their papers. They also strongly believe that the direct communication between the editors and authors are important for the welfare, quality and wellbeing of the Journal and its readers. Therefore, all activities from paper submission to paper publication are controlled through electronic systems that include electronic submission, editorial panel and review system that ensures rapid decision with least delays in the publication processes.

To build its international reputation, we are disseminating the publication information through Google Books, Google Scholar, Directory of Open Access Journals (DOAJ), Open J Gate, ScientificCommons, Docstoc and many more. Our International Editors are working on establishing ISI listing and a good impact factor for IJCSS. We would like to remind you that the success of our journal depends directly on the number of quality articles submitted for review. Accordingly, we would like to request your participation by submitting quality manuscripts for review and encouraging your colleagues to submit quality manuscripts for review. One of the great benefits we can provide to our prospective authors is the mentoring nature of our review

process. IJCSS provides authors with high quality, helpful reviews that are shaped to assist authors in improving their manuscripts.

Editorial Board Members
International Journal of Computer Science & Security (IJCSS)

Editorial Board

Editor-in-Chief (EiC)

Dr. Haralambos Mouratidis
University of East London (United Kingdom)

Associate Editors (AEiCs)

Dr. Padmaraj M. V. nair
Fujitsu's Network Communication division in Richardson, Texas (United States of America)
Dr. Blessing Foluso Adeoye
University of Lagos (Nigeria)
Associate Professor. Azween Bin Abdullah
Universiti Teknologi Petronas (Malaysia)

Editorial Board Members (EBMs)

Dr. Alfonso Rodriguez
University of Bio-Bio (Chile)
Dr. Srinivasan Alavandhar
Glasgow Caledonian University (United Kindom)
Dr. Debotosh Bhattacharjee
Jadavpur University (India)
Professor. Abdel-Badeeh M. Salem
Ain Shams University (Egyptian)
Dr. Teng li Lynn
University of Hong Kong (Hong Kong)
Dr. Chiranjeev Kumar
Indian School of Mines University (India)
Professor. Sellappan Palaniappan
Malaysia University of Science and Technology (Malaysia)
Dr. Ghossoon M. Waleed
University Malaysia Perlis (Malaysia)
Dr. Srinivasan Alavandhar
Caledonian University (Oman)
Dr. Deepak Laxmi Narasimha
University of Malaya (Malaysia)
Professor. Arun Sharma
Amity University (India)
Professor Mostafa Abd-El-Barr
Kuwait University (Kuwait)

Table of Content

Volume 4, Issue 6, December 2011

Pages

Anomaly Detection of IP Header Threats

S.H.C. Haris shajar_charis@yahoo.com
School of Computer and Communication Engineering,
University Malaysia Perlis (UniMAP),
Kangar, Perlis

Dr.Ghossoon M.Waleed Al-Saadoon ghowaleed2004@yahoo.com
College of Administrative Sciences,
Applied Science University
Kingdom of Bahrain, Manama, Jufair, P.O.Box:5055
Tel: + (973) 17728777- 149, Fax: + (973)17728915

Ass.Prof.Dr.R.B. Ahmad badli@unimap.edu.my
School of Computer and Communication Engineering,
University Malaysia Perlis (UniMAP),
Kangar, Perlis

M.A.H.A. Ghani alifhasmani@unimap.edu.my
School of Computer and Communication Engineering,
University Malaysia Perlis (UniMAP),
Kangar, Perlis

Abstract

Threats have become a big problem since the past few years as computer viruses are widely recognized as a significant computer threat. However, the role of Information Technology security must be revisit again since it is too often. IT security managers find themselves in the hopeless situation of trying to uphold a maximum of security as requested from management. At the same time they are considered an obstacle in the way of developing and introducing new applications into business and government network environments. This paper will focus on Transmission Control Protocol Synchronize Flooding attack detections using the Internet Protocol header as a platform to detect threats, especially in the IP protocol and TCP protocol, and check packets using anomaly detection system which has many advantages, and applied it under the open source Linux. The problem is to detect TCP SYN Flood attack through internet security. This paper also focusing on detecting threats in the local network by monitoring all the packets that goes through the networks. The results show that the proposed detection method can detect TCP SYN Flooding in both normal and attacked network and alert the user about the attack after sending the report to the administrator. As a conclusion, TCP SYN Flood and other attacks can be detected through the traffic monitoring tools if the abnormal behaviors of the packets are recognized such as incomplete TCP three-way handshake application and IP header length.

Keywords: TCP SYN Flood, Rate-Based Detection, Three-Way Handshake, IP Header, TCP Header

1. INTRODUCTION

Threats have been a big problem to internet security nowadays especially for security management department to maintain the level of their security from being threaten by intruders.

There are many types of threats in the internet such as phishing, hackers, worms, virus that occur every day without being realized by the users. Intruders will do anything to attack and violence the network even though the security management had upgraded their security with the newest defence methods. The intruders target to break all the security especially in government, military, banks and others because whenever the intruders can break in the network, that network might lose data, money or confidential information and documents. The objective of this paper is to detect TCP SYN Flood attack that occurs in TCP protocol before it affects the network system.

TCP SYN Flood is hard to detect and we used anomaly detection because it is the most frequently suggested approach to detect attack variants, which looks for abnormal behavior. This paper focusing on detecting threats in the local network by monitoring all the packets that goes through the networks. This paper is comprised in two main parts. First part, do the monitoring and analysis the normal flow of the network and second part is monitor and analysis the network that had been attack by TCP SYN Flood.

Results show that there are threats in normal traffic without any alarming to the users. The suggested detection method can detect TCP SYN flood and other threats in the normal network and attacked network. Then, the system will send a report to the administration for warning all the users in the network.

2. LITERATURE REVIEW

Several methods for detecting TCP SYN flood attacks have been proposed.In network security architecture, network intrusion detection systems such as SNORT [2] and Bro [3], detect signatures of known attack such as packet payload inspection, buffer overflows according to the rules that had been written in this application. It differs from the anomaly detection system such as Network Traffic Anomaly Detection (NETAD) [4]. NETAD filtered the traffic and examined only the start of incoming server requests. It starts with the IP header, every first 48 bytes is treated as an attribute and do not parse the packet into fields. NETAD used nine separated models corresponding to the most common protocols such as IP, TCP, and UDP. The anomaly score tn/r was modified to scare rare. The t value is the time since the attributes was last anomalous, n is the number of training observations, and r is the size of the set of allowed values. Only the start of incoming server requests are examined after filtered the traffic.

- The Flood Detection System (FDS), which used Cumulative Sum (CUSUM) that detect the SYN flooding attacks at leaf routers which connect end hosts to the Internet, instead of monitoring the ongoing traffic at the front end (like firewall or proxy) or a victim server itself. The detection utilizes the SYN-FIN pairs' behavior and distinguish features FDS make it immune to SYN flooding attacks, CUSUM method that make the detection robust, and it does not undermine the end-to-end TCP performance. This mechanism not only sets alarms upon detection of ongoing SYN flooding attacks, but also reveals the location of the flooding sources [5].
- Partial Completion Filters (PCF) has an independent interest because they provide a solution to the general problem of detecting imbalanced parentheses in streaming environment. It consists of parallel stages containing buckets that are incremented for a SYN and decremented for a FIN. Thus, if a destination hashes into buckets with large counters in all stages, it seems plausible that the destination is being attacked [6].
- The comparison of three types SYN Flooding detection has been done in this research. Most of the researches used TCP control packets only as an input and each is designed to be deployed at the edge of a leaf network. The results show that FDS has good detection speed but long time to return to non-alert state [7].
- The significantly and negatively affected by attacks that create high variance in the traffic rate, but faster in signaling the end of an attack, and PCF performs well with regards to both detection time and quiescence time[8].
- Architecture of an anomaly detection system is based on the paradigm of Artificial Immune Systems (AISs). Incoming network traffic data are considered by the system as signatures of potential attackers by mapping them into antigens of AISs either using some parameters of network traffic or headers of selected TCP/IP protocols. A number of methods for generation of antibodies (anomaly detectors) were implemented. The way of anomaly detection depends on the method of antibodies generation. The paper presents results of an experimental study

performed with use of real data and shows how the performance of the anomaly detection system depends on traffic data coding and methods of detectors generation [9].

- Packet Header Anomaly Detector (PHAD) learns the normal range of values for 33 fields of the Ethernet, IP, TCP, UDP, and ICMP protocols. On the 1999 DARPA off-line intrusion detection evaluation data set, PHAD detects 72 of 201 instances (29 of 59 types) of attacks, including all but 3 types that exploit the protocols examined, at a rate of 10 false alarms per day after training on 7 days of attack-free internal network traffic. In contrast to most other network intrusion detectors and firewalls, only 8 attacks (6 types) are detected based on anomalous IP addresses, and none by their port numbers. A number of variations of PHAD were studied, and the best results were obtained by examining packets and fields in isolation, and by using simple no stationary models that estimate probabilities based on the time since the last event rather than the average rate of events [10].

3. TCP SYN FLOODING

TCP SYN Flooding are a common form of denial-of-service attacks launched against IP based hosts, designed to incapacitate the target by exhausting its resources with illegitimate TCP connections [1]. A normal TCP connection usually start a transmission from client by sending a SYN to the server, and the server will allocates a buffer for the client and replies with a SYN and ACK packet. At this stage, the connection is in the half-open state, waiting for the ACK reply from the client to complete the connection setup. When the connection is complete, it called 3-way handshake and TCP SYN Flood attack manipulate this 3-way handshake by making the server exhausted with SYN request.

This application is different from TCP SYN Flooding attack which takes an advantage of the half-open state condition by sending multiple SYN packets with spoofed address. Figure 1 shows the TCP SYN Flood happened. An attacker will send multiple SYN requests to the victim server using a spoofed address. A victim server sends back a request SYN and ACK packet to the client or spoofed address and wait for confirmation or timeout expiration of SYN packets. If the client does not send back the final ACK packet, the server's resources can be easily exhausted. At this time the TCP SYN Flood attack occurred because too many SYN packet request from clients.

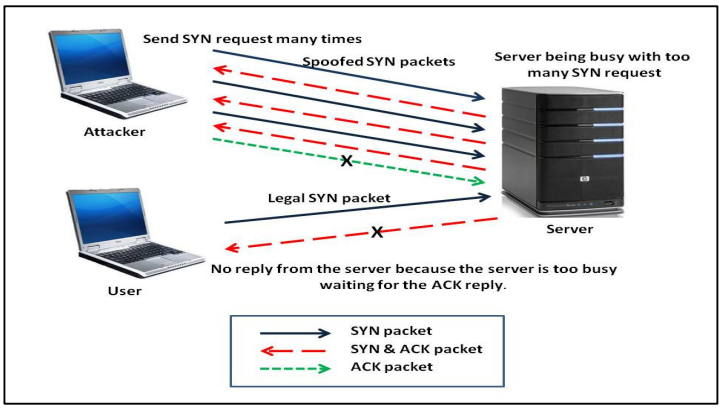


FIGURE 1: TCP SYN Flood Attack

The attack succeeds because the number of half-open connections that can be supported per TCP port is limited. When the number of half-open connections is exceeded the server will reject all subsequent incoming connection requests until the existing requests time out, creating a denial-of-service condition.

4. DETECTION METHODS

There are two types of network intrusion detection system which are signature based detection or anomaly based detection. Signature detection is a technique often used in the Intrusion Detection System (IDS) and many anti-malware systems such as anti-virus and anti-spyware. In the signature detection process, network or system information is scanned against a known attack or malware signature database. If match found, an alert takes place for further actions [10, 11].

In this paper, rate-based detection will be used for anomaly detection. Anomaly detection has three types of detection in network analysis behavior: It used protocol to detect packets that are too short which violate specific application layers protocol, rate-based detection which detects floods in traffic using a time-based model of normal traffic volumes especially Denial of Service (DoS) attacks. Lastly, it detects through the behavioral or relational changes in how individual or groups of hosts interact with one another on a network.

5. METHODOLOGY

In order to perform this research, the network under UniMAP (University Malaysia Perlis) is being used as a platform to capture the packets. The operating system used the open source GNU/Linux -Ubuntu ver. 9.04 and focusing the network inside UniMAP Research Cluster. Linux is used because this operating system is stable and also capable to act as client or server and free to modify the system. In this research, the experiments are divided into two categories which are the normal flow of the network had been monitored to see the data of the packets and checked for the threats. Secondly, the network that had been attacked with TCP SYN Flooding and the data is checked.

In order to evaluate this experiment, a simple networking testbed was constructed. Shown in Figure 2, this essentially consists of three Linux clients and one Linux PC as a traffic monitoring tools. All the data from the client will be monitored by traffic monitoring tool.

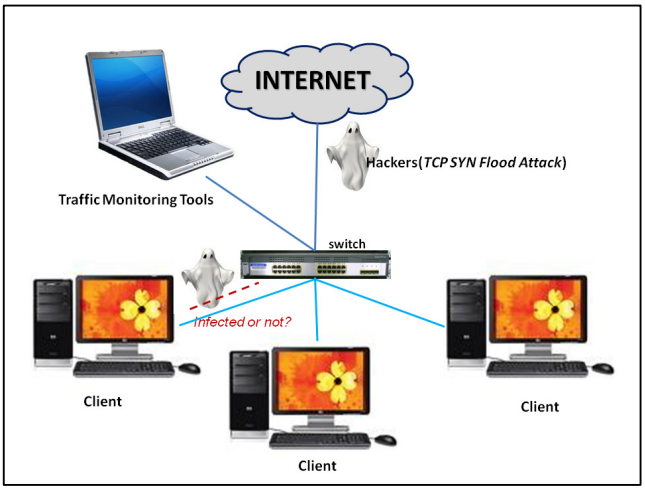


FIGURE 2: Testbed for Traffic Monitoring

This research used tcpdump software to sniff the packet that got through the network and internet in the real time. The data is saved after capturing the packets for the next process, analysis the packets. The captured data will be analyze every 1000 packets and check for threats. These packets that go through the network are analyzed according to the protocols (TCP, UDP and ICMP). The analysis packets based on IP and TCP headers of the packets from the monitored traffic. This analysis was focusing on IP header of the IP packets regardless of their layer protocol such as TCP packets and UDP packets which in transport layer protocol, and ICMP packets in the network layer protocol. UDP, ICMP and other types of packets could also be used in security breaches according on the IP payload and unusable area.

5.1 Algorithm

The algorithm for this paper is illustrated as Figure 3. The packets will go through the anomaly detection software that sniffs using tcpdump. IP Header analysis includes each field such as IP Header Length (IHL), Type of Service (ToS), Identification (ID), Flags and etc. The IP packet header consists of 20 bytes of data and if the length is below than 20 bytes, that packet is assume as abnormal packet and go for analysis before report to administrator. An option exists within the header that allows further optional bytes to be added, but this is not normally used.

The TCP header is analyzed for the next step in this process since TCP SYN Flooding is the main threats. TCP header is built on top of IP header, which is unreliable and connectionless. TCP header occupies 20 bytes and has some limitations in header length. As mentioned, normal TCP header is 20 bytes but TCP can have another 40 bytes for option. So the header size is limited to 60 bytes. TCP Flags have six flags bits namely URG, ACK, PSH, RST, SYN and FIN, each of them has a special use in the connection establishment, connection termination or control purposes. Only few combinations of the six TCP flags can be carried in a TCP packet. URG and PSH flags can be used only when a packet carries data, for instance a combination of SYN and PSH becomes invalid. Since TCP SYN Flooding attack will flood the network with SYN packets, the three-way handshake application is checked in every packet.

At this stage, packets are divided into two groups whether infected packets or normal packets. If the packet is infected, the system will distinguish the packet and go for analysis again to confirm whether the packet is truly comes from attackers. Otherwise, the normal packet will go through the network sending the data to the destination.

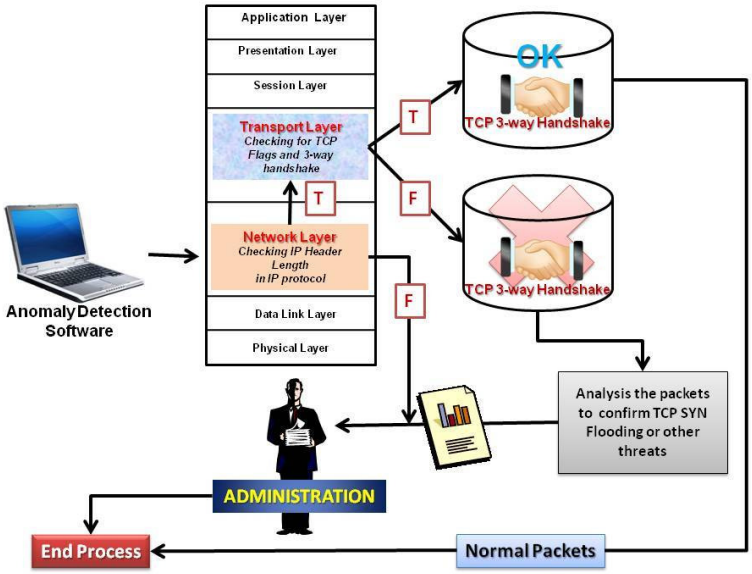


FIGURE 3: Packet Filtering Flowchart

5.2 Packet Filtering

In order to do the packet filtering, many factors will be considered. There are three main factors in this paper:

1. The traffic filtered each packet to each protocol such as TCP, UDP and ICMP.
2. TCP flags SYN, ACK, RST, FIN, are divided to each group to check the three-way handshake is complete or not.
3. IP address is valid and not a spoofed address.

These factors are important for detection method to recognize which packets are the infected packets in order to distinguish the normal packet from abnormal packet. Then the analysis for each packet is done using these factors.

6. EXPERIMENT RESULTS

The experimental result is divided into two parts. First part, monitoring and analysis the normal flow packet in the network. Analysis all the packet header and check for the threats for each packet. The normal packets behaviors are being analyzed according to each protocol and header. Each protocol has header and function according to the TCP/IP protocol.

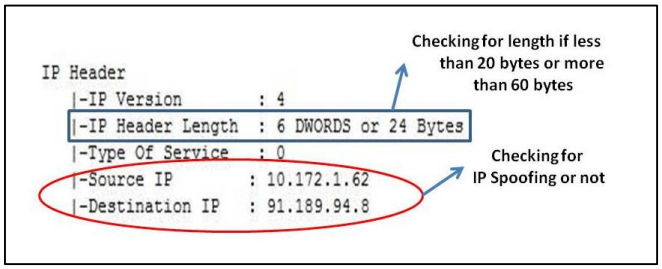


FIGURE 4: Packet Filtering Flowchart

Figure 4 above shows the field that had been observed and analyzed in IP header for the first part experiment. There are five main fields that are important in order to detect threats. This experiment is focusing on Internet Protocol Version 4 (IPV4), so the IPV must be 4 and IP header length must be equal or above than 20 bytes and equal or below than 60 bytes. ToS should be zero for normal packets and the IP address is not the spoofed address because hackers will use IP Spoofing to attack the network. At the same time, TCP three-way handshake application is checked.

The second part is analyzed the packet that been attacked by TCP SYN Flooding. This analysis is focusing in TCP protocol that is in transport layer according to Open System Interconnection (OSI) model by monitoring its behavior such as the flow of the packet, TCP header and flags. The attack had been run for half an hour and at the same time the data (packet) is save in the real time. If the data is not saved after capturing, the data may be flushed away and actual packet contents are no longer available.

In order to detect the SYN flood, the internet must be connected and this detection is focusing on port 80, HTTP. Other port than 80 the packet will be discard. HTTP has been chosen because most of the internet user or web used this port as a platform for communication. It had been reported in 1996 that 65 percent of the web users used HTTP.

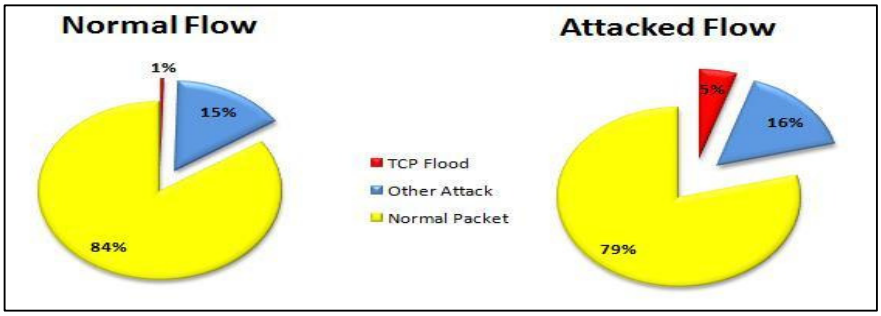


FIGURE 5: Threats in Network

From the experiment, there is SYN flood attack occurred in the normal network in the cluster but it was not much compared to other threat which is more in the network. Figure 5 shows the differences between a normal flow packet and attacked network flow packet. TCP SYN flood attack percentage had been increased after the system had been attack and this detection

method also detected other packets that have an error in the payload and IP header especially. An abnormal behaviour of the packet traffic especially in TCP protocol is important to detect in order to defence the network. Even though, the normal packets for both experiments are showing the highest range, the flooding attack can reduce the normal packets if it the flooding amount is bigger and can make the connection to the internet slower than usual.

From the analyzed packets through the TCP port, there are more threats other than SYN flood attack such as Trojan and backdoor, as shown in Figure 6 below.

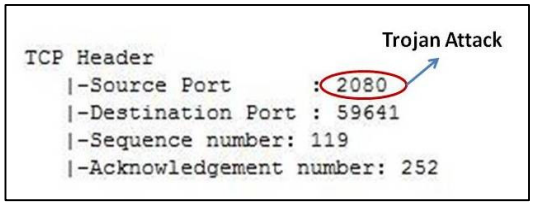


FIGURE 6: Trojan Attack in the Network

7. DISCUSSION

IP header is the main platform for detecting the threats especially in IP protocol and TCP protocol that are applied under open source Linux. Monitoring can detect abnormal behaviour in the network. In military, banking, government and many expertise departments, if SYN flood occurred in the system but there is no monitoring done, and no expertise, the network team will never realize that they had been attacked. In contrast with other work [5], the suggested method is much better because it alarming the administrative department from the start of the attack.

It had been reported that a virus called Agent.btz, variation of the "SillyFDC" worm, which spreads by copying itself to thumb drives had attack U.S. military network. When that drive or disk is plugged into a second computer, the worm replicates itself again. The attack struck hard at networks within U.S. Central Command, the headquarters that oversees U.S. involvement in Iraq and Afghanistan, and affected computers in combat zones. The attack also penetrated at least one highly protected classified network and this will affect all the network system at the same time. Such important information must be send immediately to top management as an example, terrorist is trying to ambush an army camp but the information is not sent because of the flooding. The suggested method detects through the behavioral and relational changes in how individual or groups of hosts interact with one another on a network. Comparing the suggested method with other suggested method such as [5] which detects attacks at leaf routers, variance in traffic rate as in [8], detectors generation and antibodies as in [9], or a range of values in a certain number of fields for different protocols, you will see that the suggested method has the comprehensively solution to meet the requirements for such type of attack.

Anomaly detection is important to detect SYN flood and other flooding attacks in huge network system as well to prevent the data loss and traffic jam since it will create Denial of Service (DoS) if it is uncontrolled.

8. CONCLUSION

The analysis for the packets is to detect threats that attack through the network. These threats are detected due to the IP Header (payload and unusable area). These detections are not only for TCP, but include the IGMP, ICMP and UDP. Receiving some numbers of duplicate ACKs means that the network congestion has been occurred.

In the experiment, the main threats in this paper, SYN Flood attack had been traced in a small amount. It is because the Linux operating system is stable and it is very hard to attack by the hackers.

The analysis for the packets is to detect threats that attack through the network. Threats are

detected due to the IP Header (payload and unusable area). This detection is not focusing only for TCP, but others protocol such as IGMP, ICMP and UDP also being examined.

By analyzed every packet to each category in TCP protocol (port, flags, and TCP three-way handshake) and IP header, the threats are easier to detect once we know the behavior of an attack.

In the experiment, the main threats in this paper, SYN Flood attack had been traced even in a normal network. The detection method of attacks can be improve in order to make the detection faster and effective, and alarming the security administration department whenever there is an attack or abnormal behavior in the flow of the traffic.

9. REFERENCES

1. *"Using SYN Flood Protection in SonicOS Enhanced"*,[online] available at: http://www.sonicwall.com/us/support/2134_3480.html

2. Roesch, Martin, *"Snort - Lightweight Intrusion Detection for Networks"*, Proc. USENIX Lisa '99, Seattle: Nov. 7-12, 1999.

3. Paxson, Vern, *"Bro: A System for Detecting Network Intruders in Real-Time",* Lawrence Berkeley National Laboratory Proceedings, 7'Th USENIX Security Symposium, Jan. 26-29, 1998, San Antonio TX.

4. Mahoney, M, *"Network Traffic Anomaly Detection Based on Packet",* ACM (2003).

5. H. Wang, D. Zhang, K. G. Shin, *"Detecting SYN Flooding Attacks ",* Proc. INFOCOM IEEE Communications Society, (2002).

6. R. Rao, K., Sumeet, S., & V. George, "*On Scalable Attack Detection in the Network",* Networking, IEEE/ACM Transactions on, 15(1):14-25.

7. Beaumont-Gay, M, *"A Comparison of SYN Flood Detection Algorithms",* Internet Monitoring and Protection, 2007. ICIMP 2007.

8. V.A. Siris, F.Papagalou. *"Application of anomaly detection algorithms for detecting SYN flooding attacks"*, Proc. of Globecom, IEEE Communications Society, 2004.

9. "*Signature Detection*", [online] available at: http://www.javvin.com/networksecurity/SignatureDetection.html

10. Franciszek, Seredynski & Pascal Bouvry "*Anomaly detection in TCP/IP networks using immune systems paradigm*", ELSEVER , Computer Communications 30 (2007) 740–749, _ 2006 Elsevier B.V. All rights reserved.

11. Matthew V. Mahoney and Philip K. Chan, *"PHAD: Packet Header Anomaly Detection for Identifying Hostile Network Traffic"*, Florida Institute of Technology Technical Report CS-2001-04

12. *Signature Detection",* [online] available at: http://www.javvin.com/networksecurity/SignatureDetection.html

13. M. Bykova, S. Ostermann, *"Statistical Analysis of Malformed Packets and Their Origins in the Modern Internet",* 2nd Internet Measurent Workshop (IMW 2002), Nov. 2002.

A Security Analysis Framework Powered by an Expert System

Maher Mohamed Gamal mahergamal@gmail.com
Computer Science Arab Academy of Science, Technology and Maritime Transport Cairo, Egypt

Dr. Bahaa Hasan bahaa.hasan@asc-egypt.org
Chairman & CEO of Arab Security Consultants (ASC) Cairo, Egypt

Dr. Abdel Fatah Hegazy ahegazy@aast.edu
Computer Science Arab Academy of Science, Technology and Maritime Transport Cairo, Egypt

Abstract

Today's IT systems are facing a major challenge in confronting the fast rate of emerging security threats. Although many security tools are being employed within organizations in order to standup to these threats, the information revealed is very inferior in providing a rich understanding to the consequences of the discovered vulnerabilities. We believe expert systems can play an important role in capturing any security expertise from various sources in order to provide the informative deductions we are looking for from the supplied inputs. Throughout this research effort, we have built the Open Security Knowledge Engineered (OpenSKE) framework 1, which is a security analysis framework built around an expert system in order to reason over the security information collected from external sources. Our implementation has been published online in order to facilitate and encourage online collaboration to increase the practical research within the field of security analysis.

Keywords: Security Analysis, Expert System, Vulnerability Analysis, Security Framework, Attacks.

1. INTRODUCTION

Probably any organization today will probably need to benefit from the productivity that computers bring by to many applications within the organization's field. Unfortunately, with this productivity, comes a great risk of being prone to computer security attacks due to any existing vulnerable or misconfigured software. This has led organizations today to leverage various security tools in order to keep up with the continuous threats to their valuable assets and services. Various security tools such as port scanners, anti-viruses, intrusion detection systems and similar programs have all proved their usefulness by providing network administrators with the necessary information in order to identify their systems' defects.

Unfortunately, the information revealed by these security tools mostly provides a very inferior study to how these scattered pieces of information form together a bigger meaning along with it's consequences. This is why well-funded organizations would hire highly specialized professionals (aka. Red Team [2]) in order to lay out all of the collected data and analyze any possible attack intents. They usually end up with a graph of how the present vulnerabilities on the systems can lead to one or more potential attacks. Thus, there is a dire need to gain a deeper understanding from the security reports and information that are being extracted by the deployed sentinels in order to fully understand what is really happening behind the scenes. For example, even if a port scanner does reveal some open ports on a specific host, that doesn't designate a real problem since we may have public services listening on these ports. On the other hand, having these ports open on this specific machine with no need can lead to unknown potential attacks. So let us dig deeper into how attacks are performed.

A security attack can be performed by executing one or more exploits according to what it needs in order to be accomplished. An *exploit* is a program that leverages one or more vulnerabilities located in any of the installed software in order to cause an unintended behavior on the target system.

Previous efforts have been made in order to describe the attack concepts and one that really inspired us was Templeton and Levitt's [1] effort where they modeled the components that constitute an attack and how they relate to each other. This way of thinking breaks down the notion of an attack into it's constituents. In doing this, we can start studying the requirements of an attack's component and it's effect on it's surrounding environment.

This is illustrated in Figure 1 where we have an attack that can be achieved by leveraging two exploits, each having it's own capability requirements. A capability here can be an open port, a file permission, a vulnerability in a specific library or program ...etc. Therefore, when Exploit 1's three capabilities are met, it can be executed, which consequently makes Exploit 2's capabilities satisfied and thus, Exploit 2 can be executed leading to more capabilities available.

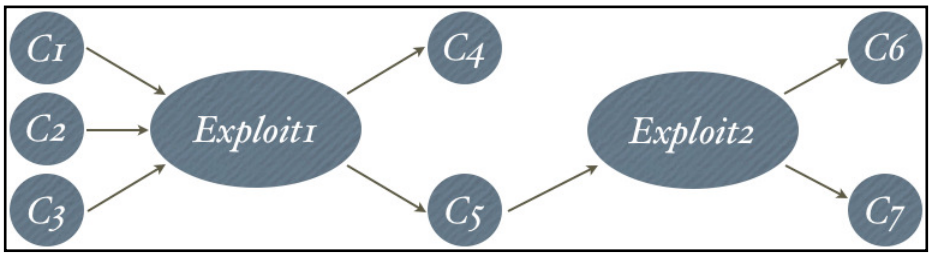


FIGURE 1 : Shows how an attack's components lead to each other through their capability requirements and offerings.

2. EARLY APPROACHES

Honestly, the field of security analysis isn't anew. A substantial amount of research has gone through several approaches to address this field. We will present the approaches that were relevant to our research in addition to what shortcomings that have been found in each of them.

2.1. Hard-Coding Vulnerability Checks

In 1987, Robert Baldwin published the first paper that proposed a rule based analysis method which was named Kuang [2]. Later came Daniel and Eugene to form this method into a practical security checker [3]. The efforts until then considered only vulnerabilities on a single host. Further

2 Red Team, http://en.wikipedia.org/wiki/Red_team

research was made to make Kuang work on multi-hosts on the same network, it was named NetKuang [4].

Unfortunately, the Kuang approach had the vulnerability checks hard-coded into it's implementation. Even though this approach was sufficient at it's time, nowadays, we are facing a rapid rate of vulnerability discoveries that render this approach impractical since any security checker nowadays needs to be able to import multiple formal specifications of vulnerabilities from various sources. In addition to this, we see that most of the attacks happening these days are a result from multi-staged sub-attacks on multi-hosts.
Nevertheless, we have borrowed the paradigm of using a rule-based method to analyze computer security in a similar fashion as we will see later on.

2.2. Model Checking

Model checking [5] is basically a state-transition system that is being checked whether it still satisfies a correctness condition. Applying model checking to network security can be in the form of modeling our systems as a state, where an attack on our systems would cause a transition from the current state to a different state. The state transition can be described in the form of the preconditions that need to be satisfied in order for the transition to be performed and the postconditions that would result from the transition. A full attack path would be a series of state transitions that would eventually violate the correctness condition (e.g. accessing classified data) upon being performed.

Unfortunately, as noted by Xinming [9], the drawback of model checking is that most state-transition sequences of the system are examined and with a large scale, this may eventually lead to a state-space explosion. In network security we only need to analyze what is feasible to be done from our current situation, not what could be done in the system's entirety disregarding it's achievability.

2.3. Attack Graph Analysis

The attack graph analysis approach, has previously attracted a hefty amount of research effort. The aim of this approach is to deliver an exploit-dependency graph which is identical to what we illustrated in Figure 1. The attack graph is used to analyze the possible actions the attacker can take in order to reach the target. Unfortunately, there has been several scalability problems as outlined in Lippmann's detailed review [6] of the previous publications on this topic. Although there has been several efforts listed in Lippmann's review that attempt to solve the scalability problems, we have decided not to take this approach as we have decided to leverage the power of a logical reasoner as we will see in the next section.

2.4. Logic-Programming

The logic-programming approach was introduced by Xinming [7] and Sudhakar [8] in their Datalog [3]-based security analysis framework MulVAL [9]. This approach has shifted our thinking of attack graphs into making them an outcome from the logical deductions performed over our domain understanding which is represented in the form of Datalog predicates. MulVAL produced full traces of the exploits that could be executed based on the experimented situations.

After looking into how MulVAL worked, we believe that MulVAL holds a couple of shortcomings which are listed below, though it still holds as one of the major inspirations to our research.

1. MulVAL is based on Datalog which can only provide an offline-mode of security analysis which means that in order for MulVAL to deduce any new information, it has to be asked for it. Although this is totally acceptable for what MulVAL was intended for (which is to generate attack traces), we believe this can be further improved to turn into an online

3 Datalog is a subset of Prolog, http://en.wikipedia.org/wiki/Datalog

analyzer where newly picked up security information is detected and fed into the analyzer which deduces new information.

2. MulVAL's domain modeling was in the form of Datalog predicates which on a large scale can turn out to be unmaintainable. A single entity's information is distributed among multiple predicates, which makes the understanding of the domain model harder to grasp and keep well maintained.
3. Datalog has mostly been used for academic purposes and we believe that in order for any open framework to be widely used and built upon, it has to be easily adoptable and the programming language used plays an important role in this.
4. In addition to the above, we intend to provide a publicly available open implementation of our framework that we hope would facilitate further research in this topic.

In the next section we will explore an Artificial Intelligence area called Expert Systems where we will see how it fits into the field of security analysis.

3. LEVERAGING AN EXPERT SYSTEM

Expert Systems [10] have long been a popular branch of Artificial Intelligence research. It's popularity has mainly stemmed from it's ability to reason over a problem based on it's current understanding of the situation.

To further understand what is meant by reasoning, it is when a system that holds some knowledge, is required to do or provide something that it was not explicitly informed with. Thus, the system must figure out what it needs to know from what it already knows.

3.1. The Structure of an Expert System

In order for expert systems to perform any kind of reasoning, they require the knowledge to be represented in a comprehensible format which would be known as it's knowledge representation. A collection of formalized pieces of information in a well-defined representation would be described as it's knowledge base and this forms the first of the two components that compose an expert system. The second part of an expert system is the logical reasoner which is the central brain that performs all of the necessary reasoning over the previously built knowledge base. The benefit of performing logical reasoning is that we can conclude new information, which can enlighten us and let us look at our situation with a better understanding.

3.2. Rule Chaining

```
IF
   <conditional expression(s)>
THEN
   <knowledge insert/update/retract statement(s)>
```

FIGURE 2 : An overly simplified structure of an expert system's rule syntax

The expert system's reasoner operates over well-defined domain rules. These rules can be thought of as *IF-THEN* statements as shown in Figure 2. Once the *IF* part of the statement is satisfied (i.e. the current situation implies that this rule should be fired) the *THEN* part is computed which can introduce additional information that could be useful to us, plus it manipulates the knowledge base which can recursively cause more rules to be fired and thus, we end up with what we call, *forward chaining*.

An another way in which expert systems can operate, is called *backward chaining*. Here the expert system tries to prove whether a goal can be reached from the current understanding of the situation. This is mainly done be reversing the way the rules are traversed and this is what was adopted by the MulVAL [9] authors by using the Datalog language.

3.3. An Analogy between Templeton's Model and an Expert System

Comparing Templeton's [1] attack model to how an expert system's reasoning works, it is obvious how expert systems fit elegantly. As illustrated in Figure 3, Templeton's attack concept is represented in the expert system as a rule statement and the capabilities are represented as any piece of knowledge that is being required by any of the domain rules of the expert system.

Requires/Provides Model	***Expert System***
Attack Concept	Rule Statement
Capability	Piece of Knowledge

FIGURE 3 : Representation of Templeton's model in an Expert System

3.4. Choosing a Suitable Expert System

The real essence behind an expert system's logical reasoner is how it organizes the rules in an efficient manner to minimize the time taken to pass through all of the *IF* parts of the rules to evaluate them upon any updates to the knowledge base. Today's expert systems mainly build over the *Rete algorithm* [11] that was designed by Charles L. Forgy in 1982, which forms as one of the most efficient algorithms in maintaining and processing the rules of an expert system.

The expert system that we have found appropriate for our goal was Drools [4]. It's an open-source Rete-based expert system shell written in Java [5] which performs forward-chaining and features a very simple rule syntax that is easily comprehensible. We have favored Drools over others due to the following.

1. It supports forward-chaining which will highly aid in providing an online security analyzer that can receive a constant feed of security events.
2. The domain model is described as an object-oriented design which allows us to highly describe our domain problem with all possible relations.
3. It's rule syntax is very simple which will highly encourage security experts to contribute in writing the security rules.
4. Drools is built over Java which we believe is one of the most popular development platforms available today.
5. The Drools project is actively maintained and well documented.

The goal of this research effort is to leverage Drools as our expert system to capture any possible security knowledge, whether it's from an expert's technical expertise or security advisories in addition to the current network situation in order to conclude meanings that weren't perceptible before. On our way to achieve this, we will be facing the notion of formalizing the information that's being fed into Drools. After that, we will inspect how the Expert rules are written. Finally, we will conclude our work with the results that we have reached and what we envision to be possible for future development.

---

[1] 4 Drools, http:

5 Java, http://en.wikipedia.org/wiki/Java_(programming_language)

4. INCORPORATING OPEN COMMUNITY-DRIVEN STANDARDS

One of the greatest challenges in building our security analysis framework was coming up with a reasonable knowledge base to work on. Long ago, the learnings of computer security have been weakly formalized or even verified for it's correctness. Even with security advisories reporting the latest vulnerabilities, they were sent out as free text to mailing lists which are difficult to depend on in our research.

Today, with the rise of several community-driven efforts under the *Making Security Measurable Initiative* [12] to establish common standards in order to unify the understanding of several aspects of computer security, we decided to take it a chance to incorporate what is possible from their publicly available XML data-sets into our security analysis framework.

Below is a brief listing and description of what suited our framework's initial scope.

1. *Common Vulnerabilities and Exposures Enumeration (CVE)*

   The CVE standard is a constantly updated comprehensive dictionary of security vulnerabilities and exposures. These are specific to public releases of widely used software. We will be referring to CVE identifiers whenever we refer to specific vulnerabilities.

2. *Common Platform Enumeration (CPE)*

   The CPE standard proposes a unified naming convention for systems, platforms and software packages, in order to avoid any ambiguity when referring to a specific package version on a specific operating system.

3. *Open Vulnerability and Assessment Language (OVAL)*

   The OVAL scanner sweeps through the inspected systems searching for vulnerabilities that match any predefined signatures and reports them in the form of standard CVE identifiers or identifiers from the National Vulnerability Database (NVD) [6].

4. *Common Weakness Enumeration (CWE)*

   The CWE describes the software security weaknesses whether it's in architecture, design or code. Weaknesses can be thought of as the root causes of vulnerabilities. Each weakness is linked to it's observed vulnerabilities.

5. *Common Attack Pattern Enumeration and Classification (CAPEC)*

   The CAPEC provides a higher level view to the weaknesses in CWE and vulnerabilities in CVE/NVD. It shows the attack patterns that the attacker can perform by leveraging the weaknesses found in our systems in order to perform any unintended behavior.

5. INTRODUCING OPENSKE, THE FRAMEWORK

Our publicly available research-oriented framework, the Open Security Knowledge Engineered 7 (pronounced as open-skee) has been designed in order to leverage Drools as it's expert system in addition to surrounding it with all of the necessary auxiliaries to facilitate it's goal of analyzing network security. We have decided to open-source the implementation and provide it publicly in

6 National Vulnerability Database, http://nvd.nist.gov/
7 OpenSKE, http://code.google.com/p/openske

order to facilitate practical collaboration and to provide a basis for future research that can build over it. Before we delve into the framework's internals, let us list what we expect the framework to serve us.

1. Identify which of our assets may be affected by the present vulnerabilities in our systems.
2. Provide a list of CWE weaknesses (root causes) behind the existing CVE vulnerabilities.
3. Provide a list of CAPEC attack patterns that can be executed based on the existing weaknesses and vulnerabilities.
4. Report any activity performed by any attacker(s).

5.1. OpenSKE's Inputs and Outputs

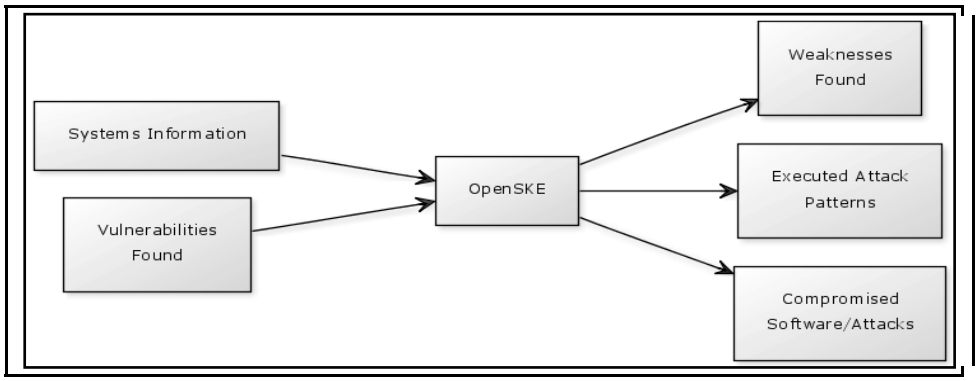


FIGURE 4 : An overview of OpenSKE's inputs and outputs

- ***OpenSKE's Inputs***

    1. *Systems Information*

        A full enumeration of all of the networked hosts with all of their user accounts, assets and applications running on them, ...etc.

    2. *Vulnerabilities Found*

        We will be supplying OpenSKE with the results of our vulnerability scanners such as the OVAL scanner to support pinpointing the vulnerabilities in our systems.

- ***OpenSKE's Outputs***

    1. *Weaknesses Found*

        A list of all weaknesses described by the CWE that have been satisfied by the current vulnerabilities in our systems.

    2. *Executed Attack Patterns*

        This constitutes a list of the CAPEC attack patterns that have been executed on our systems due to having their requirements satisfied.

    3. *Compromised Software/Assets*

        Software that has been attacked or assets that have been accessed or destroyed are reported.

We believe that by providing the above, we will be putting ourselves on a higher ground with the necessary information, tools and techniques to understand how secure our systems are.

6. OPENSKE'S DOMAIN MODEL

The domain model has been described in the form of Java classes that are inter-related together. We have tried to keep the domain model thorough enough to identify each participating entity along with it's obvious internals and behavior. Though, we haven't tied it to any particular vendor in order to keep it as independent as possible. In our illustration [8] of the domain model we will be showing a breakdown of the UML design to explore how the entities relate together. Throughout this chapter, we will see the notion of an entity's security state. This merely indicates the entity's condition (*unknown*, *safe*, *risky* or *compromised*) from a security point of view.

6.1. Hardware Domain Model

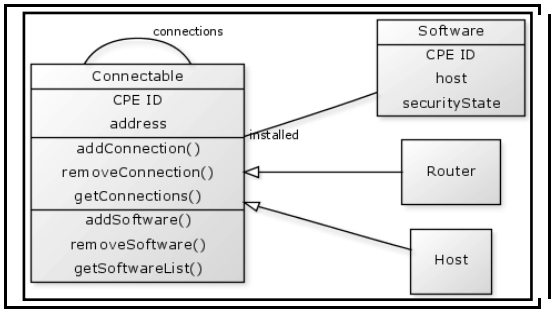


FIGURE 5 : Hardware Domain Model

Starting with the hardware domain model illustrated in Figure 5, we find that our hosts and routers inherit from a `Connectable` abstract class which provides the ability to interconnect hardware devices to each other and add software. Any possible `Connectable` subclass can contain software, the relation is also clearly shown in the diagram and has also been facilitated through the APIs. The `Host` and `Router` classes can expand, retract or override whatever functionality inherited from the `Connectable` class.

6.2. Assets Domain Model

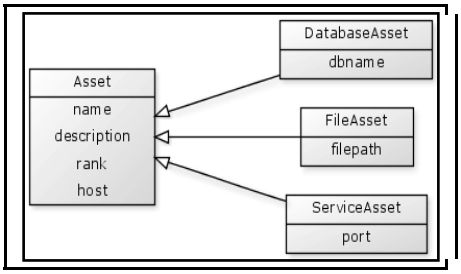


FIGURE 6 : Assets Domain Model

Next are assets, which are the most valuable resources maintained throughout the systems within an organization. Their types vary as illustrated in Figure 6 which shows different types of assets whether it's a database, file or service that needs to be secured throughout it's lifetime. The definition of an asset may be vague at times, but generally, it is anything that is important to the owning organization or person.

8 All UML diagrams presented here have been drawn using yUML (http://yuml.me)

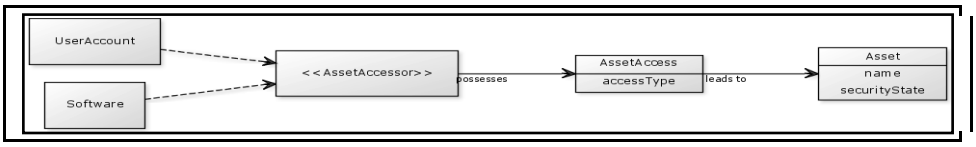


FIGURE 7 : Asset Accessibility

In addition to having our assets modeled, we had to approach how other entities would access the assets. Figure 7 shows how any entity (such as `Software` or `UserAccount`) that implements that `AssetAccessor` interface can easily possess `AssetAccesse`s to any of the available assets with respect to the `AccessType` given.

6.3. Software Domain Model

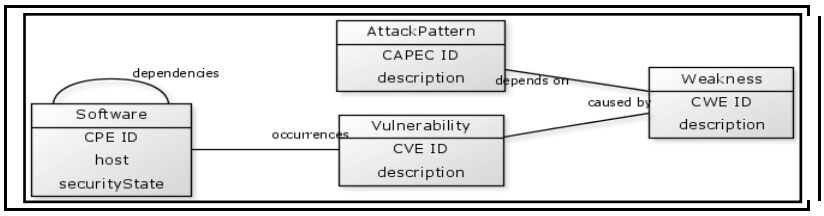


FIGURE 8 : Software & Vulnerabilities Domain Model

Software in OpenSKE is identified using their corresponding CPE identifiers to avoid any naming collisions and as illustrated in Figure 8, any software can contain occurrences of vulnerabilities identified by their CVE identifiers. Vulnerabilities are linked to their CWE weaknesses by looking up the CWE data-set. Software can depend on other software and thus, any piece of software that depends for example, on a faulty library, is potentially vulnerable as well. Implementation-wise, we have made the weaknesses of a software accessible from the software rather than having to traverse the software → vulnerabilities → weaknesses chain. This would help the reasoning to be more effective.

6.4. User Security Domain Model

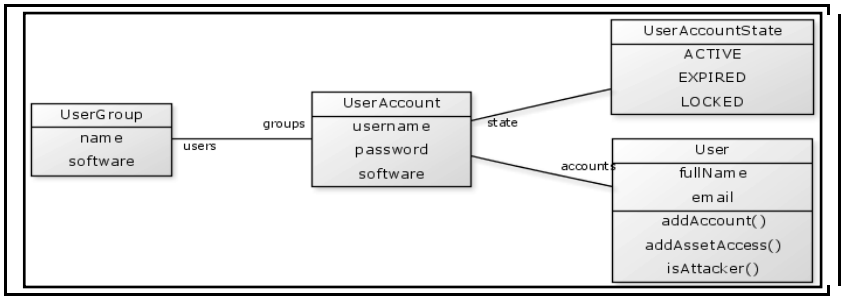


FIGURE 9 : User Security Domain Model

As shown in Figure 9, users in OpenSKE can possess multiple accounts on one or more hosts. These accounts may be further organized into groups as we usually see in most common applications. A user account is merely the credentials needed to gain an access level on a running software and it's state can be one of the listed values in Figure 9.

Now that we have covered our domain model, next we will see how we will leverage this domain model in our expert rules which will attempt to uncover more useful information from our initial understanding.

7. DEFINING THE SECURITY DOMAIN RULES

The most important possessions of an IT infrastructure are it's assets. Actually, without the assets, the systems won't be of much potential to be viable to attacks, unless the attacker intends to just gain more ground to launch further additional attacks and in this case, the hosts themselves can be considered as assets. Thus, we really need to understand where our assets stand from a security point of view. Before we delve into the rules of our system, let's quickly illustrate an example of what constitutes a Drools rule and how does the inference work over these rules.

7.1. An Anatomy of a Drools Rule

```
rule "Print assets that have become risky"
  when
    # Match an asset that has it's security state RISKY
    # Bind the matched Asset instance to the variable $asset
    $asset : Asset(
      securityState == SecurityState.RISKY
    )
  then
    print("[LOGGING] Asset '%s' has become risky !", $asset.getName());
end
```

FIGURE 10 : Drools Rule Example (Printing risky assets)

Figure 10 shows a very simple example of a Drools rule which is executed if any asset has become in a risky state (this happens when it's surrounded by vulnerable software, we'll see this later). The rule consists of two parts, the left-hand side which is the *when* part and the right-hand side which is the *then* part. The left-hand side contains the patterns to be matched upon the facts in our knowledge base. If our pattern successfully matches some facts, it is bound to the variables we have specified (in this case it's **$asset**), then the right-hand side is executed with the bound data (in this case we are printing the names of the risky assets using common Java syntax). We will illustrate a couple of rules from OpenSKE which should give a good understanding on how the rules and reasoning work.

7.2. Identifying Vulnerable Software

```
rule "Mark vulnerable software or those that depend on it as risky"
  when
    # Match any vulnerable software
    $sw : Software( vulnerabilities.size > 0 )
    or
    (
      # Match any existing risky software
      $dep : Software(
        securityState == SecurityState.RISKY
      )
      and
      # Match any software that depends on $dep
      $sw : Software(
        dependencies contains $dep
      )
    )
  then
    # Update the software as risky
    $sw.setSecurityState(SecurityState.RISKY);
end
```

FIGURE 11 : Identifying vulnerable software as risky

Figure 11 shows the rule that identifies risky software by checking if it has vulnerabilities or if it depends on any previously identified risky software. Upon having any possible facts that match the *when* part of the rule, the matched software **$sw** is updated to being risky from a security point of view, this is used later in the next section to deduce whether this risky software may affect our assets.

7.3. Identifying Assets that are Threatened

```
rule "Mark assets that are surrounded by risky software as risky"
  when
    # Match any asset available on our systems
    $asset : Asset()
    and
      exists(
        # Match any risky software on the same asset's host
        Software(
          host == $asset.host ,
          securityState == SecurityState.RISKY
        )
        or
        (
          # Match any neighbor host to the asset's host
          $neighbor : Host(
            connections contains $asset.host
          )
          and
          # Match any risky software on the neighbor host
          Software(
            host == $neighbor ,
            securityState == SecurityState.RISKY
          )
        )
      )
  then
    # Update the asset as risky
    $asset.setSecurityState(SecurityState.RISKY);
end
```

FIGURE 12 : Identifying assets surrounded by vulnerable software as risky

Figure 12 illustrates how we mark which of our assets are in jeopardy because of any surrounding vulnerable software on the same host or from a neighbor host. If we end up with a matched asset here, it is updated as being in a risky security state.

The next set of rules will be tackling a selection of some CAPEC attack patterns and how their preconditions and postconditions are modeled in OpenSKE.

7.4. Modeling the CAPEC Attack Patterns

```
rule "CAPEC-1 : Accessing Functionality Not Properly Constrained by
ACLs"
  when
    # We have an attacker
    $attacker : User(
      attacker == true
    )
    # A software that the attacker may target
    $software : Software()
    # The software contains any of the listed weaknesses
    exists(
      Weakness(
        software == $software ,
        identifier in ("CWE-285", "CWE-732",
                       "CWE-276", "CWE-693",
                       "CWE-721", "CWE-434")
      )
    )
    # Attacker has an active user account on this software
    exists(
      UserAccount(
        software == $software ,
        state == UserAccountState.ACTIVE
      ) from $attacker.getAccounts()
    )
    # The attacker can still reach this software
    eval(
      $attacker.getHost().canReach(
        $software.getHost()
      )
    )
    then
      print("[CAPEC-1] Attacker '%s' can gain un-authorized
accessibility on software '%s'", $attacker.getFullName(),
                $software.toString());
end
```

FIGURE 13 : CAPEC-1, Accessing Functionality Not Properly Constrained by ACLs

CAPEC-1 [9] which is the first attack pattern in the CAPEC dictionary shows how an attacker can gain unauthorized access to functionality that should have been protected for higher authorized people. The rule states that if we have an attacker (which is a normal `User` in OpenSKE) with an active `UserAccount` on a `Software` that contains one of the listed `Weakness`es and that the attacker can reach this software, then the attacker can possess unauthorized functionality on the target software. The CAPEC lists possible ways to mitigate the situation, but within the scope of security analysis, this may be useful when designing countermeasures to be taken against the attacks. CAPEC-1's post-conditions in specific, cannot be speculated in OpenSKE, since it highly depends on the nature of the software application being analyzed, which as you can see here, is totally unknown (i.e. software's features aren't modeled yet).

9 CAPEC-1, http://capec.mitre.org/data/definitions/1.html

```
rule "CAPEC-2 : Inducing Account Lockout"
  when
    # We have an attacker
    $attacker : User(
      attacker == true
    )
    # A software that the attacker may target
    $software : Software( accounts.size() > 0 )
    # The software contains any of the listed weaknesses
    exists(
      Weakness(
        software == $software ,
        identifier in ("CWE-400")
      )
    )
    # The software has any active user account
    $userAccount : UserAccount(
      software == $software ,
      state == UserAccountState.ACTIVE
    )
    # The attacker can reach this software
    eval(
      $attacker.getHost().canReach(
        $software.getHost()
      )
    )
    then
      # Lock the user account
      $userAccount.setState(UserAccountState.LOCKED);
      print("[CAPEC-2] Attacker '%s' has attacked the user account '%s'
on software '%s' and resulted in the account being locked",
$attacker.getFullName(), $userAccount.getUsername(),
$software.toString());
end
```

FIGURE 14 : CAPEC-2, Inducing Account Lockout

CAPEC-2 [10] involves the attacker targeting the supposedly defensive mechanism being employed in some authentication systems which is to lockout an account if it's login attempts have passed a number of tries. The rule mentions that if an attacker can reach a software with any active user accounts and that the software possesses the weakness described by CWE-400 [11], then the consequence is that the attacker can keep trying to perform multiple random logins until the account is locked (even though the original account owner had nothing to do with this).

10 CAPEC-2, http://capec.mitre.org/data/definitions/2.html
11 CWE-400, http://cwe.mitre.org/data/definitions/400.html

```
rule "CAPEC-7 : Blind SQL Injection"
  when
    # We have an attacker
    $attacker : User(
      attacker == true
    )
    # A software that the attacker may target
    $software : Software()
    # The software contains any of the listed weaknesses
    exists(
      Weakness(
        software == $software ,
        identifier in ("CWE-89", "CWE-209",
                       "CWE-74", "CWE-20",
                       "CWE-390", "CWE-697",
                       "CWE-713", "CWE-707")
      )
    )
    # The attacker can reach this software
    eval(
      $attacker.getHost().canReach(
        $software.getHost()
      )
    )
    then
      # Attacker gains access to the database assets from this software
      # We have assigned a random asset with a random access type for
      # the simulation
      $attacker.addAssetAccess(
        new AssetAccess(
          $software.getRandomAsset(AssetType.DATABASE),
          $attacker,
          AssetAccessType.getRandomValue()
        )
      );
      print("[CAPEC-7] Attacker '%s' has gained '%s' access to database
'%s' through SQL injection on software '%s'",
        $attacker.getFullName(),
        $attacker.getRecentAssetAccess().getType(),
        $attacker.getRecentAssetAccess().getAsset().getName(),
        $software.toString()
      );
end
```

FIGURE 15 : CAPEC-7 : Blind SQL Injection

The CAPEC-7 [12] is one of the common attack patterns that we see applicable to online web applications. It is similar to the previously illustrated attack patterns, but the difference lies in the consequences of the attack, which in this case, grants the attacker an `AssetAccess` to one of the `DatabaseAsset`s that the matched software possesses. The randomization performed in the value selection of the rules has been done in order to avoid fixating scenarios. In reality, this is solely up to the attacker proficiency to get the best benefits out of the attack pattern being executed.

12 CAPEC-7, http://capec.mitre.org/data/definitions/7.html

```
rule "CAPEC-16 : Dictionary-based Password Attack"
  when
    # We have an attacker
    $attacker : User(
      attacker == true
    )
    # We have an installed and running software with user accounts
    $software : Software( accounts.size > 0 )
    # The software contains any of the listed weaknesses
    exists(
      Weakness(
        software == $software ,
        identifier in ("CWE-521", "CWE-262",
                       "CWE-263", "CWE-693")
      )
    )
    # The attacker doesn't have an account on this software
    not(
      exists(
        UserAccount(
          software == $software ,
          state == UserAccountState.ACTIVE
        ) from $attacker.accounts
      )
    )
    # The attacker can reach this software
    eval(
      $attacker.getHost().canReach(
        $software.getHost()
      )
    )
    then
      # The attacker gained a user account
      $attacker.addAccount(
        $software.getRandomAccount()
      );
      print("[CAPEC-16] Attacker '%s' has hacked account '%s' on
software '%s'",
        $attacker.getFullName(),
        $attacker.getRecentAccount(),
        $software.toString()
      );
end
```

FIGURE 16 : CAPEC-16 : Dictionary-based Password Attack

CAPEC-16 [13] is one of the most commonly used attack patterns on the Internet since most people use normal words for their passwords. The attacker here is attacking a software that possesses at least one active user account and suffers from one of the listed weaknesses. Upon executing the attack, the attacker is granted an account on the targeted software. Now that we have illustrated how OpenSKE's rules work. We will be seeing their application in an experiment in the next section.

13 CAPEC-16, http://capec.mitre.org/data/definitions/16.html

8. EXPERIMENTING OPENSKE

8.1. The Experiment Setup

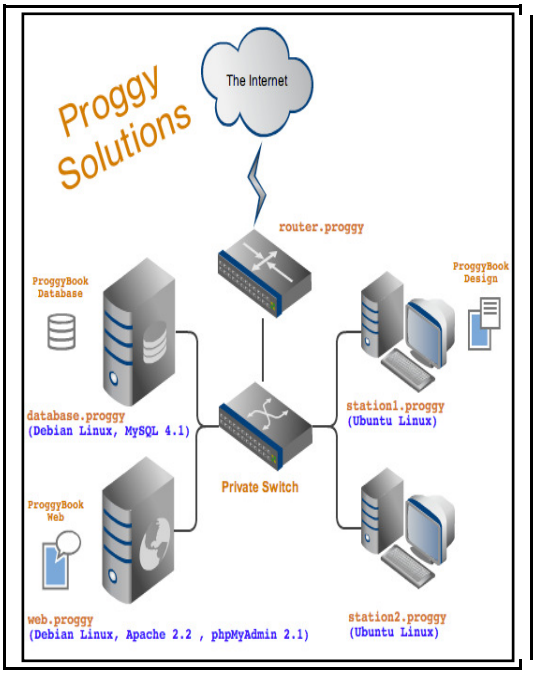


FIGURE 17 : Proggy Solutions Infrastructure [14]

Proggy Solutions is a novel startup company specialized in providing public web-services. Their most popular service, ProggyBook which is a public social-networking website running on their self-hosted Apache [15] web server and MySQL [16] database server as outlined in Figure 17. In addition to these supporting services, they use phpMyAdmin [17] as a web interface to manage their MySQL databases. Usually, there is an on-call off-site support personnel which troubleshoots any occasional issues using several tools including phpMyAdmin.

Name	***Type***	***Location***
ProggyBook Web	Service Asset	web.proggy
ProggyBook DB	Database Asset	database.proggy
ProggyBook Design	File Asset	station1.proggy

FIGURE 18 : Proggy's Valuable Assets

Figure 18 outlines Proggy's valuable assets and we will be taking these assets as our targets in this experiment. We will also assume that Proggy's systems contain the weaknesses listed below in Figure 19.

14 The infrastructure diagram was created using Gliffy (http://www.gliffy.com)
15 Apache Web Server, http://httpd.apache.org/
16 MySQL Database Server, http://www.mysql.com/
17 phpMyAdmin, http://www.phpmyadmin.net/

Software	***CWE ID***	***Description***
ProggyBook Web 1.0	CWE-20 [18]	Improper Input Validation
ProggyBook Web 1.0	CWE-285 [19]	Improper Access Control
ProggyBook Web 1.0	CWE-400 [20]	Uncontrolled Resource Consumption
phpMyAdmin 2.1	CWE-521 [21]	Weak Password Requirements

FIGURE 19 : Proggy's Systems' Weaknesses

The experiment was run on a casual Apple MacBook Pro with the following specifications.

Processor	2.53 GHz
Memory	4 GiB
Java Virtual Machine	Java(TM) SE Runtime Environment 1.6.0_20
Drools Expert System	5.1.1

FIGURE 20 : Platform Specifications

8.2. Output Results

The execution of OpenSKE resulted in uncovering the consequences of the assumed weaknesses mentioned in section 8.1. Figure 21 shows the detailed output of the experiment execution.

```
~/Workspaces/OpenSKE/openske > ./openske

>> Building OpenSKE...

>> Running OpenSKE's console...

Welcome to OpenSKE (JVM: 1.6.0_20) !
Type 'help' for help
openske> start

[OPENSKE] Running OpenSKE engine...

[OPENSKE] Initializing Drools Knowledge Base...

[OPENSKE] Loading 4 rule files...
- Loading './openske-expertise/src/main/resources/com/openske/rules/Assets.drl'
- Loading './openske-expertise/src/main/resources/com/openske/rules/Attack Patterns.drl'
- Loading './openske-expertise/src/main/resources/com/openske/rules/Logging.drl'
- Loading './openske-expertise/src/main/resources/com/openske/rules/Software.drl'

[OPENSKE] Adding 1 compiled knowledge packages to the knowledgebase...
         - Knowledge Package com.openske.rules (9 rules)

[OPENSKE] Inserting the facts into the knowledge base...

[DROOLS] Activation Created : Detect the reachability of an attacker ( if any )
[DROOLS] Activation Created : Detect the reachability of an attacker ( if any )
[DROOLS] Activation Created : Detect the reachability of an attacker ( if any )
[DROOLS] Activation Created : CAPEC-2 : Inducing Account Lockout
```

18 CWE-20, http://cwe.mitre.org/data/definitions/20.html
19 CWE-285, http://cwe.mitre.org/data/definitions/285.html
20 CWE-400, http://cwe.mitre.org/data/definitions/400.html
21 CWE-521, http://cwe.mitre.org/data/definitions/521.html

```
[DROOLS] Activation Created : CAPEC-7 : Blind SQL Injection
[DROOLS] Activation Created : Detect the reachability of an attacker ( if any )
[DROOLS] Activation Created : CAPEC-16 : Dictionary-based Password Attack
[DROOLS] Activation Created : Detect the presence of an attacker

[OPENSKE] Firing all rules...

[DROOLS] Activation Fired : Detect the presence of an attacker
[LOGGING] Attacker 'Mr. X' detected on host 'attacker.proggy'

[DROOLS] Activation Fired : CAPEC-16 : Dictionary-based Password Attack
[CAPEC-16] Attacker 'Mr. X' has breached account 'admin' on software
'cpe:/a:phpmyadmin:phpmyadmin:2.1' through a dictionary-based password attack

[DROOLS] Activation Fired : Detect the reachability of an attacker ( if any )
[LOGGING] Attacker 'Mr. X' can reach software 'cpe:/a:proggysolutions:proggyweb:1.0'

[DROOLS] Activation Fired : CAPEC-7 : Blind SQL Injection
[CAPEC-7] Attacker 'Mr. X' has gained 'READ_WRITE' access to database 'ProggyBook
Database' through SQL injection on software 'cpe:/a:proggysolutions:proggyweb:1.0'

[DROOLS] Activation Fired : CAPEC-2 : Inducing Account Lockout
[CAPEC-2] Attacker 'Mr. X' has attacked the user account 'admin' on software
'cpe:/a:proggysolutions:proggyweb:1.0' and resulted in the account being locked

[DROOLS] Activation Fired : Detect the reachability of an attacker ( if any )
[LOGGING] Attacker 'Mr. X' can reach software 'cpe:/a:apache:apache:2.2'

[DROOLS] Activation Fired : Detect the reachability of an attacker ( if any )
[LOGGING] Attacker 'Mr. X' can reach software 'cpe:/a:phpmyadmin:phpmyadmin:2.1'

[OPENSKE] Engine took 4.21 seconds !
```

FIGURE 21 : OpenSKE's experiment output.

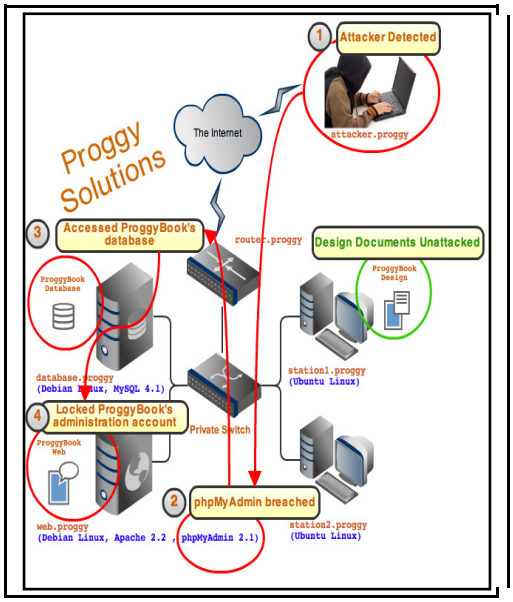


FIGURE 22 : OpenSKE's experiment output manual visualization

We have highlighted the important sections of the output which indicate the deductions made by OpenSKE in addition to mapping them on the infrastructure diagram in Figure 22. We will see that the first rule fired was the detection of the attacker *Mr. X* at `attacker.proggy`. The attacker attempted a dictionary-attack on the public phpMyAdmin interface and since phpMyAdmin held user accounts with default passwords (such as root/root or admin/admin) it was very easy for

the attacker to gain access to the user accounts available. The attacker then attempted a SQL injection attack on the ProggyBook website which resulted in a READ_WRITE [22] access to the ProggyBook database. In addition to this, the ProggyBook developers thought it was a good idea to lock accounts on the third login failure. The attacker took advantage of this and attacked the admin account which resulted in the administration section being locked.

The total time taken for the initial execution was **4.21** seconds in performing the following items.

1. Initializing the knowledge base
2. Loading the rules
3. Inserting the facts
4. Firing the rules and updating the knowledge base accordingly

OpenSKE took way lesser time in consequent runs (by doing a `restart` from the OpenSKE console), due to the optimizations that are constantly performed by the Java Virtual Machine throughout the execution.

9. COMPARATIVE EVALUATION

In order to see how valuable the deductions of OpenSKE are, let us consider how do the results shown in the previous section compare to the findings of previous efforts.

9.1. The Execution Model

- The MulVal framework [9] operates in the form of a question-based approach. For example, after it's initialization, we start asking whether the attacker can execute code on a private server. When the backward-chaining process has been done, it replies with either the attack trace or none.
- The Model-Checking approach [5] operates after being provided with a system correctness condition that the model checker must stop upon having this condition no longer satisfiable. When the condition becomes violated, a counter-example is returned with the state transitions sequence that led to the system correctness violation.
- The OpenSKE framework operates in the form of an online approach which upon any update in the knowledge base, fires the rules waiting for such update which consequently performs additional deductions which is fed into the knowledge base again. Execution ends when no more deductions can be reached.

9.2. Syntactic Clarity

- The MulVal framework [9] is written in Datalog, thus any form of knowledge is represented as Datalog tuples similar to what we know from first-order logic. Object relations between the domain entities are defined in the predicate names, thus we can write as much as relations as possible as long as we can distinguish them clearly. Writing the rules requires reusing previous predicates in a recursive manner to achieve the desired understanding.
- The Model-Checking approach [5] was exercised using the Symbolic Model Verifier (SMV) [23] which had the system represented in arrays of booleans and literals. The initialization of the model had to be done with care in specifying the initial and next states

22 This was a randomly chosen access type for the SQL injection as it's totally based on the attempted SQL which isn't currently modeled in OpenSKE.

23 Symbolic Model Verifier, http://www.cs.cmu.edu/~modelcheck/smv.html

of each attribute. The original authors sacrificed the resulting program's clarity due to the limited feature set of the SMV checker.

- The OpenSKE framework leverages the Java programming language in order to represent the domain model in an object-oriented paradigm which allows modeling the domain in a highly precise and accurate manner. The rules are described in Drools's simple readable rule syntax which helps in comprehending the rule logic easily.

9.3. Scalability with Larger Infrastructures

- The MulVal framework [9] was benchmarked by it's authors and the results were impressive since MulVal was achieving the analysis in less than 1 second up until 400 hosts which then rises to 3.85 seconds with 1000 hosts.
- The Model-Checking approach [5] wasn't tested with a larger infrastructure than the one attempted by the original authors, but it was noted that the model checking approach may face a state-space explosion with a large number of state possibilities.
- The OpenSKE framework has been benchmarked with several infrastructure sizes and the results are shown in Figure 23.

Infrastructure	Rules Load Time	Facts Load Time	Deductions Time
Proggy Solutions	0.48	0.09	0.01
200 Hosts	0.44	0.76	0.01
500 Hosts	0.47	3.08	0.01
1000 Hosts	0.42	15.06	0.03

FIGURE 23 : OpenSKE Benchmark Results (Time unit is seconds)

The timings shown in Figure 23 have been collected by running OpenSKE in `benchmark` mode for each infrastructure size. In `benchmark` mode, only high-level statistical output is written to the console to minimize any irrelevant any I/O. The reason why the deduction times are very negligible is because as soon as the facts are inserted, the rule matching is performed and the supposed rules to run are registered as to be activated once the rule firing is signaled.

10. OPENSKE'S CURRENT SHORTCOMINGS

Unfortunately, with the bright side shown in OpenSKE, it still suffers from a set of shortcomings which are listed below.

1. The domain model can be heavily expanded to further devices, systems and relations. The more we try to diversify, the more it will become applicable to more use cases.
2. We haven't mentioned how *time* plays an important role in the execution of attacks. Time has always been a crucial factor in many security related events. Temporal reasoning can facilitate this and OpenSKE can be further expanded to leverage the Drools Fusion [24] component in order to accomplish this.
3. We have supplied OpenSKE with Proggy's systems information in a manual manner, this can be automated by detecting the network topology and the running systems on a frequent basis.
4. Rules were being fired in an haphazard sequence which may at times make it incomprehensible, thus guiding them to go through a workflow process will highly organize the steps of OpenSKE's execution. OpenSKE can leverage Drools Flow [25] in order to achieve this.

24 Drools Fusion, http://www.jboss.org/drools/drools-fusion.html
25 Drools Flow, http://www.jboss.org/drools/drools-flow.html

5. We haven't considered retracting facts yet, but in reality this is unavoidable. For example, shutting down hosts or running services will dramatically change whatever has been deduced before, or at least prevent possible deductions to be made.

11. CONCLUSION

We envision that OpenSKE can be integrated with various security sentinels such as intrusion detection tools, firewalls, in which OpenSKE acts as the brain sitting at the back making sense of what's happening in order to take possible actions or provide the activities in a comprehensible manner to the administrators. Although our current ruleset has mostly described the CAPEC attack patterns, we believe further wisdom can be captured from security experts and formalized in order to deduce more in-depth meanings.

Finally, we highly welcome fellow researchers in the security analysis field to leverage the publicity [26] of OpenSKE's implementation in researching different topics and building further tools.

12. ACKNOWLEDGMENTS

Finally, I would like to thank my family for supporting me with all they had such that I would accomplish my research. My mentors have constantly provided me with wise and concise advice on various matters. Finally, I would like to thank the reviewers for their patience and effort in providing insightful input.

13. REFERENCES

5. Steven J. Templeton, Karl Levitt. “A Requires/Provides Model for Computer Attacks”. ACM Press, 2000.

6. Robert W. Baldwin. “Rule based Analysis of Computer Security”. MIT, 1987.

7. Daniel Farmer, Eugene H. Spafford. “The COPS Security Checker System”. Purdue, 1994.

8. Dan Zerkle, Karl Levitt. NetKuang – “A Multi-Host Configuration Vulnerability Checker”, California, 1996.

9. Ronald W. Ritchey, Paul Ammann. “Using Model Checking to Analyze Network Vulnerabilities”. IEEE Symposium on Security and Privacy, 2000.

10. R. P. Lippmann, K. W. Ingols. “An Annotated Review of Past Papers on Attack Graphs”. MIT 2005.

11. Xinming Ou. “A logic-programming approach to network security analysis”. Princeton University, 2005.

12. Sudhakar Govindavajhala. “A Formal Approach to Practical Network Security Management”. Princeton University, 2006.

13. Xinming Ou, Sudhakar Govindavajhala, Andrew W. Appel. “MulVAL: A Logic-based Network Security Analyzer”. Proceedings of the 14th USENIX Security Symposium, 2005.

14. Edward A.Feigenbaum. “Expert Systems : Principles and Practice”, The Encyclopedia of Computer Science and Engineering, 1992.

26 OpenSKE, http://code.google.com/p/openske

15. CL Forgy, Rete: “A fast algorithm for the many pattern/many object pattern match problem”. Artificial Intelligence, 1982.

16. Robert A. Martin. “Making Security Measurable and Manageable”, MILCOM 2008.

17. T. Tidwell, R. Larson, K. Fitch and J. Hale. “Modeling Internet Attacks”, IEEE 2001.

18. Sean Barnum, Amit Sethi. “Attack Patterns as a Knowledge Resource for Building Secure Software”, OMG Software Assurance Workshop: Cigital, 2007.

A Genetic Algorithm for Reliability Evaluation of a Stochastic-Flow Network With Node Failure

A. Younes a_y_hamed@yahoo.com
Computer Science Department,
Faculty of Science, Sohag University,
Sohag, Egypt

M. R. Hassan m_r_hassan73@yahoo.com
Computer Science Department,
Faculty of Science, South Valley University,
Aswan, Egypt.

Abstract

The paper presents a genetic algorithm to compute the reliability of a stochastic- flow network in which each arc or node has several capacitis and may fail. I.e. Calculate the system reliability such that the maximum flow is not less than a given demand. The algorithm is based on generating all lower boundary points for the given demand and than the system reliability can be calculated in terms of such points. The proposed algorithm can be used for a network with large number of nodes and links. Also, the paper investigates the problems that are found in the solutions that obtained by using other previous methods.

Keywords: Genetic Algorithms, Stochastic-flow Network, System Reliability.

1. INTRODUCTION

The reliability of a computer network (in the case of no flow happen) is defind as probability that any source node can successfully communicate with any terminal node. This basically gives the probability that no node in a network is disconnected from the rest of the network and it doesn't yeild any information about the performance aspects, Ahuja and Kumar [1]. Rai and Aggarwal [2], preseneted a method for finding the terminal-pair reliability expression of a general network. Rai and Aggarwal [3], defined the network reliability for a computer-communication network and proposed a method based on spanning trees for its evaluation. Younes [4], presented an algorithm to find the spanning trees of a given network in terms of links for using them to compute the network reliability. The algorithm uses the connection matrix of a given network to find the spanning trees, and also is based on a relation that uses the probability of unions of the spanning trees to obtain the network reliability.

The system reliability of a flow network is the probability that the maximum flow of the network is not less than the given demand d. The capacity of each arc in this network, which is defined as the maximum flow passing the arc per unit time, has two levels 0 and positive integer. Lin [5], proposed a simple algorithm to generate all lower boundary points for a given demand d, and then the system reliability can be calculated in terms of such points. Lin [6], constructed a stochastic-flow network to model the information system. The studied problem is to evaluate the possibility (them mission reliability) that a given amount of multicommodity can be sent through an information network under the cost constraint. Lin [7], proposed a new performance index, the probability that the upperbound of the system capacity equals the demand vector subject to the budget constraint, to evaluate the quality level of a multicommodity limited-flow network. Statitsatin and Kapur [8], presented an algorithm to search for lower boundary points and used it for computing the exact reliability. Lin and Yeh [9], the network reliability under a components-assignment can be computed in terms of

minimal paths, and state-space decomposition. Subsequently, they proposed an optimization method based on a genetic algorithm.

In recent years, gentetic algorithms (GAs) have been applied to various problems in the network design ([10-16]). Lo and Chang [17], proposed a multiojective hybrid genetic algorithm to solve the capacitated multipoint network design problem.

In this paper, a gentetic algorithm (GA) is proposed to evaluate the reliability of a stochastic flow network. The proposed GA is based on minimal paths (MPs) to find all lower boundary points for d and then calculate the system reliability in terms of such points.

The paper is organized as follows: The assumptions and notation used given in Section 2. Section 3 describes the problem of calculating the network reliability. Section 4 presents the proposed algorithm for calculating the network reliability. The over all algorithm presented in section 5. In Section 6 shows how to use the proposed algorithm to calculate the reliability of a stochastic-flow network for two example networks and presents the discussion.

2. NOTATIONS and ASSUMPTIONS

Notations:

G(A, N, M) A stochastic-flow network with a set of arcs **A=** $\{a_i \mid 1 \le i \le n\}$, a set of nodes $N = \{a_i \mid n+1 \le i \le n+p\}$ and $M = \{M^1, M^2, ..., M^{n+p}\}$ with M^i (an integer) being the maximum capacity of each component a_i (arc or node).

X Capacity vector; X= (x_1, x_2, …, x_{n+p}).

F Flow vector; F = (f_1, f_2, …, f_m).

MPs Minimal paths.

mp_j Is a minimal path no. j; j = 1, 2, …, m.

L_j Is the maximum capacity of mp_j; $L_j = \min\{M^i \mid a_i \in mp_j\}$.

V(X) The maximum flow under X; $V(X) = \max\{\sum_{j=1}^{m} f_j \mid F \in U_x\}$, where U_x = {F | F is feasible under X}.

R_d System reliability to the given demand d.

pop_size is the population size.

max_gen is the maximum number of generations.

p_m is the GA mutation rate.

p_c is the GA crossover rate.

Assumptions:

1- The capacity of each component a_i is an integer-valued random variable which takes values $0 < 1 < 2 < ... < M^i$ according to a given distribution..

2- Flow in G must satisfy th so-called flow-conservation law.
the processing elements, and edges denote the communication links.

3. The capacities of different components are statistically independent.

3. PROBLEM DESCRIPTION

Given the demand d, The system reliability R_d is defind by, Lin [5]:

$$R_d = \Pr\{X \mid V(X) \ge d\} \quad \ldots(1)$$

Where X is a lower boundary point for d.

And X can be deduced from F = (f_1, f_2, …, f_m) by using the following equation:

$$x_i = \sum_{j=1}^{m} \{f_j \mid a_i \in mp_j\} \text{ for each i = 1, 2, …, n+p.} \quad \ldots(2)$$

So, the main purpose of the proposed GA in this paper is to find the set of all feasible solutions of F that satisfies the flowing two constraints:

$$\sum_{j=1}^{m}\{f_j \mid a_i \in mp_j\} \le M^i \text{ for each i = 1, 2, …, n+p,} \quad \ldots(3)$$

$$\sum_{j=1}^{m} f_j = d\,. \quad \ldots(4)$$

4. THE PROPOSED GENETIC ALGORITHM

This section describes the basic components of the proposed GA.

1. REPRESENTATION

If the network has m number of minimal paths, then the chromosome CH has m fields, each field represents the (current) flow on each path. I.e.

$$CH = (f_1, f_2, \ldots, f_m), \quad f_j \text{ is current flow on } mp_j.$$

2. INITIAL POPULATION

The initial population is generated according to the following steps:

Step 1: Randomly generate a chromosome CH in the initial population in the form:

$$CH = (f_1, f_2, \ldots, f_m)$$

where $f_i \in \{0,1,...,d-1\}$; d is the given demand.

Step 2: If the generated chromosome in step 1 doesn't satisfy eq. 4, discard it and go to step 1.

Step 3: Repeat steps 1 to 3 to generate pop_size chromosomes.

3. THE OBJECTIVE FUNCTION

The problem can be formulated as:

Find the set of all feasible solutions F

Such that equations 3 and 4 have been satisfied

4. Crossover Operator

In the proposed GA, one-cut point crossover (i.e. an integer value is randomly generated in the range (0, m-1) where m is the length of the chromosome) is used to breed two offsprings (two new chromosomes) from two parents selected randomly according to pc value, as shown in the flowing example (The network has 5 MPs):

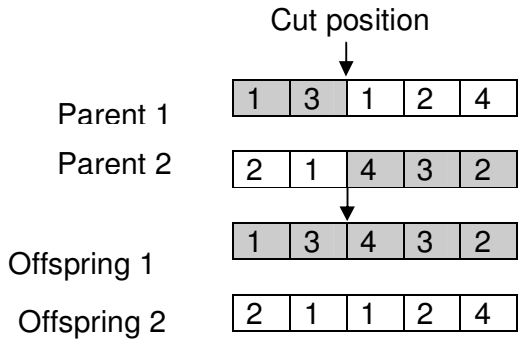


FIGURE 1: Single point crossover

5. MUTATION OPERATOR

A child undergoes mutation according to the mutation probability P_m .

Step 1: Generate a random number r_m, $r_m \in [0,1]$.

Step 2: If $r_m < P_m$, the chromosome is chosen to mutate and go to step 3; otherwise skip this chromosome.
Step 3: For each component of the child do:
Begin;
if $f_i >= 1$, then set $f_i = 0$.
Else
set $f_i = 1$.
End do.

Fig.2 shows the an example of performing the mutation operation on a given chromosome:

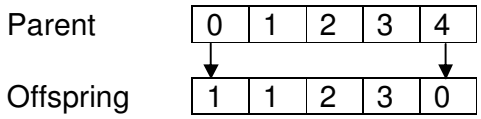


FIGURE 2: The mutation operator.

6. TERMINATION CONDITION

The execution of the GA is terminated when the number of generations exceeds the specified number of maximum generations or the set of all feasible solutions of F have been generated.

5. AN ALGORITHM FOR SYSTEM RELIABILITY EVALUATION

Given the demand **d**, then $\mathbf{R_d = Pr\{X \mid X \geq X^i}$ for a lower boundary point $\mathbf{X^i}$ for **d** $\mathbf{\}}$ = $\mathbf{Pr\{\cup_{all\ lower\ boundary\ points\ X^i\ for\ d}\{X \mid X \geq X^i\}\}}$, i = 1, 2, …, q and q is the total number of lower boundary points for **d**. Applying an inclusion-exclusion rule to compute $\mathbf{Pr\{\cup_{all\ lower\ boundary\ points\ X^i\ for\ d}\{X \mid X \geq X^i\}\}}$, see [5]. The following algorithm is presented in this paper to calculate $\mathbf{R_d}$ according to the bove rules:-

Step 1: Generate all possible intersections for all lower boundary points X.

Step 2: Calculate the probability (accumulative probability) for each **X** and also for each intersection.

Step 3: Calculate $\mathbf{R_d}$ as follows:

Set $\mathbf{B_1 = \{X \mid X \geq X^1\}, B_2 = \{X \mid X \geq X^2\}, ..., B_q = \{X \mid X \geq X^q\}.}$

Apply inclusion-execlusion rule to calculate $\mathbf{R_d}$ by using the relation**:**

$$\mathbf{R_d = \sum_i Pr\{B_i\} - \sum_{i \neq j} Pr\{B_i \cap B_j\} + \sum_{i \neq j \neq k} Pr\{B_i \cap B_j \cap B_k\} - ... +}$$
$$\mathbf{+ (-1)^{q-1} Pr\{B_1 \cap B_2 \cap ... \cap B_q\}}$$

6. THE OVERALL ALGORITHM

This section presents the proposed GA for computing the system reliability of a stochastic-flow network. The steps of this algorithm are as follows:

Step 1: Set the parameters: pop_size, max_gen, P_m , P_c, ns = 0, and set

gen = 0.

Step 2: Generate the initial population, as described in section 4.1.

Step 3: To obtain chromosomes for the new population; select two chromosomes from the parent population according to P_c. Apply crossover, then mutate the new child according to P_m parameter.

Step 4: If the new child satisfies the two constraints (equations 3 and 4) and it doesn't equal to any pre generated child, then keep it and increase **ns**. If it fails to satisfy them, discard this child and reapply the mutation opereator to the original parent.

Step 5: Set gen =gen + 1;

If gen > max_gen, then go to step 3, otherwise goto step 3 /* to get a new generation.

Step 6: Report the set of all feasible solutions (equal to ns) F, and generate X from F using eq. 2.

Step 7: Suppose the result of step 6 is: X^1, X^2, …, X^q. Then, obtain all lower boundary points for d by removing non-minimal ones in X^1, X^2, …, X^q.

Step 8: Calculate R_d according to the algorithm given in section 5 and print out the results.

7. EXPERIMENTAL RESULTS & DISCUSSION

This section shows how to use the proposed GA to calculate the reliability of a stochastic-flow network for an example network with two different demands and presents the discussion of the obtained solutions.

1. EXPERIMENTAL RESULTS

To illustrate the proposed algorithm for computing the system reliability, consider the flowing example network shown in Fig. 3 taken from [5]. This network has 4 nodes and 6 arcs. The arcs are numbered from a_1 to a_6 and the nodes from a_7 to a_{10}. The same capacity distribution of each component given in [5] will be used to calculate R_5 (i.e. the demand d = 5).

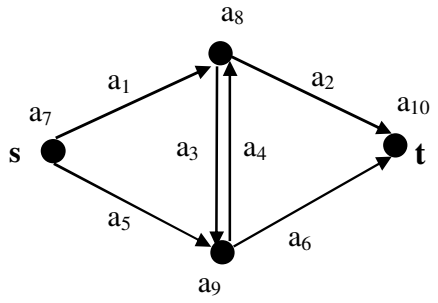


FIGURE 3: Computer network

There are 4 MPs :

$mp_1 = \{a_7, a_1, a_8, a_2, a_{10}\}$
$mp_2 = \{a_7, a_1, a_8, a_3, a_9, a_6, a_{10}\}$
$mp_3 = \{a_7, a_5, a_9, a_6, a_{10}\}$
$mp_4 = \{a_7, a_5, a_9, a_4, a_8, a_2, a_{10}\}$

The proposed algorithm will generate all lower boundary points for the demand **d = 5** as follows:

Step 1: Find the set of all feasible solution of **F = (f_1, f_2, f_3, f_4)** according to the constraints (3) and (4);

$$\mathbf{f_1+f_2\le 2, f_1+f_4\le 3, f_2\le 2, f_4\le 3, f_3+f_4\le 3, f_2+f_3\le 3,} \quad ...(3)$$
$$\mathbf{f_1+f_2+f_3+f_4\le 6, f_1+f_2+f_4\le 5, f_2+f_3+f_4\le 4, f_1+f_2+f_3+f_4\le 5.}$$
$$\mathbf{f_1+f_2+f_3+f_4\le 5.} \quad ...(4)$$

There are four solutions:

(2 0 2 1), (1 1 2 1), (2 0 3 0) and (1 1 1 2)

Step 2: Transform F into X= (x_1, x_2, …, x_{10}) according to eq. 2.

$x_1 = f_1 + f_2$, $x_2 = f_1 + f_4$, $x_3 = f_2$, $x_4 = f_4$, $x_5 = f_3 + f_4$, $x_6 = f_2 + f_3$,
$x_7 = f_1 + f_2 + f_3 + f_4$, $x_8 = f_1 + f_2 + f_4$, $x_9 = f_2 + f_3 + f_4$, $x_{10} = f_1 + f_2 + f_3 + f_4$.

The set of X that can be obtained from F is:

X^1 = (2 3 0 1 3 2 5 3 3 5)
X^2 = (2 2 1 1 3 3 5 3 4 5)
X^3 = (2 2 0 0 3 3 5 2 3 5)
X^4 = (2 3 1 2 3 2 5 4 4 5)

Step 3: Obtain all lower boundary points for **d = 5** by removing non-minimal ones in { X^1, X^2, X^3, X^4}, using the same algorithm in [5]. The only two vectors are lower boundary points for **d = 5:**

X^1 = (2 3 0 1 3 2 5 3 3 5) and X^3 = (2 2 0 0 3 3 5 2 3 5)

Step 4: Calculate the system reliability for the given demend **d = 5** according to the algorithm given in **Section 5**:

Set $\mathbf{B_1 = \{X | X \ge X^1\}}$ and $\mathbf{B_2 = \{X | X \ge X^3\}}$.

Apply inclusion-execlusion rule to calculate $\mathbf{R_5}$ by using the relation**:**

$$\mathbf{R_5 = Pr\{B_1\} + Pr\{B_2\} - Pr\{B_1 \cap B_2\}}$$
$$\mathbf{= Pr\{X | X \ge (2,3,0,1,3,2,5,3,3,5)\} + Pr\{X | X \ge (2,2,0,0,3,3,5,2,3,5)\}}$$
$$\mathbf{- Pr\{X | X \ge (2,3,0,1,3,3,5,3,3,5)\}}$$

= 0.824242

Simailarly, The system reliability for the demend d = 4 is: R_4 = 0.889351. And, the results of F, X, and the set of lower boundary points for that demand is given as follows:
The set of all feasible solutions of F is:

(2, 0, 2, 0), (1, 1, 2, 0), (1, 0, 3, 0), (2, 0, 1, 1), (1, 1, 1, 1),
(0, 2, 1, 1), (1, 0, 2, 1), (0, 1, 2, 1), (1, 1, 0, 2), (0, 2, 0, 2),
(1, 0, 1, 2), (0, 1, 1, 2), (0, 1, 0, 3)

The set of X that can be obtained from F is:

X^1 = (2, 2, 0, 0, 2, 2, 4, 2, 2, 4)
X^2 = (2, 1, 1, 0, 2, 3, 4, 2, 3, 4)
X^3 = (1, 1, 0, 0, 3, 3, 4, 1, 3, 4)
X^4 = (2, 3, 0, 1, 2, 1, 4, 3, 2, 4)
X^5 = (2, 2, 1, 1, 2, 2, 4, 3, 3, 4)
X^6 = (2, 1, 2, 1, 2, 3, 4, 3, 4, 4)
X^7 = (1, 2, 0, 1, 3, 2, 4, 2, 3, 4)
X^8 = (1, 1, 1, 1, 3, 3, 4, 2, 4, 4)
X^9 = (2, 3, 1, 2, 2, 1, 4, 4, 3, 4)
X^{10} = (2, 2, 2, 2, 2, 2, 4, 4, 4, 4)
X^{11} = (1, 3, 0, 2, 3, 1, 4, 3, 3, 4)
X^{12} = (1, 2, 1, 2, 3, 2, 4, 3, 4, 4)
X^{13} = (1, 3, 1, 3, 3, 1, 4, 4, 4, 4)

The set of lower boundary points is:

X^1 = (2, 2, 0, 0, 2, 2, 4, 2, 2, 4),
X^2 = (2, 1, 1, 0, 2, 3, 4, 2, 3, 4)
X^3 = (1, 1, 0, 0, 3, 3, 4, 1, 3, 4),
X^4 = (2, 3, 0, 1, 2, 1, 4, 3, 2, 4)
X^7 = (1, 2, 0, 1, 3, 2, 4, 2, 3, 4),
X^{11} = (1, 3, 0, 2, 3, 1, 4, 3, 3, 4),

Note: For the studied network example, the parameters setting in the proposed algorithm are:
The population size (pop_size) = 50
The GA crossover rate (p_c) = 0.95
The GA mutation rate (p_m) = 0.05
The maximum number of generations (max_gen) =1000.

The following table summerizes the results of applying the proposed algorithm on another network example taken from [18]:

The demand (d)	The set of lower boundary points	The reliability
2	(0, 1, 0, 1, 2, 1, 2, 1, 2, 2) (1, 1, 0, 0, 1, 1, 2, 1, 1, 2) (2, 1, 1, 0, 0, 1, 2, 2, 1, 2) (1, 2, 0, 1, 1, 0, 2, 2, 1, 2) (1, 0, 1, 0, 1, 2, 2, 1, 2, 2,)	0.958636
3	(1, 1, 0, 0, 2, 2, 3, 1, 2, 3) (2, 3, 0, 1, 1, 0, 3, 3, 1, 3) (2, 0, 2, 0, 1, 3, 3, 2, 3, 3) (1, 0, 1, 0, 2, 3, 3, 1, 3, 3) (0, 2, 0, 2, 3, 1, 3, 2, 3, 3) (1, 3, 0, 2, 2, 0, 3, 3, 2, 3) (0, 1, 0, 1, 3, 2, 3, 1, 3, 3) (3, 2, 1, 0, 0, 1, 3, 3, 1, 3) (1, 2, 0, 1, 2, 1, 3, 2, 2, 3) (2, 1, 1, 0, 1, 2, 3, 2, 2, 3) (2, 2, 0, 0, 1, 1, 3, 2, 1, 3)	0.946564

TABLE 1: The results of network example taken from [18]

2. DISCUSSION

This section investigates the problem of the obtained solution to the above examples given in Lin[5].

For the given demand **d = 5** and according to [5], the set of all feasible solutions of F is:
(2 0 2 1), (1 1 2 1), (2 0 3 0), (1 1 1 2), (0 2 1 2) and (0 2 0 3)

But, the last two solutions (0 2 1 2) and (0 2 0 3) don't satisfy the costraint:

$$f_2 + f_3 + f_4 \leq 4 .$$

So, the two solutions (0 2 1 2) and (0 2 0 3) must be eliminated and the set of solutions become:

(2 0 2 1), (1 1 2 1), (2 0 3 0) and (1 1 1 2)

Which satisfy the constraints 3 and 4 and compatable with the sloution obtained by the proposed GA when comparing the set of lower boundary points and the reliability value.

8. CONCLUSION & FUTURE WORK

This paper presented a genetic algorithm to calculate the system reliability of a stochastic-flow network to given demand d. The algorithm is based on determining the set of all feasible solutions of the flow vector and generate the set of all lower boundary points for the given demand d and then calculate the reliability. Finally we illustrate the using of the proposed algorithm by calculating the reliability of a flow network to given sample network taken from literature. Also, The algorithm is efficeint and may be extended to compute the reliability of a flow network in two or multicommodity cases.

9. REFERENCES

1. Ahuja Sanjay P. and Kumar Anup "Reliability and Performance based measure for computer networks and distributed systems", Proceedings of the IEEE SoutheastCon Conference, Charlotte, NC, 1993.

2. Rai Suresh and Aggarwal K. K., "An efficient method for reliability evaluation of a general network", IEEE Transactions on Reliability, 27(3): 1978.

3. Aggarwal K. K. and Rai Suresh, "Reliability evaluation in computer communication networks", IEEE Transactions on Reliability, 30(1): 1981.

4. Younes A., "Spanning trees and reliability of computer networks", Egyptian Informatics Journal, 6(1): 2005.

5. Yi-Kuei Lin, "A simple algorithm for reliability evaluation of a stochastic-flow network with node failure", Computers and Operations research, 28: 1277-1285, 2001.

6. Yi-Kuei Lin, "Reliability evaluation for an information network with node failure under cost constraint", IEEE Transactions on Systems, Man. And Cybernetics, Part A: System and Humans, 37(2): 180-188, 2007.

7. Yi-Kuei Lin, "System reliability of a limited-flow network in multicommodity case", IEEE Transactions on Reliability, 56(1): 17-25, 2007.

8. Satitsatian Sarintip and Kapur Kailash C., "An algorithm for lower reliability bounds of multistate two-terminal networks", IEEE Transactions on Reliability, 55(2): 199-206, 2006.

9. Yi-Kuei Lin and Cheng-Ta Yeh, ''Evaluation of Optimal Network Reliability Under Components-Assignments Subject to a Transmission Budget", IEEE Transactions on Reliability, 59(3), 2010.

10. Altiparmark Fulya, Dengiz Berna and Smith Alice E.,"Reliability optimization of computer communication networks using genetic algorithms", Proceedings of the 1998 IEEE International Conference on Systems, Man, and Cybernetics-Intelligent Systems For Humans In A Cyberworld, SMC'98, Hyatt Regency La Jolla, San Diego, California, USA, October 11-14, pp. 4676-4681, 1998.

11. Coit David W. and Smith Alice E., "Use of a genetic algorithm to optimize a combinatorial reliability design problem", Proceeding of the Third IIE Research Conference, 467-472, 1994.

12. Coit David W. and Smith Alice E., "Penalty guided genetic search for reliability design optimization", Accepted to Computers and Industrial Engineering, Special Issue on Genetic Algorithms Vol. 30(4): 1996.

13. Dengiz Berna, Altiparmark Fulya and Smith Alice E., “A genetic algorithm approach to optimal topological design of all terminal networks”, Intelligent Engineering Systems Through Artificial Neural Network, 5: 405-410, 1995.

14. Dengiz Berna, Altiparmak Fulya, Smith Alice E., “Local search genetic algorithm for optimization of highly reliable communications networks”, IEEE Transactions on Evolutionary Computation, 1: 179-188, 1997.

15. Dengiz Berna, Altiparmark Fulya and Smith Alice E., “Genetic algorithms design of networks considering all-terminal reliability”, The Sixth Industerial Engineering Research Conference Proceedings IERC’97, Miami Beach, Florida, USA, May 17-18, pp. 30-35, 1997.

16. Dengiz Berna and Aabbas C., “A simulated annealing algorithm for design of computer communication networks”, In Proceedings of world Multiconference on Cybernetics and Informatics-SCI 2001, Volume 5, 2001.

17. Lo Chi-Chun and Chang Wei-Hsin, “A multiobjective hybrid genetic algorithm for the capacitated multipoint network design problem”, IEEE Transactions on Systems, Man. And Cybernetics, Part B: Cybernetics, 30(3): 461-469, 2000.

18. Yi-Kuei Lin, “Two-commodity reliability evaluation for a stochastic-flow network with node failure”, Computers & Operations Research 29: 1927–1939, 2002.

DDoS Protections for SMTP Servers

Michael Still mikal@stillhq.com
School of Computer Science
The Australian National University
ACT 0200 Australia

Eric C. McCreath ericm@cs.anu.edu.au
School of Computer Science
The Australian National University
ACT 0200 Australia

Abstract

Many businesses rely on email of some form for their day to day operation. This is especially true for product support organizations, who are largely unable to perform their role in the company if their in boxes are flooded with malicious email, or if important email is delayed because of the processing of attack traffic. Simple Message Transfer Protocol (SMTP) is the Internet protocol for the transmission of these emails. Denial of Service (DoS) attacks are deliberate attempts by an attacker to disrupt the normal operation of a service with the goal of stopping legitimate requests for the service from being processed. This disruption normally takes the form of large delays in responding to requests, dropped requests, and other service interruptions.

In this paper we explore the current state of research into Distributed Denial of Service (DDoS) attack detection, protection and mitigation for SMTP servers connected to the Internet. We find that whilst there has been significant research into DDoS protection and detection generally, much of it is not relevant to SMTP servers. During our survey we found only two papers directly addressing defending SMTP servers against such attacks.

Keywords: Distributed Denial of Service, email, Simple Mail Transfer Protocol, Survey Paper.

1. INTRODUCTION

Allman [4] states that spam costs US businesses $87 billion a year. It seems reasonable to assume that if a low level attack is costing that much, then a complete outage would impose an even greater burden on an enterprise. Interestingly, despite the importance of SMTP to modern business operations, little research appears to have been applied to how to protect SMTP from deliberate attack, apart from whatever protection may be derived from generic defenses.

SMTP is a unique protocol in terms of its needs of DDoS protection. This is largely because of the need to sync queued email to disk, so as to not lose email in the queue in the case of a system failure. In fact, SMTP is an unusually easy protocol to DDoS [8], requiring relatively small amounts of bandwidth to render inoperable. Also SMTP is of increasing importance to modern business operations, yet, approaches focused on DDoS protection for SMTP have not gained much attention. These factors make DDoS protection for SMTP an area of research interest and significance.

Denial of service attacks may be grouped into two main categories:

1. Attacks that exploit flaws in the implementation of the server system, normally in the form of misconfigurations [34, 23, 22, 31, 33, 36]. For example SYN flooding works on the assumption that the server's TCP implementation allocates memory for the TCP connection at the time that the SYN packet is received. The attacker therefore sends many SYN packets, but does not ACK connection establishment when the server offers it. The server therefore has this memory allocated until the TCP connection times out [12]. Modern operating systems either limit the number of connections per source, or use techniques such as SYN cookies to avoid allocating memory at the time of the SYN packet. These vulnerabilites may exist at the application layer as well as the operating system layer. For example, if you can send a request that causes the application server to crash, then you have denied access to that server until it can be restarted, either manually or automatically. Another example is a request that takes a disproportionally long time to respond to – for example, early versions of Microsoft's IIS web server would take extremely long times to parse certain malformed URLs [30].
2. An attack is simply a distributed attempt to consume all of a scarse resource [31, 33, 23, 36] such as CPU, network IO or disk IO. These attacks are termed Distributed Denial of Service (DDoS) attacks [22] as the flood traffic comes from many machines, and is not a single flow on the network [27]. When an attack targets a host's upstream network bandwidth specifically, then it is often termed a "bandwidth attack" [23, 36]. These flooding attacks are often not detected by traditional signature detection schemes [25], and are harder to defend against with simple address based filtering. Often these attacks use clients which forge their sender address, such forging if used is known as IP spoofing. These clients are known as zombies [19, 33], and are often poorly secured home machines on broadband connections [46]. A group of zombies under the control of a single hacker (or group of hackers) is known as a botnet. This flooding behavior is exacerbated by these attacking zombies ignoring TCP flow control mechanisms, whereas legitimate clients will reduce the size of their traffic flows – thus increasing the proportion of traffic which is malicious [36]. Worse, these attacks do not imply a mis-configuration on the part of the site administrator, and are much harder to defend against. The implementation of bandwidth attacks is based on the volume of requests, not the content of the requests [36].

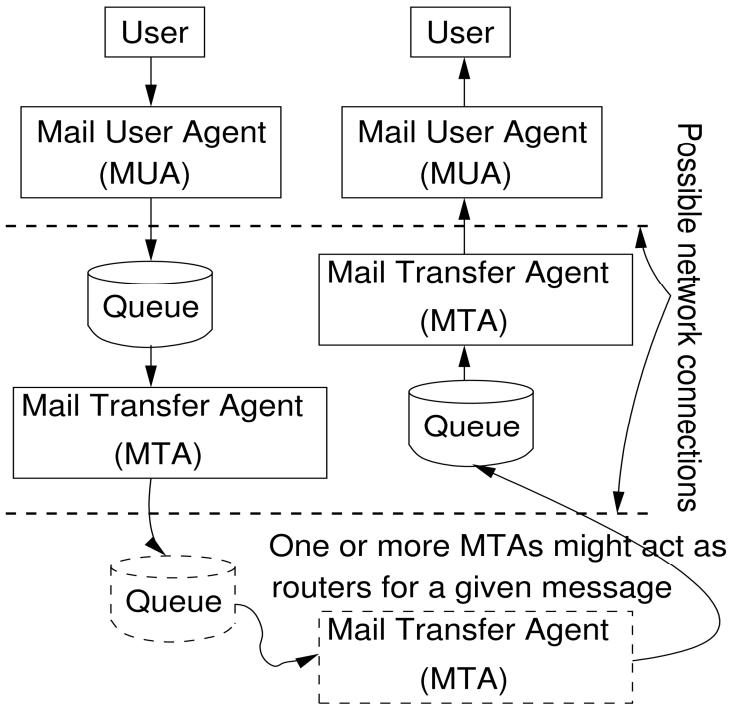


FIGURE 1: A common SMTP path

Both of these categories apply to SMTP servers. For example, a recent instance of a SMTP specific vulnerability in the implementation of server software is [14], which details a MX record parsing vulnerability in some Microsoft SMTP server implementations. However, this review focuses on the second form of attack, where a large number of requests are being made at any one time. This is because SMTP servers are unusually vulnerable to these distributed attacks because of the relatively low amount of bandwidth [8] required to saturate the available disk bandwidth of current servers.

Interestingly, the requests sent as part of DDoS are not necessarily malicious, they are just timed in such a way as to cause disruption to normal levels of service. For example, a large number of incoming emails following some sort of catastrophic event such as the 9/11 attacks on New York's World Trade Center can be characterized as a DDoS despite the intent not being malicious. The event causing the flood of emails also does not need to be catestrophic – large email newletter campaigns have also been known to cause SMTP servers to stop responding to requests in a reasonable amount of time [8].

2. SMTP ROUTING PRIMER

It is important to briefly introduce how an email is typically routed by SMTP servers, as this is important background to the DDoS protections discussed later in this review. This section is only a brief summary however, and reference to the relevant RFCs as well as Stevens' description in [39] are recommended for more detail. A common SMTP path is shown in Figure 1. In this example, the user creates an email using a Mail User Agent (MUA), and when they select the send action the mail is delivered to a Mail Transport Agent (MTA) on either the same machine as the MUA, or another machine.

This MTA routes the email to the destination Mail eXchanger (MX), possibly via a number of other MTAs depending on local configuration. The MX is simply another MTA, but it is listed in the Domain Name System (DNS) as being capable of delivering email for the destination domain name. At the MX, the email is sent through another variable length chain of MTAs until it reaches the MTA that can deliver mail to the recipient's mailbox on disk. This final MTA then uses a Mail

Delivery Agent (MDA) which may be built into the MTA or be a separate program such as procmail to actually write the message to disk. The mailbox is then checked periodically by another MUA, which displays the mail to the recipient. It is possible that there is a network connection between the mailbox and the destination MUA, or they might be on the same machine.

There are a few more aspects of this design that deserve more attention:

- This is an unusually complicated path. Most email will flow from a MUA to a local MTA, via one routing MTA (called a smart host) to the destination MX's MTA, and then to the recipient MUA. There is little research to support this assertion however.
- Every MTA along the delivery path is required to reliably add the email to its queue, as once the email is accepted, it is deleted from the sender's queue. This incurs costly disk syncs to ensure the data is queued reliability.
- It is possible to insert additional MTAs in the delivery path, which act much like proxies. This is commonly done to implement functionality such as virus and spam scanning. These checks can be extremely expensive to execute, this slows this MTA down futher.
- There are very few guarantees for how quickly an email will be delivered. This will depend on the number of MTAs in the mail's path, how busy they are, and how long the mail stays in each queue before being processed.

A successful SMTP DDoS needs only to cause congestion on the last provisioned portion of this path to cause an outage for the end user.

3. PRIOR DDoS RESEARCH

There has been extensive research into DDoS attacks. This section discusses this research in the context of SMTP servers specifically. We discuss: how common DDoS attacks are; existing methods for detecting attacks; and finally existing attack defenses. Unfortunately, not much of this research has examined SMTP specifically. The existing research specifically addressing SMTP servers that we could find was [8, 9]. We therefore comment on the specific implications of existing research on SMTP as appropriate.

3.1 How Common are DDoS Attacks?

CERT data indicates that security attacks overall are becoming much more common – so common in fact that CERT no longer reports individual incidents [36]. There is also existing research into the prevalence of DDoS attacks, which finds that the volume of attack traffic arriving at networks is significant. For example, Pang et al. [35] find that the Lawrence Berkeley National Laboratory experienced 8 million connection attempts to unused addresses in just one day. This was two thirds of the traffic received on that day. Moore et al. [33] found that the rate of attacks is relatively constant, although it has nearly tripled in the three years of sampling the paper covers. Clearly, scanning and attempted DDoS attacks are common on the modern Internet.

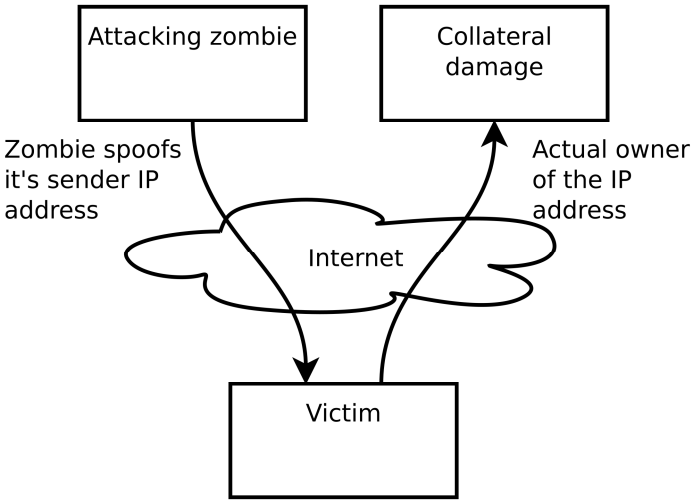


FIGURE 2: Conceptual overview of backscatter analysis

It should be noted that this research is largely dependent on either packet capture of some form[1] or backscatter analysis [33, 35].

Backscatter is essentially a form of network level "collateral damage". Figure 2 shows that when a zombie sends an attack packet with a forged sender IP address, it runs the risk of either using a real machine's IP address as the source address, or an address from an unallocated block of network addresses. When the server under attack attempts to reply to the attack packet, it will instead reply to the real owner of the address. This real owner can detect that an unsolicited packet has arrived because of the state of the IP connection at the time the packet arrives.

Backscatter analysis is the process of using these unsolicited packets to infer information about DDoS attacks occurring on the larger Internet. More discussion of this technique may be found in [33], although it should be noted that not all unsolicited traffic is an indication of a spoofed attack packet [35], for example network scanning and broadcast packets with all result in unsolicited traffic. The utility of this form of analysis is reduced by the decrease in the number of attacks using spoofing [28].

There are now several teams of researchers using large unallocated IP address blocks to analyze the backscatter from DDoS attacks which use address spoofing for example [36]. They refer to their packet capture setups as "network telescopes" because of the manner in which they amplify the signal from distant events.

These techniques however do not provide estimates for how commonly SMTP servers are attacked. SMTP servers are advertised in MX records for domain names, and are unlikely to be attacked based on simple scans for IP address space. In fact, random IP addresses are quite unlikely to run into a SMTP server, as shown by our recent surveys of SMTP servers on the Internet. Additionally, because SMTP traffic is not vulnerable to spoofing, backscatter analysis provides no assistance. We are unable to find any reference in existing research to the prevalence of SMTP DDoS attacks.

3.2 Attack Detection Methods

1 And therefore the requirement that the researchers have access to a network which either was attacked, was attacking, or provided transit to an attack.

It is important to successful DDoS mitigation that attack detection is both quick, and unlikely to incorrectly identify non-malicious traffic as an attack2 [36]. The system also needs to be able to implement an effective response that favours legitimate traffic once an attack is detected [32].

There are three main forms of DDoS detection discussed in the literature [31, 22, 25]:

1. Pattern detection – these techniques seek to find patterns in requests, and then determine if those patterns are associated with legitimate requests. Often these systems have predefined lists of signatures which indicate a common attack. These specific behaviours (such as executing a port scan) are considered indicative of malicious intent. This technique is widely deployed in the form of many Intrusion Detection Systems (IDS) such as snort [38]. Such schemes are possibly better implemented at a higher level of the network stack, where more information about the connection between the client and server is known (such as the user who is currently connected) [47].
2. Anomaly detection – a base line for "normal" traffic is generated and then used to identify possible attacks. These anomalies may be in the form of unusual traffic flows (for example a large amount of traffic to a machine which generally receives little), or a behavior (for example a failure to respect TCP flow control mechanisms for a TCP flow) [36]. This is hard to do on real networks, as traffic flows can be highly variable, whilst not being malicious. However, this approach holds the most promise for SMTP as anomalies would present themselves as unusual traffic flows, either in a larger than normal number of emails being delivered to one recipient, or a larger number of emails than usual coming from a limited number of clients [5]. Further research into this option is desirable. The baseline data for these anomaly detection systems is often at the packet level. There has been some discussion that moving these systems to the TCP layer would provide a more holistic view of flows and therefore improve the accuracy of attack detection [47]. Further moving this anomaly detection to the application layer would provide further benefits – such as knowledge of the specific users which are creating flows.
3. Third party detection – these are systems which do not perform any attack detection themselves, but act on instructions from an external source. This might be in the form of a commercial service, or a network wide traceback mechanism such as CenterTrack [5, 40].

3.3 Evaluation of Attack Detection Methods

Extensive research has been conducted into generic DDoS attack detection. However, there have been limited research into how to make these detection schemes scale well. One example of such research is [25], which investigates aggregation techniques as a method of improving performance. Many of these existing techniques, such as port scan detection, are currently implemented in the form of large vectors which do not scale to high data rates [25]. Aggregation of flows is an option for improving performance, but this can result in "behavioral aliasing" where either an aggregate falsely identifies non-malicious traffic as malicious, or an aggregate which fails to identify malicious traffic because it is masked by otherwise unrelated non-malicious traffic in the flow [25].

There is promise for these detection techniques for detecting attacks against SMTP servers, although there is currently little research into how to perform this detection. The only directly relevant research the authors have found during their review is [7]. Here, a very naive anomaly detection algorithm is used, with attack protection being triggered by overall processing queue length hitting a defined threshold (either queue overflow, or queue length meeting defined parameters). Once protection is triggered, attack traffic is identified by looking for network adresses with higher means than normal. This method is vulnerable to "traffic laundering" through constructs such as botnets, as individual network addresses can still be responsible for very small amounts of traffic, and the widely distributed. Traffic from identified sources is then discarded.

Whilst this implementation shows promise, it suffers from naive triffering and simplistic behaviour once triggered. We believe that attempting to cluster traffic using a variety of attributes would be a more accurate triggering mechanism, and possibly would be able to be used permanently, instead of only when in "attack mode". Future research into this area would be promising.

3.4 Attack Defenses

In this section we enumerate the various DDoS attack defenses discussed in the literature, and provide an evaluation of their effectiveness in the case of SMTP servers. Most of the existing evaluations assume that a solution to DDoS attacks should either be implemented at the source of the attack [44, 46], or be built into TCP/IP itself [31, 5, 25, 36]. Also some proactive approaches are possible [45].

DDoS attack defenses overall may be grouped into four categories:

1. Over provisioning – provide enough server capacity to handle the system peak load, plus a concurrent DDoS attack. This is a common technique, despite difficulty in predicting the largest DDoS attack which might occur. An example of this technique is Content Delivery Networks (CDN)s covered in Section 3.8. This is probably the most common approach employed by likely targets of SMTP DDoS, with many such organizations deploying large clusters of mail servers.
2. Routing controls – have attack traffic not routed to the server under attack. Examples include: some forms of overlay networks (Section 3.8); push back mechanisms (Section 3.7); various changes to core Internet protocols such as reworking how network addresses are allocated (Section 3.6). This approach holds significant promise for SMTP servers, and has been initially investigated by Bencsáth [9].
3. Currency proposals – DDoS attacks are premised on the assumption that clients are cheap and that servers are expensive. If this is made no longer true by making client connections more expensive, then many attackers will no longer be able to afford significant traffic levels. Currency does not have to be monetary – another commonly cited proposal is to use proof of expenditure of computational resources as currency – for example the computation of hashes. These are covered more in Section 3.5.
4. Authentication systems – such as whitelists[2]; blacklists[3]; and CAPTCHAs[4] .

3.5 Currency Proposals

One class of proposals to stop both spammers and DDoS attacks is to change the economic model used by the attackers. Both spammers and zombies operate on the assumption that clients are cheap, and that many may be used at once. There are currency proposals which aim to change this. Currency proposals include:

- System resources – teergruben-like systems [16] extend the length of possible spam SMTP connections dramatically, in an attempt to have spammers use capacity in their TCP stacks as payment for having sent the spam. Additionally, general tar pitting systems[5] are useful for rate limiting some forms of abusive sender [21].
- Expended effort – for example, proof that the sender has consumed a certain minimum number of CPU cycles in order to allow the delivery of this one email. Examples include Microsoft's Penny Black project [18, 17, 1, 2] and Hashcash[6]. Generally these schemes use the computation of hashes as proof of resource consumption.
- Money – finally, these are escrow proposals where actual money is held by a third party on the promise that the request from the client is not malicious. The escrow payment is released if the recipient agrees.

These proposals offer an interesting solution to DDoS attacks, as they make it more expensive to attempt to flood a server with traffic. However, these proposals suffer from the same practical limitations as ingress and egress filtering – to be effective they require a large scale deployment,

2 A list of users or servers always allowed to connect.
3 A list of users or servers never allowed to connect.
4 A simple character recognition puzzle used to separate machines from humans.
5 Algorithms which increasingly slow connections from systems which are deemed to be using more than their fair share of a finite resource such as server capacity.

which is difficult to achieve on the Internet. Additionally, computational time on zombie machines is effectively free, so expending resources is not a large burden in this case.

3.6 Address Allocation Changes

Handley and Greenhalgh [20], propose breaking the IP address space into "server" and "client" addresses. Clients would then be able to only initiate connections to servers, which would respond. Servers would not be allowed to initiate their own connections to clients, and clients would not be allowed to connect to other clients. They argue that this will stop zombies from receiving commands over the network. The authors also argue in favor of changes to the IP protocol to make it clearer when a session is still being setup.

The assertion that breaking the address space into client and server addresses would stop zombies from receiving commands ignores the possibility of the zombies polling a server for commands, which is common already because of the widespread use of Network Address Translation (NAT). This proposal also ignores the breakage of peer to peer applications that this proposal causes, although the authors address this by the suggestion of either providing two addresses to some machines, or building a network of proxies to forward on the connections from these client machines, which undermines the separation concept. Additionally in the SMTP case it is common for "client IPs" to contact servers. For example, roaming laptops and mobile phones often end email via remote authenticating SMTP servers to simplify configuraton.

3.7 Push Back Mechanisms

Push back is a mechanism in which routers upstream of the server under attack[6] are asked to start dropping packets to the server under attack [26, 27]. They address the failure of attacking zombie machines to respond correctly to TCP flow control mechanisms [36], as IP assumes that a client will respond to such requests with a reduced traffic level [26]. Lakshminarayanan et al. [26] proposes that push back be implemented by allowing hosts to add filtering rules to the router on the ISP's side of the network link offering transit to the attack, based on the assumption that the ISP is better provisioned to handle the level of traffic caused by the attack than the transit link is. This assumes that it is the network link to the server that is the resource being saturated during the attack. A proposed implementation is as shown in Figure 3.

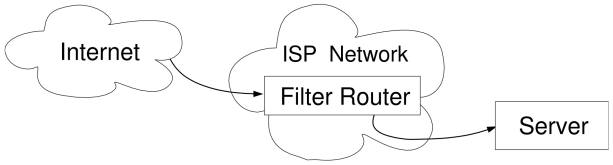


Figure 3: A remote traffic filtering implementation

In this case the server would push updates to the filtering rules to the router to protect the link between the server and the ISP's router from saturation. If the ISP was unwilling to allow customers to update filters on their routers, then the client could host their own proxy at the ISP, which implements the filtering required before sending legitimate traffic through the router to the server. The server under attack can then update the filter rules on the proxy without affecting the ISP's routing configuration.

Accordng to Bencsath's early research into their utility with SMTP [9], push back mechanisms show promise because they offer a means of controlling aggregate flows which are the result of

6 These routers are not necessarily well placed in the Internet's topology. The criteria is simply that they are between the attacking machines and the server under attack, and are better connected to the Internet than the server under attack. This is attractive because it is much more likely that these routers are under the same control as the server under attack, which means it is more likely that filtering rules can be implemented quickly compared with routers closer to the origin of the attack.

combining many low bandwidth flows into one larger flow. These smaller flows might all individually respect flow control mechanisms, but when combined still cause an overload condition [27]. Such mechanisms also offer assistance in the handling of unexpected traffic from "flash crowds" (a large unexpected burst of otherwise legitimate user traffic - sometimes referred to as the "Slashdot effect") [27]. These push back mechanisms are not a perfect solution to DDoS attacks, and in some cases can in fact make the situation worse [27].

3.8 Overlay Networks

There are DDoS attack protections that do not require the modification of core Internet protocols, forexample proposals that harness overlay networks to provide protection. Lakshminarayanan et al. [26] argue that the ability of a host to control the traffic sent to it is fundamental to the solution to the DDoS problem, especially as it is the end hosts who know the most about the traffic flows they are receiving. Whilst push back mechanisms go some way to offering control of the traffic sent to a server, Lakshminarayanan et al. are representative of the group of researchers who argue that further control is needed. Therefore, there are several proposals for proxy services which make the servers that provide appli cations which might be attacked anonymous. The idea is that you cannot attack something which you cannot find, and that the proxy network is so over-provisioned that it isn't vulnerable to realistic DDoS attacks. These proxy networks are a special case of an "overlay network".

More generally an overlay network is a network constructed on top of another network. There are a number of proposals[24, 26] which utilize an overlay network based on Internet Indirection Infrastructure (*i3*). In the *i3* network, a host registers an identifier, and packets requiring that host identifier are sent to *i3. i3* then looks up the identifier, and forwards the traffic onto the host. It is argued that because servers are not widely known to the Internet, they do not expose other ports than those required to implement the application to attack. Further, a server can stop traffic flow by simply unregistering its identifier (although this will affect legitimate users of the service as well). An i3 based proxy system also allows for the implementation of "next generation" IP services such as mobile clients, multicast and anycast.

As mentioned earlier, overlay networks are not always proxy based. An example of a non-proxy overlay network is VIPnet, proposed by Brustoloni in [11]. VIPnet implements a DDoS mitigation system by offering preferred routing to important users of a server in return for payments to the transit ISPs for the VIP user. Clearly this is not a generally applicable solution. Another non-proxy overlay network is CenterTrack [40], which uses an overlay equipped with IP traceback capable routers to perform network traceback on networks otherwise not capable of performing such analysis [36].

The goal of DDoS protection mechanisms is to minimize the harm to genuine users of an attack on the service. Two common ways to providing this harm minimization is to either stop the malicious requests from consuming resources on the server, or massively over provision the system [31] so that these malicious requests do not affect the requests of genuine users. For relatively static content which needs to be served globally, a common technique is to implement a Content Distribution Network (CDN), also known as Content Delivery Networks. CDNs are used to increase throughput for popular or vulnerable sites [37, 43, 10, 42, 41]. The most well known of these CDNs is run by a company called Akamai, and is composed of over 20,000 servers operating in 71 countries and 1,000 networks [41, 3]. Akamai deploys these servers onto ISP networks at no charge to the ISP. This is attractive to ISPs as it reduces their bandwidth expenses. CDNs are constructed from a set of geographically distributed proxies (also known as surrogates), which return results instead of the sites main servers. CDNs have often been compared to peer to peer (P2P) download networks [15, 37]. One notable difference is that CDNs are centrally controlled and managed, whereas P2P networks are not. CDNs have a number of advantages:

1. It moves the content closer to the user, thus reducing latency when fulfilling requests. This is because the TCP three way handshake happens over a much lower latency network path, thus improving TCP session setup speed.

2. It reduces peering and transit costs for ISPs by allowing them to reduce the number of times the same object must traverse their peering links, because the content is hosted at the ISP it need only be transferred over the ISP's peering links once.
3. It limits the region which is affected by a DoS attack. For example, if an attacker is on a network with a CDN proxy for the site they are attacking, then their requests will be responded to by that local proxy. Therefore, the only effect of the DoS attack is to reduce the speed of the site for other users of that local network, not all users of the site around the world.

CDNs are only useful for sites where the data to be added to the CDN is read only and where personalization mechanisms such as cookies are not allowed to reduce the cache-ability of objects from the site. This is especially true for sites which require cookies for all requests, even those where none is needed [10]. We are unaware of any CDN provider which currently supports distributing SMTP servers.

4. THE CURRENT STATE OF SMTP

What is the current state of SMTP servers on the Internet? These servers face several challenges, including consistent low level attacks from spammers, as well as email borne viruses and worms. DDoS protections for SMTP servers can be informed by previous work on these problems.

Unsolicited Commercial Email, also known as spam, may be characterized as illegitimate requests coming from many machines[7], however the request rate is low enough that it does not cause server outages and therefore cannot be characterized as a DDoS attack. Current estimates of spam rates indicate that up to 74.5% of emails sent are spam[29]. Whilst this is a significant percentage of the current SMTP traffic levels on the Internet, it has now been sustained for so long that it is considered part of the status quo and SMTP servers connected to the Internet are configured to handle the current spam workload.

Current spam detection techniques can be broken into two broad categories: content based techniques; and sender behavior based techniques. The content based techniques commonly used are [46]:

1. Email address filters – also known as origin-based filters [13]. These are simply lists of email senders or emails servers who are known spammers (a blacklist), known non-spammers (a whitelist) and possibly suspicious senders (a greylist).
2. Heuristic filters, including machine learning approaches, based on known spam features – for example words such as "viagra" [13].

Wong et al. [44] determine that outgoing email worms can be detected from the pattern of DNS requests that they make when sending their email. They propose implementing a mail worm watchdog on DNS servers to alert when worm email is being sent.

5. CONCLUSION

We have brought together some of the research on DDoS from the perspective of SMTP. This provides a useful starting point for research in this area. For there is significant scope and value of future research into: the state of SMTP transactions on the Internet, the vulnerability of SMTP servers to DDoS attacks, and the creation of defense approaches.

We believe that a viable approach to SMTP server DDoS protection is to deploy push back routers as intermediaries between the senders of email and the receiving server, as described in [7]. These servers could be deployed much like a Content Delivery Network, and therefore provide protection for more than one SMTP server at any given time. However, the push back routers should archive email which is categorized as having a high probability of being an attack,

7 Some of these machines in fact being zombies.

and this email should be processed by the recipient servers during non-peak periods where further analysis of the traffic is possible. Further work is also required on attack traffic detection.

6. REFERENCES

[1] M. Abadi, A. Birrell, M. Burrows, F. Dabek, and T. Wobber. Bankable Postage for Network Services. In Proceedings of the 8th Asian Computing Science Conference. Springer-Verlag, 2003.

[2] M. Abadi, M. Burrows, M. Manasse, and T. Wobber. Moderately hard, memory-bound functions. ACM Transactions on Internet Technology (TOIT), 5(2):299–327, 2005.

[3] Akamai. Technology overview, 2007. Available from http://www.akamai.com/html/technology/index.htm, accessed on 5 July 2007.

[4] Eric Allman. Spam, Spam, Spam, Spam, Spam, the FTC, and Spam. Queue, 1(6):62–69, 2003.

[5] Tom Anderson, Timothy Roscoe, and David Wetherall. Preventing Internet denial-of-service with capabilities. SIGCOMM Comput. Commun. Rev., 34(1):39–44, 2004.

[6] Adam Back. Hashcash - A Denial of Service Counter-Measuree, 2002. Available from http://www.hashcash.org/papers/hashcash.pdf, accessed on 7 July 2007.

[7] Boldizsár Bencsáth. New Approaches to Mitigate Network Denial-of-Service Problems. PhD thesis, BME Informatikai Tudományok doktori iskola, 2009.

[8] Boldizsár Bencsáth and Miklós Aurél Rónai. Empirical analysis of denial of service attack against smtp servers. In Proceedings of The 2007 International Symposium on Collaborative Technologies and Systems, pages 72–79. IEEE, 2007.

[9] Boldizsár Bencsáth and István Vajda. Protection against ddos attacks based on traffic level measurements. In 2004 International Symposium on Collaborative Technologies and Systems, pages 22–28., San Diego, CA, USA, January 2004.

[10] L. Bent, M. Rabinovich, G. M. Voelker, and Z. Xiao. Characterization of a large web site population with implications for content delivery. In WWW '04: Proceedings of the 13th international conference on World Wide Web, pages 522–533, New York, NY, USA, 2004.

[11] JosÃl' Brustoloni. Protecting electronic commerce from distributed denial-of-service attacks. In WWW '02: Proceedings of the 11th international conference on World Wide Web, pages 553–561,New York, NY, USA, 2002.

[12] CERT. CERT Advisory CA-1996-21 TCP SYN Flooding and IP Spoofing Attacks, 1996. Available from http://www.cert.org/advisories/CA-1996-21.html, accessed on 4 October 2007.

[13] Duncan Cook, Jacky Hartnett, Kevin Manderson, and Joel Scanlan. Catching spam before it arrives: domain specific dynamic blacklists. In ACSW Frontiers '06: Proceedings of the 2006 Australasian workshops on Grid computing and e-research, pages 193–202, Darlinghurst, Australia, Australia, 2006.

[14] Microsoft Corporation. Microsoft Security Bulletin MS10-024: Vulnerabilities in microsoft exchange and windows smtp service could allow denial of service (981832), April 2010.

[15] Shibsankar Das and Jussi Kangasharju. Evaluation of network impact of content distribution mechanisms. In InfoScale '06: Proceedings of the 1st international conference on Scalable information systems, page 35, New York, NY, USA, 2006.

[16] Lutz Donnerhacke. Teergurbing FAQ. Available from http://www.iks-jena.de/mitarb/lutz/usenet/teergrube.en.html, accessed on 12 November 2007.

[17] C. Dwork, A. Goldberg, and M. Naor. On memory-bound functions for fighting spam. Advances on Cryptology (CRYPTO 2003), Santa Barbara, CA, USA, August, 2003.

[18] C. Dwork and M. Naor. Pricing via Processing or Combatting Junk Mail. Proceedings of the 12th Annual International Cryptology Conference on Advances in Cryptology, pages 139–147, 1992.

[19] Hikmat Farhat. Protecting TCP services from denial of service attacks. In LSAD '06: Proceedings of the 2006 SIGCOMM workshop on Large-scale attack defense, pages 155–160, New York, NY, USA, 2006.

[20] Mark Handley and Adam Greenhalgh. Steps towards a DoS-resistant internet architecture. In FDNA'04: Proceedings of the ACM SIGCOMM workshop on Future directions in network architecture, pages 49–56, New York, NY, USA, 2004.

[21] Tim Hunter, Paul Terry, and Alan Judge. Distributed Tarpitting: Impeding Spam Across Multiple Servers. In LISA '03: Proceedings of the 17th USENIX conference on System administration, pages 223–236, Berkeley, CA, USA, 2003.

[22] Alefiya Hussain, John Heidemann, and Christos Papadopoulos. A framework for classifying denial of service attacks. In SIGCOMM '03: Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications, pages 99–110, New York, NY, USA, 2003.

[23] Frank Kargl, Joern Maier, and Michael Weber. Protecting web servers from distributed denial of service attacks. In WWW '01: Proceedings of the 10th international conference on World Wide Web, pages 514–524, New York, NY, USA, 2001.

[24] Angelos D. Keromytis, Vishal Misra, and Dan Rubenstein. SOS: secure overlay services. In SIGCOMM '02: Proceedings of the 2002 conference on Applications, technologies, architectures, and protocols for computer communications, pages 61–72, New York, NY, USA, 2002.

[25] Ramana Rao Kompella, Sumeet Singh, and George Varghese. On scalable attack detection in the network. In IMC '04: Proceedings of the 4th ACM SIGCOMM conference on Internet measurement, pages 187–200, New York, NY, USA, 2004.

[26] Karthik Lakshminarayanan, Daniel Adkins, Adrian Perrig, and Ion Stoica. Taming IP packet flooding attacks. SIGCOMM Comput. Commun. Rev., 34(1):45–50, 2004.

[27] Ratul Mahajan, Steven M. Bellovin, Sally Floyd, John Ioannidis, Vern Paxson, and Scott Shenker. Controlling high bandwidth aggregates in the network. SIGCOMM Comput. Commun. Rev., 32(3):62–73, 2002.

[28] Z. Morley Mao, Vyas Sekar, Oliver Spatscheck, Jacobus van der Merwe, and Rangarajan Vasudevan. Analyzing large DDoS attacks using multiple data sources. In LSAD '06: Proceedings of the 2006 SIGCOMM workshop on Large-scale attack defense, pages 161–168, New York, NY, USA, 2006.

[29] MessageLabs. MessageLabs Intelligence. Available from http://www.messagelabs.com/mlireport/MLI_Report_October_2007.pdf, accessed on 12 November 2007.

[30] Microsoft. Microsoft Security Bulletin (MS00-030): Frequently Asked Questions. Available from http://www.microsoft.com/technet/security/bulletin/fq00-030.mspx, accessed on 12/11/2007.

[31] Jelena Mirkovic and Peter Reiher. A taxonomy of DDoS attack and DDoS defense mechanisms. SIGCOMM Comput. Commun. Rev., 34(2):39–53, 2004.

[32] Jelena Mirkovic, Max Robinson, and Peter Reiher. Alliance formation for DDoS defense. In NSPW '03: Proceedings of the 2003 workshop on New security paradigms, pages 11–18, New York, NY, USA, 2003.

[33] David Moore, Colleen Shannon, Douglas J. Brown, Geoffrey M. Voelker, and Stefan Savage. Inferring Internet denial-of-service activity. ACM Trans. Comput. Syst., 24(2):115–139, 2006.

[34] Judith M. Myerson. Identifying enterprise network vulnerabilities. Int. J. Netw. Manag., 12(3):135–144, 2002.

[35] Ruoming Pang, Vinod Yegneswaran, Paul Barford, Vern Paxson, and Larry Peterson. Characteristics of internet background radiation. In IMC '04: Proceedings of the 4th ACM SIGCOMM conference on Internet measurement, pages 27–40, New York, NY, USA, 2004.

[36] T. Peng, C. Leckie, and K. Ramamohanarao. Survey of network-based defense mechanisms countering the DoS and DDoS problems. ACM Comput. Surv., 39(1):3, 2007.

[37] S. Saroiu, K. P. Gummadi, R. J. Dunn, S. D. Gribble, and H. M. Levy. An analysis of Internet content delivery systems. SIGOPS Oper. Syst. Rev., 36(SI):315–327, 2002.

[38] Snort Team. Website, 2007. Available from http://www.snort.org/, accessed on 1/12/2007.

[39] W. Richard Stevens. The Protocols (TCP/IP Illustrated, Volume 1). Addison-Wesley Professional, 1993.

[40] R. Stone. Centertrack: an IP overlay network for tracking dos floods. In Proc of the 9th conf. on USENIX Security Symposium - Volume 9, pages 15–15, Berkeley, CA, USA, 2000.

[41] Ao-Jan Su, David R. Choffnes, Aleksandar Kuzmanovic, and Fabián E. Bustamante. Drafting behind Akamai (travelocity-based detouring). In SIGCOMM '06: Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications, pages 435–446, New York, NY, USA, 2006.

[42] Chitra Venkatramani, Olivier Verscheure, Pascal Frossard, and Kang-Won Lee. Optimal proxy management for multimedia streaming in content distribution networks. In NOSSDAV '02: Proceedings of the 12th international workshop on Network and operating systems support for digital audio and video, pages 147–154, New York, NY, USA, 2002.

[43] Limin Wang, Vivek Pai, and Larry Peterson. The effectiveness of request redirection on CDN robustness. SIGOPS Oper. Syst. Rev., 36(SI):345–360, 2002.

[44] Cynthia Wong, Stan Bielski, Jonathan M. McCune, and Chenxi Wang. A study of mass-mailing worms. In WORM '04: Proceedings of the 2004 ACM workshop on Rapid malcode, pages 1–10, New York, NY, USA, 2004.

[45] Y. Xiang and W. Zhou. An Active Distributed Defense System to Protect Web Applications from DDoS Attacks. In The Sixth International Conference on Information Integration and Web Based Application & Services, 2004.

[46] Mengjun Xie, Heng Yin, and Haining Wang. An effective defense against email spam laundering. In CCS '06: Proceedings of the 13th ACM conference on Computer and communications security, pages 179–190, New York, NY, USA, 2006.

[47] Ying Xu and Roch GuÃl'rin. On the robustness of router-based denial-of-service (DoS) defense systems. SIGCOMM Comput. Commun. Rev., 35(3):47–60, 2005.

Implementation of New Routing Protocol for Node Security in a Mobile Ad Hoc Network

Virendra Singh Kushwah kushwah.virendra248@gmail.com
Department of Computer Science
Hindustan Institute of Management and Computer Studies,
Farah, Mathura, INDIA

Gaurav Sharma gauravsharma53@gmail.com
Department of Computer Science
GLA University,
Mathura, INDIA

Abstract

A routing protocol plays important role to handle entire network for communication and determines the paths of packets. A node is a part of the defined network for transferring information in form of packets. If all packets transferred from source to destination successfully, it has been assumed that the routing protocol is good. But, an attacker turns this dealing as a speed breaker and turning point of a highway. So, prevention from attacks and secure packets, a new routing protocol is being introduced in this paper. The proposed routing protocol is called by SNAODV (Secure Node AODV). This paper is also tried to maximize throughput as compared with AODV and SAODV.

Keywords: AODV, Routing, Secure, Packets, Network.

1. INTRODUCTION

A mobile ad hoc network (MANET) consists of a group of devices (or nodes) that rely on the wireless communication medium and themselves for data transmission. A node in an ad hoc network has direct connection with a set of nodes, called neighbouring nodes, which are in its communication range. The number of nodes in the network is not necessarily fixed. A MANET does not have base stations or routers. Each node acts as a router and is responsible for dynamically discovering other nodes it can directly communicate with. However, when a message without encryption is sent out through a general tunnel, it may be maliciously attacked. Nodes cooperate by forwarding packets on behalf of each other when destinations are out of their direct wireless transmission range. A centralized administrator and/or a pre-deployed network infrastructure are not necessary for a MANET to be set up, thus making its deployment quick and inexpensive.

In addition, Nodes ability to move freely ensures a flexible and versatile dynamic network topology which can be desirable in many situations. Hence, in addition to acting as hosts, each mobile node does the functioning of routing and relaying messages for other mobile nodes. Being mobile, any node can communicate to other nodes. Nodes do not necessarily know each other and come together to form an ad hoc group for some specific purpose. While limited bandwidth, memory, processing capabilities and open medium make its disadvantages. There are two types of possible attacks on nodes in MANET: passive attacks and active attacks. In passive attacks, adversaries simply drop and refuse to forward other nodes requests of assigning keys. In active attacks, in contrast, adversaries may return a fake reply (e.g. an invalid partial key) to the node requesting key. However, the security of MANET is still a challenge issue.

2. PROBLEM STATEMENT

There are a number of solutions for securing routing protocols in MANETs. We know there are two authentication models for securing routing is available that are ARAN [14] and SAODV [15] since they are closely related to our proposed model. In general, the existing schemas/models for secure routing are based on the assumptions of the availability of key management infrastructures which are unrealistic and contrast to the ad hoc network concepts. Moreover, these schemas do not consider intermediate nodes during the routing steps; therefore, the nodes may perform fabrication attacks. From these weaknesses of current approaches, our goal is to design a schema which performs point-to-point message authentication without a deployed key management infrastructure.

When two nodes are communicating, there may be any chance to steal packets, destroy packets or corrupt packets by malicious nodes. There are following two questions:-

1. Are nodes making right communication?
2. Are packets being saved during transmissions?

If these two questions are solved, at least it is understandable to prevent from misbehaviour nodes which make interfered between two or more right nodes during transmission of packets. So prevention is better than cure. To detect malicious nodes and remove those nodes is two way process [2]. So follow two processes, it is better process to use certificate on those nodes. If those nodes are secured, at least packets can be saved from attackers during transmission.

3. LITERATURES REVIEW

Security has become wide research area in MANETs. Most existing papers on deploying key management in MANETs usually mention flooding briefly as a way to distribute key in an ad hoc network using AODV routing protocol. Most secure communication protocols rely on a secure, robust and efficient key management scheme. Key management is also a central aspect for security in mobile ad hoc networks. In mobile ad hoc networks, the computational load and complexity for key management are strongly subject to restriction by the node's available resources and the dynamic nature of network topology.

1. A secure identity based key management scheme is proposed suitable for applying in MANETs. Similar to other ID-based cryptosystems, a trusted key generation center is needed in this scheme for verifying user identity and generating the corresponding private keys. [4]

2. Research work in key management scheme and handlings about limited number of nodes are possible in an ad hoc network. When the number of nodes increases, most of them become either inefficient or insecure. The main problem of any public key based security system is to make user's public key available to others in such a way that is authencity is verifiable. [5]

3. Using novel hierarchical security scheme, called Autonomous Key Management (AKM), which can achieve flexibility and adaptivity, and handles MANET with a large number of nodes. AKM is based on the hierarchical structure and secret sharing to distribute cryptographic keys and provide certification services. AKM also enables the ability to issue certificates with different levels of assurance with the help of a relatively small number of nodes. [6]

4. SEKM (Secure and Efficient Key Management) builds a public key infrastructure (PKI) by applying a secret sharing scheme and using an underlying multicast server groups. In SEKM, each server group creates a view of the certificate authority (CA) and provides certificate update service for all nodes, including the servers themselves. The advantage is that in SEKM it is easier for a node to request service from a well maintained group rather than from multiple "independent" service providers which may be spread in a whole area. [7]

5. In Authenticated Acknowledgement Scheme (AAS) to detect such selfish nodes, routes containing such nodes will be eliminated from consideration. The source node will be able to choose an appropriate route to send its data. The AAS scheme is a network-layer technique to detect the selfish nodes and to mitigate their effects. It can be implemented as an add-on to existing routing protocols for MANETs, such as DSR. The AAS scheme detects misbehavior through the use of a new type of authenticated acknowledgment scheme termed AAS, which assigns a fixed route of two hops (three nodes) in the opposite direction of the data traffic route. When a node wishes to communicate with another node, a methodology is performed by the sending and receiving nodes, which ensures authentication and integrity. [8]

6. In [14], the authors categorized three kinds of threats which are modification, impersonation and fabrication in AODV and DSR. On the basic of this analysis, the authors proposed a protocol called ARAN (Authenticated Routing for Ad hoc Networks) using cryptographic certificates to bring authentication, message-integrity and non-repudiation to the route discovery process based on the assumption of existing of a trusted certificate server. It is not appropriate with ad hoc networks because it forms a centralized element. Moreover, in this protocol, because the source node cannot authenticate intermediate nodes in the routing path, intermediate malicious nodes can use error message attacks to networks.

7. In [15], the authors extend the AODV routing protocol to guarantee security based on the approach of key management scheme in which each node must have certificated public keys of all nodes in the network. This work uses two mechanisms to secure the AODV messages: digital signature to authenticate the fixed fields of the messages and hash chains to secure the hop count field. This protocol uses public key distribution approach in the ad hoc network; therefore, it is difficult to deploy and computationally heavy since it requires both asymmetric cryptography and hash chains in exchanging messages. The protocol also did not consider the authentication of intermediate nodes; hence it could not prevent the attack of falsifying error messages in ad hoc networks.

4. SYSTEM MODEL

The principle of our model is that messages in ad hoc network must be authenticated to guarantee the integrity and non-repudiation so that the protocol and nodes can be prevented against several kinds of attacks. Each node in a network has its own a pair of public key *e* and private key *d* following RSA Public-key Crypto-system [13] by self-generation, and each node contains a list of neighbour nodes with records containing the information of a neighbour node including neighbour address, neighbour public key, and a shared secret key. This information is formed after the key agreement between two neighbour nodes to negotiate a pair of keys and a shared secret key. The details of our security schema for AODV are described as the following sections.

A. Key Agreement Process between Neighbor Nodes

A node joining a network requires sending key agreement messages to its neighbours to negotiate a shared secret key. The concept of this process is based on HELLO message in ad-hoc routing protocols. The node broadcasts a message indicating the negotiation request with neighbour nodes:

$$<KEY\_AGREEMENT\_REQ, request\_id, sender\_address, PK_S>$$

On receiving this request, nodes reply a message:

$$<KEY\_AGREEMENT\_REP, request\_id, sender\_address, neighbour\_address, PK_N>$$

(Where PK_S and PK_N are the public key of the sender node and replying node, respectively; *request_id* is a sequence number generated by the sender node) to indicate the receiving of the request message and inform that it is ready for the key agreement process. For each received message, the request node (i.e.; node A) creates a new

record in its neighbour list. Each record contains filled neighbour address and filled neighbour public key; the other fields of the record are empty. For each new record in the list, the request node (A) negotiates a secret key with the neighbour node (B) by the following steps:

1. Generate a key Ks by using a secure random number generator,
2. Encrypt Ks with PK_B (node B's public key) = encrypt PK_B (Ks),
3. Send an offer message
   <KEY_PASS, encrypt PK_B (Ks)> to B,
4. Wait ACK (acknowledgement) from B and check message integrity to finish the negotiation

When node B receives the key passing message, it decrypts "*encrypt PK_B (Ks)*" by its private key (pB) to get the shared key K. Then, node B sends the ACK message

<KEY_ PASS_ ACK, request_id, $HASH_{Ks}$ (request_id)>

to indicate successful shared secret key negotiation, where $HASH_{Ks}$ *(request_id)* is the hashed message of *request_id* by the shared key *Ks*.

Since RSA algorithm is used in the negotiation, the confidentiality of the shared key is guaranteed between the two nodes. The shared key is used for authenticating messages between two adjacent nodes later in AODV routing protocol. In the case a node does not have a shared key with its neighbour nodes; it cannot participate in routing transactions.

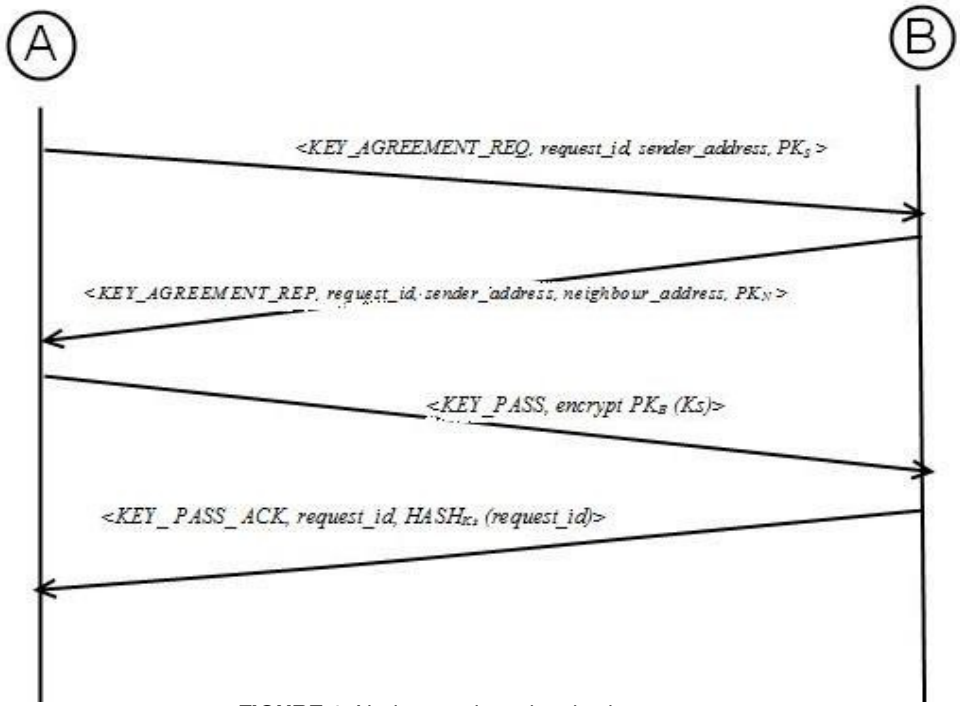


FIGURE 1: Node to node authentication process

B. Route Request

Route request (RREQ) is initiated by a source node (S) and then propagated by intermediate nodes until the message reaches its destination node (D). On receiving RREQ, an intermediate node I, according to our designed routing protocol, checks whether the message will be re-broadcasted or not. If the message needs to be re-broadcasted and the sender is in node I's neighbour list, it will send (unicast) a message to request the authentication process from the sender:

<RREQ_REQ, source_address, broadcast_id>.

When receiving the authentication request, the sender creates an authentication reply message containing

<RREQ_REP, source_address, broadcast_id, $HASH_{Ks}$ (RREQ)>

Where $HASH_{Ks}$ *(RREQ)* is the hashed value of RREQ message by the shared key Ks between the two nodes. The authentication reply message is unicasted back to node I. Node I on receiving the message will check the integrity of the RREQ message by hashing the message with using the shared key Ks and then comparing with the received hashed digest. If the comparison is successful (the integrity of the RREQ message is guaranteed), node I continues steps following AODV such as set up reverse path, increase the hop count, rebroadcast the message and so on; otherwise, the RREQ will be discarded. The process continues until the message reaches the destination. The destination also authenticates the sender of RREQ (neighbour of the destination) by the same procedure.

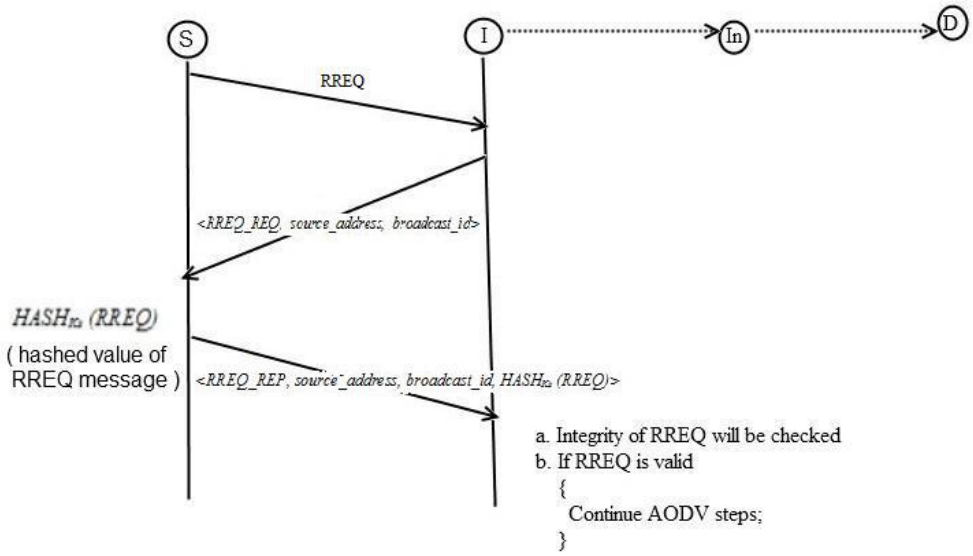


FIGURE 2: Representation of the message authentication

C. Route Reply and Route Maintenance

Route replies (RREP) in AODV are also targets for attacks by malicious nodes. In our model, when receiving a RREP, a node requests the sender to proof the integrity and non-repudiation of the message by sending an authentication message. The request for authentication is

<RREP_REQ, destination_address, destination_sequence#>

and the reply is

<RREP_REP, destination_address, destination_sequence#, $HASH_{Ks}$ (RREP)>

where $HASH_{Ks}$ *(RREP)* is the hashed value of RREP message by the shared key Ks between the two nodes. After the authentication process is successful, a node continues to the steps in AODV, otherwise, the node drops RREP since it is invalid.

In route maintenance process, only route error report message (RERR) is a target for attacks in AODV protocol. Our schema requires the authentication process in sending route error messages to prevent attacks from malicious nodes. The authentication request and response for RERR is

<RERR_REQ, unreachable_destination_address, unreachable_destination_sequence#>,

And

<RERR_REP, unreachable_destination_address, unreachable_destination_sequence#, $HASH_{Ks}$ (RERR)>,

respectively.

D. Routing Message formats

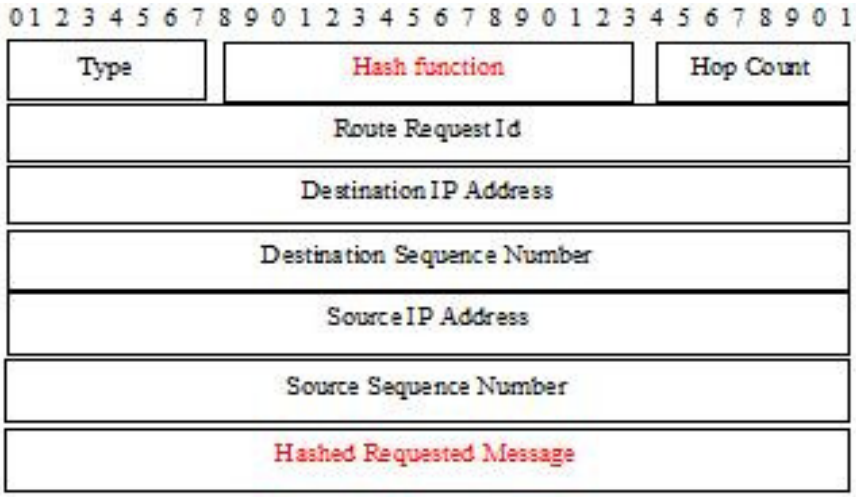


FIGURE 3: RREQ message format of SNAODV

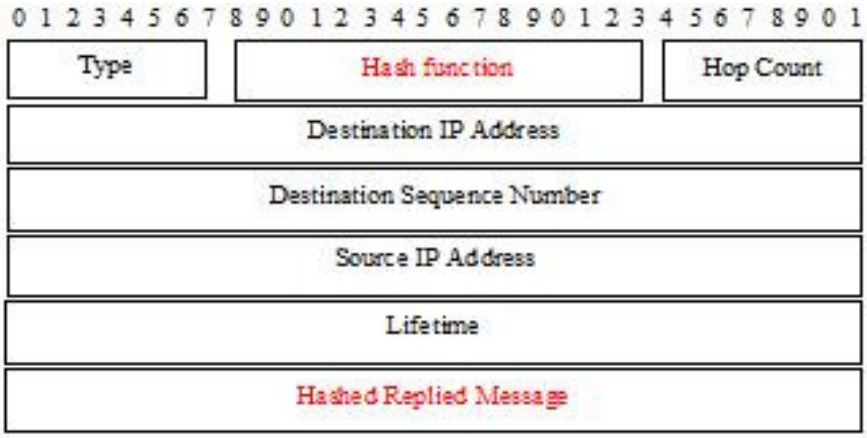


FIGURE 4: RREP message format of SNAODV

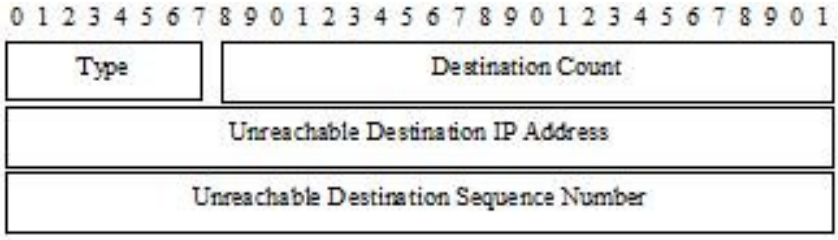


FIGURE 5: RERR message format of SNAODV

E. Algorithm for node-to-node authentication of the System Model

1. *Sender node broadcasts a message indicating the negotiation request with neighbour nodes*
   <KEY_AGREEMENT_REQ, request_id, sender_address, PK_S >
2. *Sender node gets reply a message*
   <KEY_AGREEMENT_REP, request_id, sender_address, neighbour_address, PK_N >

3. *The request node (A) negotiates a secret key with the neighbour node (B) by the following steps:*
   a. *Generate a key Ks by using a secure random number generator,*
   b. *Encrypt Ks with PK_B (node B's public key) = encrypt PK_B (Ks),*
   c. *Send an offer message*
      <KEY_PASS, encrypt PK_B (Ks)> to B,
   d. *Wait ACK (acknowledgement) from B and check message integrity to finish the negotiation*

4. *Node B sends the ACK message*
   <KEY_ PASS_ ACK, request_id, $HASH_{Ks}$ (request_id)>

5. SIMULATION AND RESULTS

Simulation of the work has been done on QualNet 5.0.1 for implementing new designed routing protocol. We have implemented RREQ and RREP message formats for new routing protocol using hash function i.e.; MD5 (Message Digest 5). It has been given in above figures. Simulation done on the following parameters basis:

Parameters	Value
Simulation Area	1500m x 1500m
Number of nodes	30 (4 nodes are wormhole)
Simulation duration	120 s
Routing protocol	AODV and SNAODV
Mobility pattern of nodes	Random waypoint

TABLE 1: Simulation setup

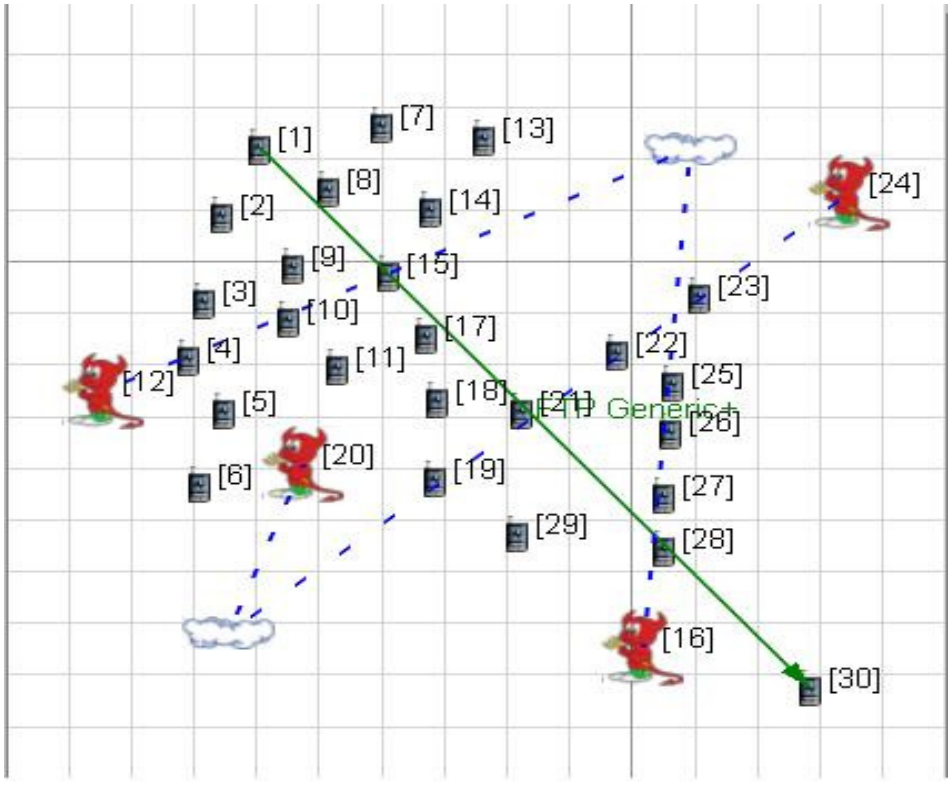


FIGURE 6: 30 nodes MANET environment with 4 blackhole nodes

Parameters	AODV	SNAODV
Throughput	2435	2700
Number of RREQ packets initiated	19	24
Number of data packets sent as source	183	208
Number of data packets received	62	64
Number of RREQ packets retried	30	29
Number of RREQ packets received by dest	19	23
Number of RREP packets initiated as dest	14	20
Number of RREP packets received as Source	15	22
Number of Data Packets Dropped for no route	Node1=31, Node2 = 1, Node11= 1	Node1=24, Node2 = 11, Node11= 0

TABLE 2: Simulation results

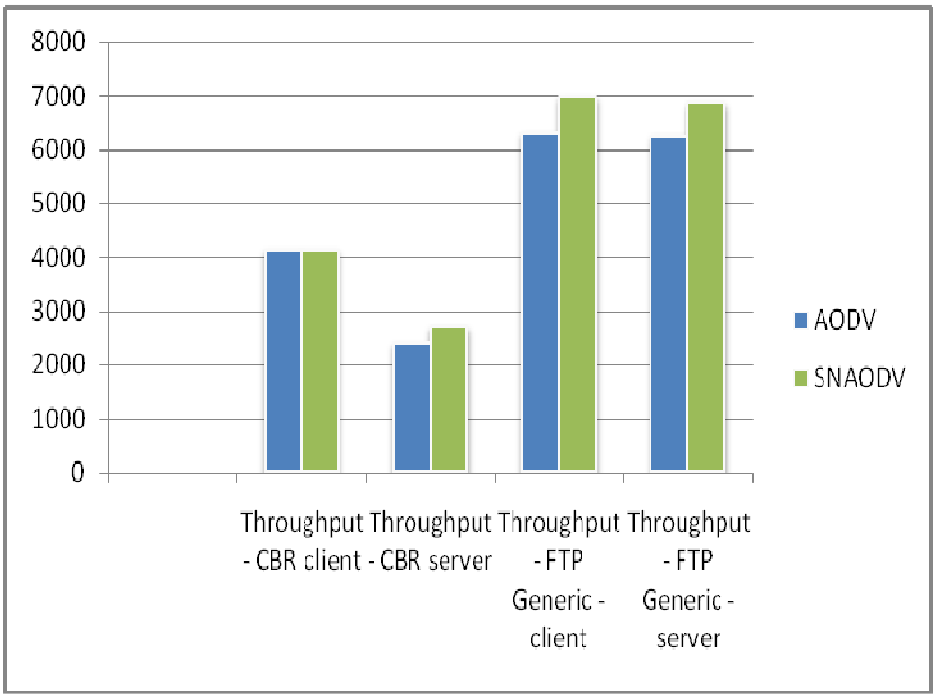


FIGURE 7: Throughput based comparison between AODV & SNAODV

The results have been come out from simulated on Qualnet 5.0 tool on the above simulation parameters and the results are being shown that the goal of new protocol to maximize the throughput. Throughput values of CBR client of both routing protocols are same while throughput values of CBR server is different in our new proposed protocol has higher values than AODV. Same process is in FTP Generic server.

6. CONCLUSION

This paper presents a new secure routing protocol for MANETs. It also provides node to node authentication and enables mobile user to ensure the authenticity of user of peer node. The significant advantage of our solution is to get all packets meaning that packet will be transmitted from source to destination without losing packets. The system model solved the security problem in the ad hoc network and is also suitable for application to other wired and

wireless network. This paper is maximizing throughput of the network on the various parameters. One advantage of the SNAODV protocol is that no key assumption is required like SAODV has.

7. REFERENCES

1. L.Zhou and Z.Haas,"Securing AdHoc Networks," IEEE Network, vol.13, no.6, page no.24–30, November/December 1999

2. B. Sukla, "Detection/Removal of Cooperative Black and Gray Hole Attack in Mobile Ad-Hoc Networks", In proceeding of the World Congress on Engineering and Computer Science 2008, October 22-24,2008,San Francisco,USA

3. Nguyen H and Morino H,"A Key Management Scheme for Mobile Ad Hoc Networks Based on Threshold Cryptography for Providing Fast Authentication and Low Signaling Load", IFIP International Federation for Information Processing 2005, LNCS 3823, page no. 905-915,2005

4. A.Kapil and S.Rana, "Identity-Based Key Management in MANETs using Public Key Cryptography", International Journal of Security (IJS), Volume (3): Issue (1), published in Computer Science journal in March, 2009.

5. S. Capkuny, L. Buttyan, J.-P. Hubaux, "Self-organized public-key management for mobile ad hoc networks", Technical Report 2002/34, EPFL/IC, May 2002.

6. Bo. Zhu, Robert H. Deng, Mohan S. Kankanhalli, Guilin Wang, "Efficient and robust key management for large mobile ad hoc networks", In Proceedings of Computer Networks 48 (2005), page no. 657–682, 2005.

7. Bing Wu, Jie Wu, Eduardo B. Fernandez, Mohammad Ilyas, Spyros Magliveras, "Secure and efficient key management in mobile ad hoc networks", Journal of Network and Computer Applications 30 (2007), page no. 937–954, 2007.

8. M. Gunasekaran, P. Sampath and B. Gopalakrishnan, "AAS: An Authenticated Acknowledgement-Based Scheme for Preventing Selfish Nodes in Mobile Ad Hoc Networks", International Journal of Recent Trends in Engineering, Vol 1, No. 1, May 2009, page no. 294-298, 2009.

9. Andreas Hafslund and Jon Andersson, Thales Norway AS, "2-Level Authentication Mechanism in an Internet connected MANET", 6th Scandinavian Workshop on Wireless Ad-hoc Networks, 2006.

10. Marcelo M. Carvalho, Cintia B. Margi, Katia Obraczka and J. J. Garcia-Luna-Aceves, "Modeling Energy Consumption in Single-Hop IEEE 802.11 Ad Hoc Networks", Proceeding in 13th International conference on Computer communication and networks 2004, 2004.

11. Qualnet 4.5.1 Network Security model library.pdf

12. K. Sanzgiri, D. Laflamme, B. Dahill, B. Levine, C. Shields and E. Royer. An Authenticated Routing for Secure Ad Hoc Networks. Journal on Selected Areas in Communications special issue on Wireless Ad hoc Networks, March 2005.

13. Man,Y.R.: Internet Security cryptographic: principles, algorithms and protocols. Wiley Publishing House, Chichester(2004).

14. Kimaya, S., et al.: Authenticated routing for ad hoc networks. Journal on Selected Areas in Communications 23(3), 598–610 (2005).

15. Zapata, M.G., Asokan, and N.: Securing Ad hoc Routing Protocols. In: Proc. of the ACM workshop on Wireless security, Atlanta, USA, pp. 1–10 (2002).

16. Kushwah, Virendra Singh and Tapaswi, Shashikala, "Securing Node In MANETs Using Node Based Key Management Scheme", In proceeding of the IEEE Xplore 2010 International Conference on Advances in Computer Engineering – ACE 2010, June 21-22, 2010 at Bangalore, INDIA.

17. C. Yang. Designing secure e-commerce with role-based access control. International Journal of Web Engineering and Technology, 3(1):73–95, 2007.

18. David F. Ferraiolo, John F. Barkley, and D. Ri hard Kuhn. A role based access control model and reference implementation within a corporate intranet. In ACM Transactions on Information Systems.

19. Xin Wang, Yanchun Zhang, Hao Shi ;" Access Control for Human Tasks in Service Oriented Architecture "; in IEEE/ the Fourth International Conference on Computer and Information Technology (CIT'04);2004 IEEE Computer, 29(2):38–47, 1996.

20. Mathias Kohler and Andreas Schaad . ProActive Access Control for Business Process-driven Environments. In IEEE/ Annual Computer Security Applications Conference 156 .2008.

21. Barkley, J., Beznosov, K., and Uppal, J., "Supporting Relationship in Access Control Using Role Based Access Control", Proceedings of ACM Role-Based Access Control Workshop, Fairfax, Virginia, USA, pp. 55-65, 1999.

22. Bernardi, P., Gandino, F., Lamberti, F., Montrucchio, B., Rebaudengo, M., and Sanchez, E.R., "An Anti-Counterfeit Mechanism for the Application Layer in Low-Cost RFID Devices", In International Conference on Circuits and Systems for Communications, IEEE, July, pp.207-211, 2006.

23. Xu Feng ,Lin Guoyuan , Huang Hao , Xie Li;"Role-based Access Control System for Web Services"; In Proceedings of the 4th IEEE International Conference on Computer and Information Technology ,2004.

24. Ateniese, G., Camenisch, J., and Madeiros, B. de, "Untraceable RFID tags via insubvertible encryption", Proceedings of the 12 ACM conference on Computer and communications security, November, pp.92-101, 2005.

A Novel Technique for Image Steganography Based on DWT and Huffman Encoding

Amitava Nag amitava.nag@aot.edu.in
Dept. of Information Technology
Academy of Technology,
West Bengal University of Technology, Hoogly – 721212,India.

Sushanta Biswas biswas.su@gmail.com
Dept. of Engineering and Technological Studies
University of Kalyani,
Kalyani, Nadia – 741 235, West Bengal, India

Debasree Sarkar dsarkar70@gmail.com
Dept. of Engineering and Technological Studies
University of Kalyani,
Kalyani, Nadia – 741 235, West Bengal, India

Partha Pratim Sarkar ppsarkar@klyuniv.ac.in
Dept. of Engineering and Technological Studies
University of Kalyani,
Kalyani, Nadia – 741 235, West Bengal, India

Abstract

Image steganography is the art of hiding information into a cover image. This paper presents a novel technique for Image steganography based on DWT, where DWT is used to transform original image (cover image) from spatial domain to frequency domain. Firstly two dimensional Discrete Wavelet Transform (2-D DWT) is performed on a gray level cover image of size M × N and Huffman encoding is performed on the secret messages/image before embedding. Then each bit of Huffman code of secret message/image is embedded in the high frequency coefficients resulted from Discrete Wavelet Transform. Image quality is to be improved by preserving the wavelet coefficients in the low frequency sub-band. The experimental results show that the algorithm has a high capacity and a good invisibility. Moreover PSNR of cover image with stego-image shows the better results in comparison with other existing steganography approaches. Furthermore, satisfactory security is maintained since the secret message/image cannot be extracted without knowing decoding rules and Huffman table.

Keywords: Steganography, Frequency Domain, DWT, Huffman Coding, Information Hiding.

1. INTRODUCTION

Information hiding is an old but interesting technology [1]. Steganography is a branch of information hiding in which secret information is camouflaged within other information. The word steganography in Greek means “covered writing” (Greek words “stegos” meaning “cover” and “grafia” meaning “writing”) [2]. The main objective of steganography is to communicate securely in

such a way that the true message is not visible to the observer. That is unwanted parties should not be able to distinguish any sense between cover-image (image not containing any secret message) and stego-image (modified cover-image that containing secret message). Thus the stego-image should not deviate much from original cover-image. Today steganography is mostly used on computers with digital data being the carriers and networks being the high speed delivery channels. Figure. 1 shows the block diagram of a simple image steganographic system.

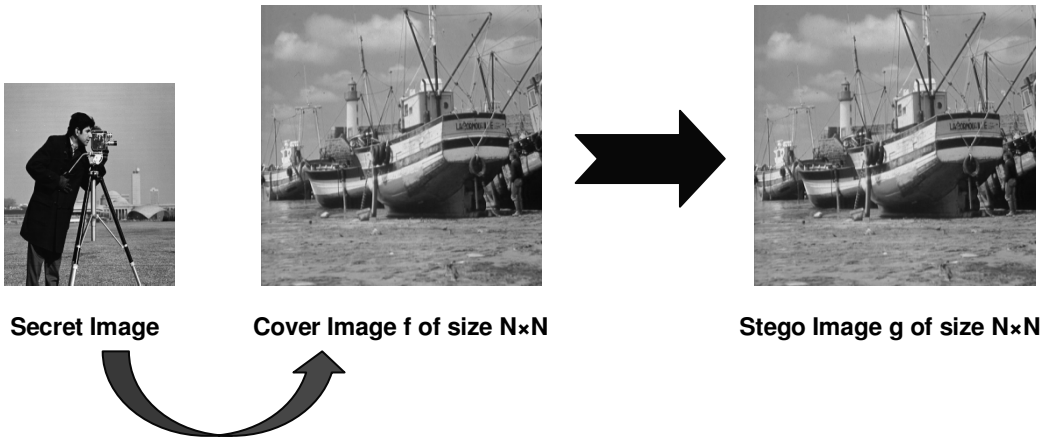


FIGURE. 1: The block diagram of a simple steganographic system

2. RELATED WORK

Image steganography schemes can be classified into two broad categories: spatial-domain [3,4,18] based and transform-domain based [5,6,7]. In spatial domain approaches, the secret messages are embedded directly. The simplest steganography is the Least Significant Bit (LSB) approach which was modified by several algorithms. In [8], a new steganography technique, named, “modified side match scheme” was proposed. It reserves the image quality, increases embedding capacity but is not robust against attack because it is a spatial domain approach and no transfer is used.

In [9] using VQ compression method is compressed the gray-level secret image before embedding. Next the compressed gray-level secret image is encrypted and then embedded into the DWT coefficients of the cover image. Though this paper provides a recovery scheme to repair the secret image if the stego-image is destroyed, but the PSNR of the stego-images are less than 36dB.

In the paper [10], the proposed steganography scheme embeds the secret message by modifying the Gabor coefficient of the cover image.

Abdelwahab and Hassan [11] used a data hiding technique in the DWT domain where 1-level DWT is performed on both secret and cover images. The disadvantage of this method is that the extracted data is not totally identical to the embedded version.

In [12], Bao P. and Ma X. embed a watermark in the singular value decomposition in the wavelet domain of an image.

In [13], Maity S.P. and Kundu M.K. proposes a blind watermarking techniques to embed the watermark redundantly in the multilevel wavelet coefficients of the LL and RR band of the cover image. The scheme is claimed to have robutness and have the ability to detect the degree of external attack already occurred in watermarked image, but PSNR is very low.

In [17], the major importance is given on the secrecy as well as the privacy of text messages, where the authors combines cryptography ,steganography and along with that an extra layer of security has been imposed in between them.

According to Raja et al. [16] fast Fourier transform (FFT) methods introduce round-off errors; thus it is not suitable for hidden communication.

The DWT based approach scheme [14] using a mapping table, the secret message is embbed in the high frequency coefficients resulted from Discrete Wavelet Transform. Among all other methods mentioned earlier, this method provides better quality of image, increases embedding capacity and is also robust against attack. Based on the same embedding capacity of [14], our proposed method improves both image quality and security.

2.1 Huffman Encoding and Huffman Table(HT)

Before embedding the secret image into cover image, it is first encoded using Huffman coding [15]. Huffman codes are optimal codes that map one symbol to one code word. For an image Huffman coding assigns a binary code to each intensity value of the image and a 2-D M2 × N2 image is converted to a 1-D bits stream with length LH < M2 × N2.

Huffman encoding is used to serve the following three:

Lossless Compression –It increases the embedding capacity

Security by means of encoding – Huffman encoded bit stream cannot reveals anything. To extract the exact meaning, the Huffman table is required to decode.

It provides one type of **authentication**, as any single bit change in the Huffman coded bit stream, Huffman table is unable to decode.

2.2 Discrete Wavelet Transform

Wavelets are special functions which (in a form analogous to sins and cosines in Fourier analysis) are used as basal functions for representing signals. The discrete wavelet transform (DWT) we applied here is Haar-DWT, the simplest DWT. In Haar-DWT the low frequency wavelet coefficient are generated by averaging the two pixel values and high frequency coefficients are generated by taking half of the difference of the same two pixels.

For 2-D images, applying DWT (Discrete Wavelet Transform) separates the image into a lower resolution approximation image or band (LL) as well as horizontal (HL), vertical (LH) and diagonal (HH) detail components as shown in figure 3.

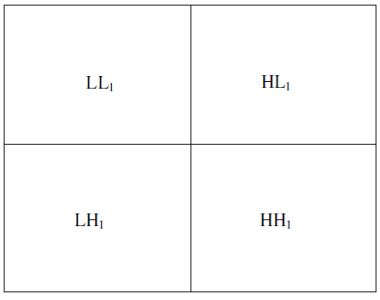


FIGURE 2: Components of 1-level 2-Dimensional Discrete Wavelet Transform

With the DWT, the significant part(smooth parts) of the spatial domain image exist in the approximation band that consists of low frequency wavelet coefficients and the edge and texture details usually exist in high frequency sub bands, such as HH, HL, and LH. The whole procedure

explained above is called the one-level 2-D Haar-DWT. The one-level 2-D Haar-DWT applied on the image "boat" is shown in Figure 4.

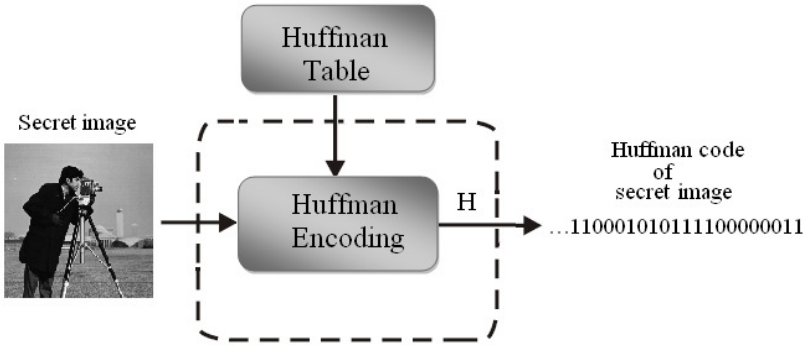
(a) (b)

FIGURE 4: (a) Original image of boat, (b) Result after the one-level 2-D Haar-DWT

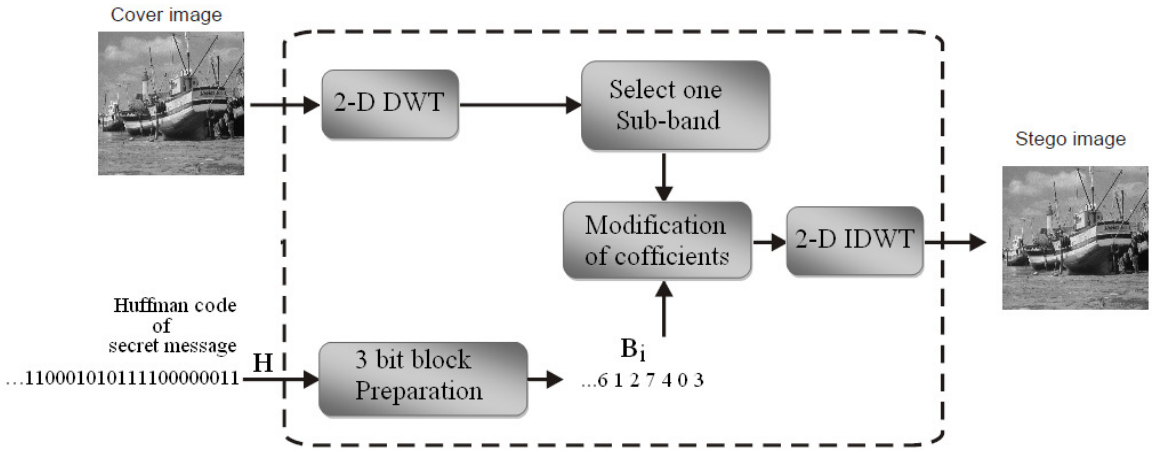
The human eyes are not sensitive to the small changes in the edges and textures of an image but very sensitive to the small changes in the smooth parts. This allows the secret message/image to be embedded at high frequency sub-bands without being perceived by the human eye.

3. PROPOSED IMAGE STEGANOGRAPHY ALGORITHM

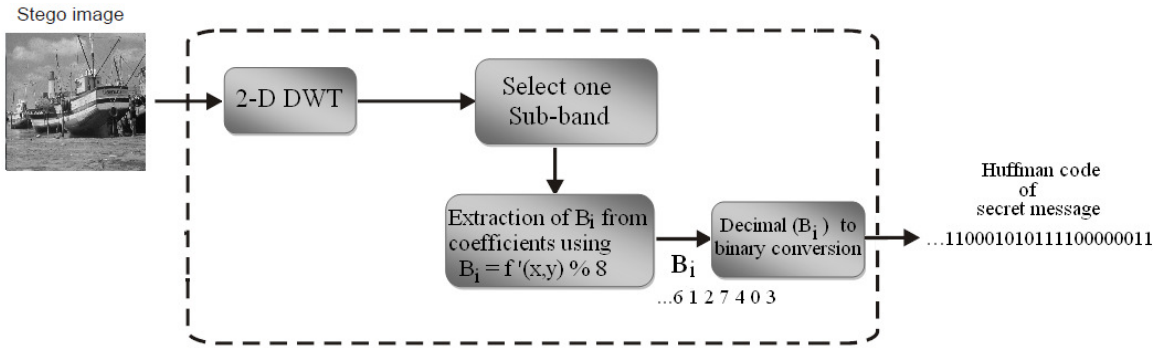
Hiding the secret message/image in the special domain can easily be extracted by unauthorized user. In this paper, we proposed a steganography technique using DWT (Discrete Wavelet Transform) for hiding a large amount of data with high security, a good invisibility and no loss of secret message. The basic idea to hide information using DWT is to alter the magnitude of the DWT coefficients of three sub-bands, HH, HL, and LH of cover image. The schematic/ block diagram of the whole process is given in figure 2((a) to (d)).



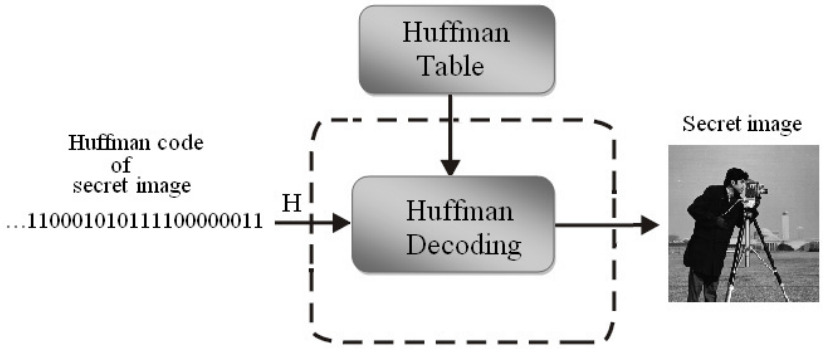
(a) Huffman encoding of secret image (or message)



(b) Insertion of a Huffman code of secret image (or message) into a Cover image



(c) Removal of Huffman code of secret Image (or message)



(d) Huffman decoding of secret image (or message)

3.1 3-bit Block Preparation

Huffman code H is now decomposed into 3-bits blocks and thus form a decimal value ranging from 0 to 7. For example, the binary sequence …. 110 001 010 111 100 000 011 …. will be changed to the decimal sequence (D)… 6 1 2 7 4 0 3… . The decimal sequence (D) is defined as follows:

$$D = \left\{ B_i \mid 1 \leq i \leq \frac{8 \times M \times N}{3}, B_i \in \{0,1,2,3,4,5,6,7\} \right\}$$

3.2 Embedding of Secret Message / Image

We proposed the secret message/image embedding scheme comprises the following five steps:

Step 1: Decompose the cover image by using Haar wavelet transform.

Step 2: Huffman encoding.
Perform Huffman encoding on the 2-D secret image S of size M2 × N2 to convert it into a 1-D bits stream H.

Step 3: 3-bit block (Bi) preparation
Huffman code H is decomposed into 3-bits blocks and thus form a decimal value ranging from 0 to 7. For example, the binary sequence …. 110 001 010 111 100 000 011 will be changed to the decimal sequence (Bi) … 6 1 2 7 4 0 3.

Step 4: Bits replacement
Select one sub-band for embedding the secret message. If we donate 'f ' as coefficients matrix of the selected sub-band, then using the following equation, the 3 least significant bits of wavelet coefficients is replaced by the 3 bits of Huffman encoded bit stream in the form of 3 bit block Bi.

$$f'(x,y) = f(x,y) - f(x,y) \% 8 + Bi \quad \text{----------(1)}$$

Step 5: IDWT
Apply the Haar inverse DWT (IDWT) on the DWT transformed image, including the modified sub-band to produce a new image f1 which contains secret image.

Embedding Algorithm

Input: An M1×N1 carrier image and a secret message/image.
Output: A stego-image.

1. Obtain Huffman table of secret message/image.
2. Find the Huffman encoded binary bit stream of secret-image by applying Huffman encoding technique using Huffman table obtained in step 1.
3. Decompose the cover image by using Haar wavelet transform
4. Calculate the size of encoded bit stream in bits.
5. Repeat for each bit obtained in step 4
   (a) Insert the 3 consecutive bits into 3 LSB position in each DWT coefficient of the selected sub-band.
6. Repeat for each bit obtained in step 2
(a) Insert the 3 consecutive bits into 3 LSB position in each DWT coefficient(excluding the first four coefficients in each sub-band) of the selected sub-band.
7. Repeat for each bit of the Huffman table
   (a) Insert the 3 consecutive bits into 3 LSB position in each DWT coefficient of the selected sub-band.
8. Apply inverse DWT.
9. End.

3.3 Extraction of the Secret Message / Image

The stego-image is received in spatial domain. DWT is applied on the stego-image to transform the stego-image from spatial domain to frequency domain. The following formula is used to extract bit stream from wavelet coefficients in the form of blocks Bi .

$$Bi = f'(x,y) \% 8 \quad \text{---------- (2)}$$

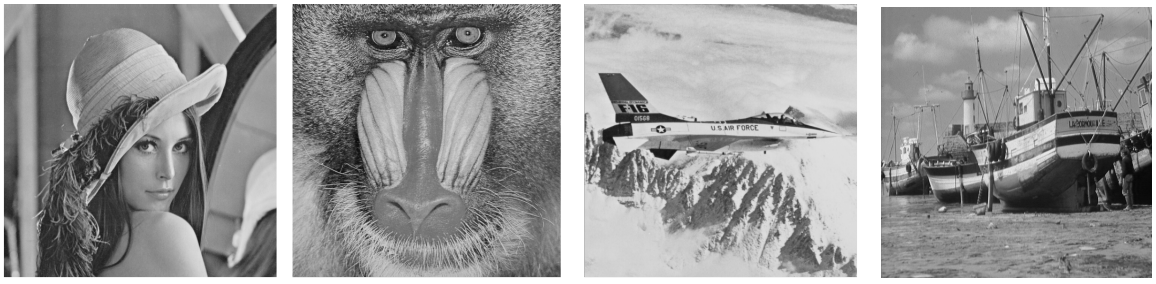
The size of the encoded bit stream and the encoded bit stream of secret message/image are extracted along with the Huffman table of the secret message/image. The block diagram of the extracting process is given in figure 4((c) and (d)) and the extracting algorithm as follows:

Extraction Algorithm

Input: An M1×N1 Stego-image.
Output: Secret image.
1. Apply DWT to the stego-image.
2. The size of the encoded bit stream is extracted from 1st four DWT coefficients in each subband by collecting the 3 least significant bits.
3. The 3 least significant bits of all of the DWT coefficients inside each sub-bands(excluding the first four coefficients in each sub-bands) are collected and added to a 1-D array.
4. Repeat step 3 until the size of the 1-D array becomes equal to the size extracted in step 2.
5. Construct the Huffman table by extracting 3 bits from the LSB of all of the DWT coefficients inside each sub-bands excluding the coefficients used in step 2 and step 3.
6. Decode the 1-D array obtained in step 3 using the Huffman table obtained in step 5.
7. End.

4. SIMULATION RESULTS

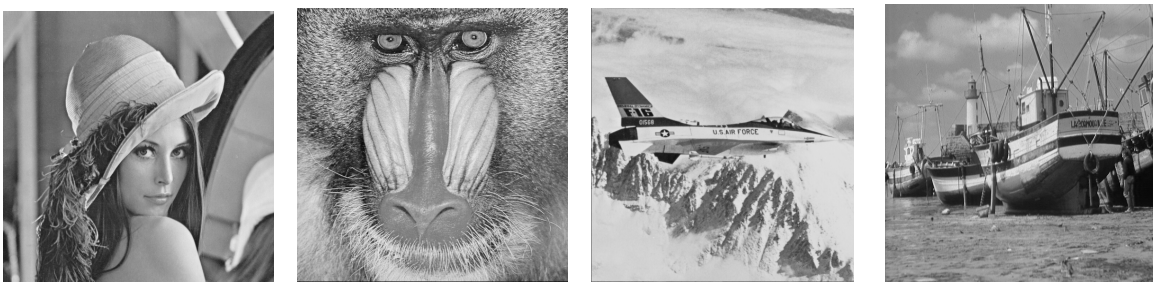
In this section, some experiments are carried out to prove the efficiency of the proposed scheme. The proposed method has been simulated using the MATLAB 7 program on Windows XP platform. A set of 8-bit grayscale images of size 512 × 512 are used as the cover-image to form the stego-image. The Figure 6 (a) – (d) shows the original cover (carrier) images and Figure 7 shows the original secret message.



(a) Lena **(b) Baboon** **(c) Airplane** **(d) Boat**

FIGURE 6: Four cover-images for simulations

FIGURE 7: Secret Image to be embedded



(a) Lena (b) Baboon (c) Airplane (d) Boat

FIGURE 8: stego-images of the proposed methods

FIGURE 9: Extracted Secret Image

Here we are embedding a 8-bit grayscale image of size 256 × 248 into a 8-bit grayscale images of size 512 × 512 i.e. 507904 bits are embedded into a 512 × 512 carrier image. Here, PSNR value is utilized to evaluate the invisibility of the stego-images.

TABLE 1: COMPARISON OF RESULTS FOR THE PROPOSED METHOD AND DWT BASED MODEL[14]

Cover Image (512 ×512)	DWT base [14]		Our Method	
	Capacity (bits)	PSNR (dB)	Capacity (bits)	PSNR (dB)
Lena	507856	46.0882	507856	54.93
Airplan	507856	45.9961	507856	54.67
Baboon	507670	46.1948	507670	55.11
Boat	507867	46.1385	507867	54.80

To compare the proposed approach with the DWTIS method [14], table 1 exhibit the capacity and PSNR after the secret data is embedded using those two approaches. From table 1 it is clear that for the same capacity, the PSNR of our proposed algorithm is better than the one in reference [14]. From table 1, it is noticed that for all images, PSNR is nearly 55. Figure 8 shows the resulted stego-images of the proposed methods and figure 9 extracted Image

5. CONCLUSION

Generally, image steganography method does not provide much attention on the basic demand of secrecy and privacy. In this paper, the major importance is given on the secrecy as well as the privacy of information. The embedding process is hidden under the transformation (DWT and IDWT) of cover image. These operations provide sufficient secrecy. On the other hand to obtain privacy we have used huffman encoding. In a similar type of paper[14] the authors have provided their attention on the security by a well designed mathemetical mapping. Our paper deals with the Huffman encoding. After comparision it is found that in our paper PSNR is higher than the mentioned paper. Here lies the novelity of our research work.

6. REFERENCES

[1] N. F. Johnson and S. Katzenbeisser, "*A survey of steganographic techniques*". Information Hiding, Artech House, pp. 43-78, 2000.

[2] Moerland, T. "*Steganography and Steganalysis*". Leiden Institute of Advanced Computing Science, www.liacs.nl/home/ tmoerl/privtech.pdf

[3] Chan, C.K. and Cheng. L.M. "*Hiding data in image by simple LSB substitution*". Pattern Recognition, 37: 469 – 474, 2003.

[4] Chang, C.C and Tseng, H.W. "*A Steganographic method for digital images using side match*". Pattern Recognition Letters, 25: 1431 – 1437, 2004.

[5] Chen, T.S., Chang C.C., and Hwang, M.S. "*A virtual image cryptosystem based upon vector quantization*". IEEE transactions on Image Processing, 7,(10): 1485 – 1488, 1998.

[6] Chung, K.L., Shen, C.H. and Chang, L.C. "*A novel SVD- and VQ-based image hiding scheme. Pattern Recognition Letters"* 22: 1051 – 1058, 2001.

[7] Iwata, M., Miyake, K., and Shiozaki, A. "*Digital Steganography Utilizing Features of JPEG Images, IEICE Transfusion Fundamentals*". E87-A(4):929 – 936, 2004.

[8] Chen, P.Y. and Wu, W.E. "*A Modified Side Match Scheme for Image Steganography*". International Journal of Applied Science and Engineering, 7(1): 53 – 60, 2009..

[9] Chu, Y.P., Guo, S.W., Chan, Y.K. and Wu, H.C. "*Image Hiding Based on a Hybrid Technique of VQ Compression and Discrete Wavelet Transform*", International Computer Symposium, 313-317,2004.

[10] Mythreyi S and Vaidehi V. "*Gabor Transform based Image Steganography*", IETE Journal of Research, 53(2):. 103 – 112,2007.

[11] A.A. Abdelwahab, L.A. Hassan. "*A discrete wavelet transform based technique for image data hiding*", in: Proceedings of 25th National Radio Science Conference, Egypt, 2008.

[12] Bao, P and Ma, X. "*Image Adaptive Watermarking Using Wavelet Domain Singular Value Decomposition*", IEEE Transaction on Circuits and Systems for Video Technology, 15(1):2005

[13] Maity S.P. and Kundu M.K., "*A Blind CDMA Image Watermarking Scheme in Wavelet Domain*" IEEE International Conference:2633 – 2336,2004.

[14] Chen, P.Y. and Wu, W.E. "*A DWT Based Approach for Image Steganography*", International Journal of Applied Science and Engineering, 4,3: 275 –290.

[15] Jayaraman, S., Esakkirajan, S. and Veerakumar, T. "*Digital Image Processing*", Tata McGraw Hill Education Private Limited, India, 2009.

[16] K.B. Raja, C.R. Chowdary, K.R. Venugopal, L.M. Patnaik. "A secure image steganography using LSB, DCT and compression techniques on raw images". Proceedings of IEEE 3rd International Conference on Intelligent Sensing and Information Processing, ICISIP'05, Bangalore, India, 14–17 December 2005..

[17]Debnath Bhattacharyya, Poulami Das, Samir kumar Bandyopadhyay and Tai-hoon Kim. "*Text Steganography: A Novel Approach*," International Journal of Advanced Science and Technology, vol.3, pp.79-85, February2009.

[18] H. Arafat Ali. "*Qualitative Spatial Image Data Hiding for Secure Data Transmission*". GVIP Journal, 7(1):35-43, 2007.

Performance Comparison of Automatic Speaker Recognition using Vector Quantization by LBG KFCG and KMCG

Dr. H. B. Kekre hbkekre@yahoo.com
Senior Professor, MPSTME, SVKM's NMIMS University Mumbai-400056.

Ms. Vaishali Kulkarni vaishalikulkarni6@yahoo.com
Ph.D Research Scholar, MPSTME, SVKM's NMIMS, Mumbai-400056.
Associate Professor, MPSTME, NMIMS University Mumbai-400056.

Abstract

In this paper, three approaches for automatic Speaker Recognition based on Vector quantization are proposed and their performances are compared. Vector Quantization (VQ) is used for feature extraction in both the training and testing phases. Three methods for codebook generation have been used. In the 1st method, codebooks are generated from the speech samples by using the Linde-Buzo-Gray (LBG) algorithm. In the 2nd method, the codebooks are generated using the Kekre's Fast Codebook Generation (KFCG) algorithm and in the 3rd method, the codebooks are generated using the Kekre's Median Codebook Generation (KMCG) algorithm. For speaker identification, the codebook of the test sample is similarly generated and compared with the codebooks of the reference samples stored in the database. The results obtained for the three methods have been compared. The results show that KFCG gives better results than LBG, while KMCG gives the best results.

Keywords: Speaker Identification, Vector Quantization (VQ), Code Vectors, Code Book, Euclidean Distance, LBG, KFCG, KMCG

1. INTRODUCTION

The goal of speaker recognition is to extract the identity of the person speaking. Speaker recognition technology [1] – [3] makes it possible to use the speaker's voice to control access to restricted services, for example, for giving commands to computer, phone access to banking, database services, shopping or voice mail, and access to secure equipment. Speaker Recognition is the process of automatically recognizing who is speaking on the basis of individual information included in speech signals. It can be divided into Speaker Identification and Speaker Verification [3] – [5]. Speaker identification determines which registered speaker provides a given utterance from amongst a set of known speakers (also known as closed set identification). Speaker verification accepts or rejects the identity claim of a speaker (also known as open set identification).

Speaker Identification task can be further classified into text-dependent or text-independent task [4] – [5]. In the former case, the utterance presented to the system is known beforehand. In the latter case, no assumption about the text being spoken is made, but the system must model the general underlying properties of the speaker's vocal spectrum. In general, text-dependent systems are more reliable and accurate, since both the content and voice can be compared [3], [4].

Speaker Recognition systems have been developed for a wide range of applications [6] – [9]. Still, there are a number of practical limitations because of which widespread deployment of applications and services is not possible.

Vector Quantization (VQ) maps a 'k' dimensional vector space to a finite set C = {C1, C2, C3… CN}.The set C is called codebook consisting of 'N' number of codevectors and each code vector Ci= {ci1, ci2, ci3… cik} is of dimension k. The key to VQ is the good codebook. The method most commonly used to generate codebook is the Linde-Buzo-Gray (LBG) algorithm [10], [11] which is also called as Generalized Lloyd Algorithm (GLA).VQ [10] – [12], [20] is an efficient data compression technique and has been used in various applications involving VQ-based encoding and VQ based recognition. VQ has been very popular in the field of speech recognition. [13] – [19]. We have proposed speaker identification using VQ by LBG algorithm [24] and KFCG algorithm [25]. In this paper we propose speaker identification using VQ by KMCG algorithm. Also comparison of the results obtained by LBG, KFCG and KMCG is shown.

Recognition systems have been developed for a wide range of applications. [15] Although many new techniques were invented and developed, there are still a number of practical limitations because of which widespread deployment of applications and services is not possible. Vector Quantization [1] - [4] is an efficient data compression technique and has been used in various applications involving VQ-based encoding and VQ based recognition. Vector Quantization has been very popular in the field of speech recognition. [5] – [7], [13, 14].

The recognition Process

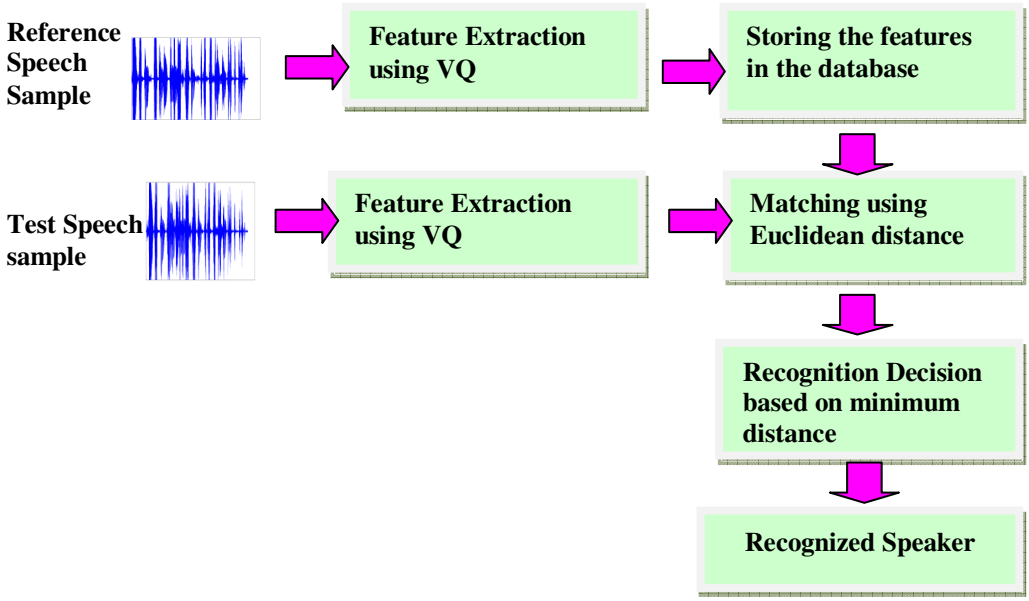


FIGURE 1: Speaker identification system

The General scheme for Speaker Identification using vector quantization is shown in Fig. 1. At the training stage, features are extracted from the reference speech samples by using VQ and these are stored as feature vectors in the database. In the testing phase, again features are extracted from the test pattern and compared against the reference templates at the pattern matching stage. Matching is done by comparing the Euclidean Distance. After comparison, the test pattern is labeled to a speaker model at the decision stage. The labeling decision is generally based on the minimum distance criterion.

In the next section we present the two codebook generation approaches. Section 3 describes the three codebook generation algorithms. Section 4 consists of results and conclusions in section 5.

2. CODE BOOK GENERATION APPROACH

a. Without Overlap

The speech signal has amplitude range from -1 to +1. It was first converted into positive values by adding +1 to all the sample values. Then the sample values were converted into a 16 dimensional vector space. (Training vector 'v'). The code books for different size of code vectors were found using the algorithms (LBG, KFCG and KMCG) discussed in the next section.

b. With Overlap

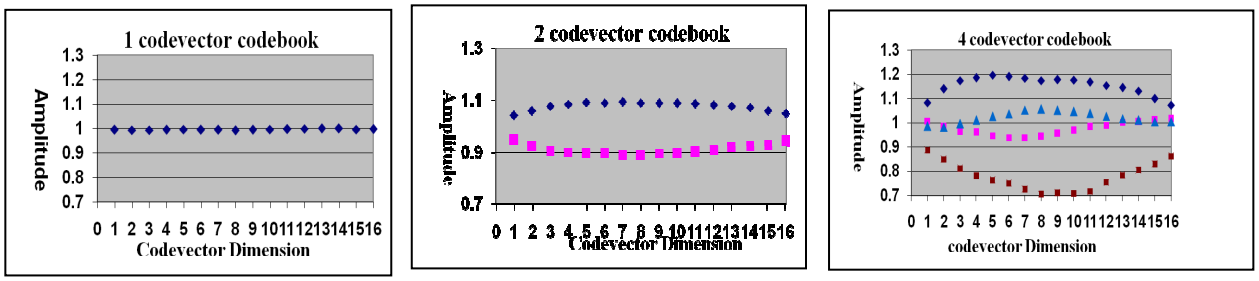
The speech signal was converted into positive range in the same manner as in approach A. The samples were converted into a 16 dimensional vector space by considering an overlap of 4 between the samples of consecutive blocks. E.g. the first vector was from sample 1 to 16, whereas second vector was from 13 to 28 and the third from 25 to 40 and so on. The code books were then generated similarly as in approach A.

3. CODEBOOK GENERATION ALGORITHMS

c. LBG (Linde-Buzo-Gray) Algorithm

For generating the codebooks, the LBG algorithm [11, 12] is used. The LBG algorithm steps are as follows [1, 11]:
Design a 1-vector codebook; this is the centroid of the entire set of training vectors.
Double the size of the codebook by splitting each current codebook yn according to the rule
yn+ = yn(1+ε)
yn+ = yn(1-ε)
where n varies from 1 to the current size of the codebook, and ε is a splitting parameter.
Find the centroids for the split codebook. (i.e., the codebook of twice the size)
Iterate steps 2 and 3 until a codebook of size M is designed.

Figure 2 shows the process of obtaining four codevectors using the LBG algorithm. Figure 2(A) shows the one vector codebook which is the centroid of the entire training set. Figure 2(B) shows the two vector codebook obtained by splitting the training set. Figure 2(C) shows the four vector codebook.



(A) (B) (C)

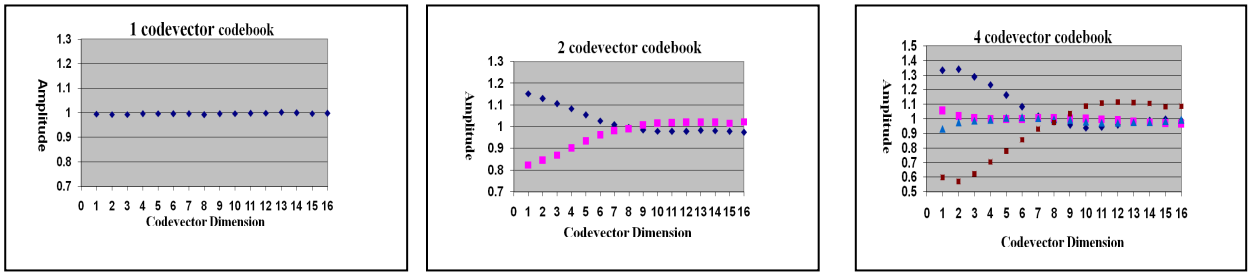
FIGURE 2: Generation of four codevectors using LBG algorithm

d. Kekre's Fast Codebook Generation Algorithm (KFCG)

In this algorithm for generating the codebook the following procedure is used [20] – [23]:

1. Initially we have only one cluster which is the entire set of training vectors. Design a 1-vector codebook; which is the centroid of the cluster.
2. Split the cluster into two by comparing the first element of all the training vectors in the cluster with the first element of the centroid as follows:
3. If vi, 1>c1, 1 then vi,1 is grouped into C1 (cluster 1).
4. Else vi,1 is grouped into C2 (cluster 2).
5. Where v is the training vector and c is the centroid.
6. Find the centroids of C1 and C2 (this is 2-vector codebook). Now split C1 into two clusters by comparing the second element of all the training vectors in C1 with the second element of its centroids explained in step 2 above. Similarly split C2 into two clusters by comparing the second element of all the training vectors in C2 with the second element of its centroid.
7. Now four clusters are formed. Centroids of these four clusters are computed (this is 4-vector codebook). These four clusters are split further by comparing the third element of the training vectors in that cluster with the third element of its centroid as explained in step 2 above.
8. The process is repeated until a codebook of size M is designed.

Figure 3 shows the process of obtaining four codevectors using the KFCG algorithm. Figure 3(A) shows the one vector codebook which is the centroid of the entire training set. Figure 3(B) shows the two vector codebook obtained by splitting the training set by comparing with the first element of all the training vectors in the cluster with the first element of the centroid. Figure 3(C) shows the four vector codebook obtained similarly by splitting the two clusters.



(A) (B) (C)

FIGURE 3: Generation of four codevectors using KFCG algorithm

a.

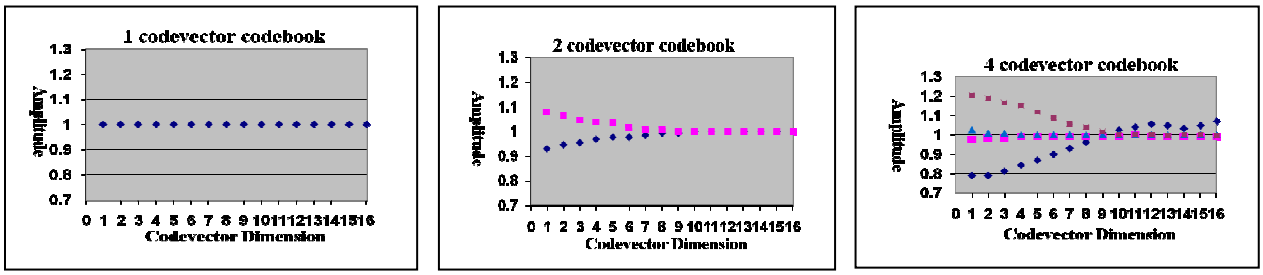
A. Kekre's Median Codebook Generation Algorithm (KMCG)

In this algorithm for generating the codebook the following procedure is used [26]:

1. Initially we have only one cluster which is the entire set of training vectors. Sort the training vectors with respect to the first element of the vector, i.e. with respect to the first column of vector v. Design a 1-vector codebook; which is the median of the cluster.
2. The training vector is then split into two by considering the median. Each of these parts is then again sorted with respect to the second element of the training vectors i.e. with respect to the second column of the vectors. We will obtain two clusters with equal number of training vectors. The median of these clusters is then found and thus we get a two vector codebook.
3. Each of the two vectors is again split into half i.e. four parts. These four parts are further sorted with respect to the third column of the vectors and four clusters are obtained and accordingly four codevectors are obtained.
4. The above process is repeated till we get the codebook of the desired size.

Figure 4 shows the process of obtaining four codevectors using the KMCG algorithm. Figure 4(A) shows the one vector codebook which is the median of the entire training set. Figure 4(B) shows the two vector codebook obtained by dividing the training set into half by sorting with respect to the first element of all

the training vectors i.e. with respect to the first column of all the training vectors in the cluster, and then taking the median of these two clusters. Figure 4(C) shows the four vector codebook obtained similarly by diving the two clusters into half by sorting with respect to the second column of the training vectors in the cluster and then taking median of these four clusters.



(A) (B) (C)

FIGURE 4: Generation of four codevectors using KMCG algorithm

4. RESULTS

Basics of Speech Signal

The speech samples used in this work are recorded using Sound Forge 4.5. The sampling frequency is 8000 Hz (8 bit, mono PCM samples). Table 1 shows the database description. The samples are collected from different speakers. Samples are taken from each speaker in two sessions so that training model and testing data can be created. Twelve samples per speaker are taken. The samples recorded in one session are kept in database and the samples recorded in second session are used for testing.

TABLE 1: Database Description

Parameter	Sample characteristics
Language	English
No. of Speakers	50
Speech type	Read speech
Recording conditions	Normal. (A silent room)
Sampling frequency	8000 Hz
Resolution	8 bps

The feature vectors of all the reference speech samples are stored in the database in the training phase. In the matching phase, the test sample that is to be identified is taken and similarly processed as in the training phase to form the feature vector. The stored feature vector which gives the minimum Euclidean distance with the input sample feature vector is declared as the speaker identified.

Figure 5 shows the results obtained for text-dependent system by varying the number of feature vectors (code vectors) without overlap for a sample set of 50 speakers. As seen from the figure, for text-dependent samples, maximum accuracy (76%) is achieved with 8 feature vectors for LBG (distortion of 0.005). For LBG (distortion of 0.01) the maximum accuracy obtained is 74%. In both the cases (distortion of 0.005 and 0.01), the accuracy decreases with the increase in the number of feature vectors. For KFCG the results are better and consistent. Accuracy does not drop as the number of feature vectors is increased. The maximum accuracy is 90% for feature vectors size of 64 or more. For KMCG accuracy

increases as the number of feature vectors (code vector size) is increased. The maximum accuracy is 96% using 128 feature vectors (codebook size).

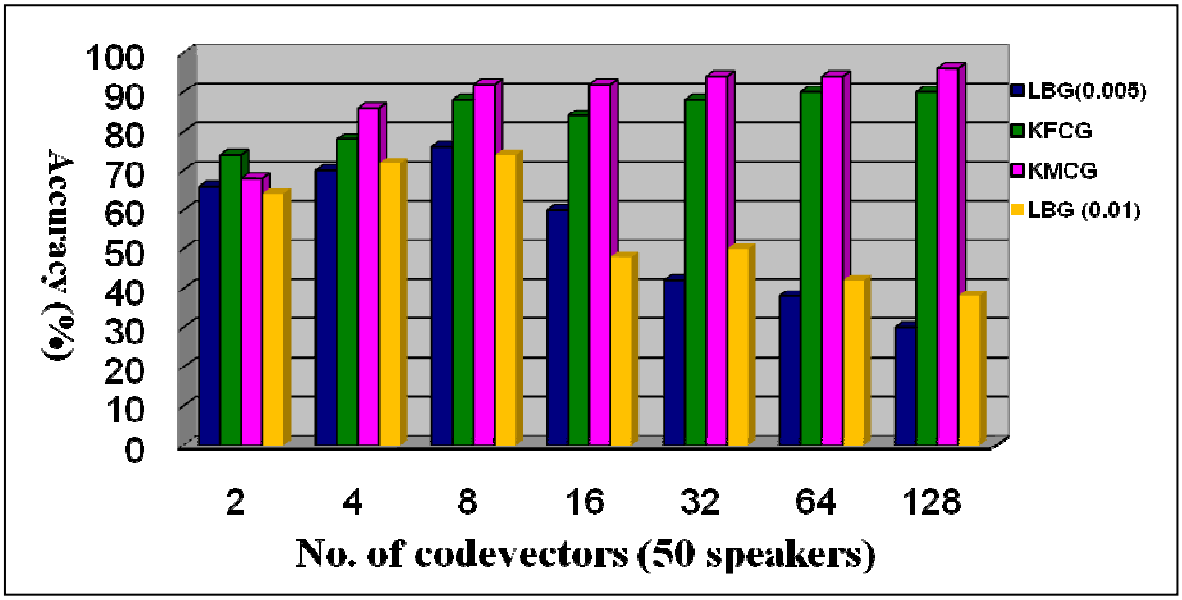


FIGURE 5: Performance comparison of LBG (0.005), LBG (0.01), KFCG and KMCG (without overlap)

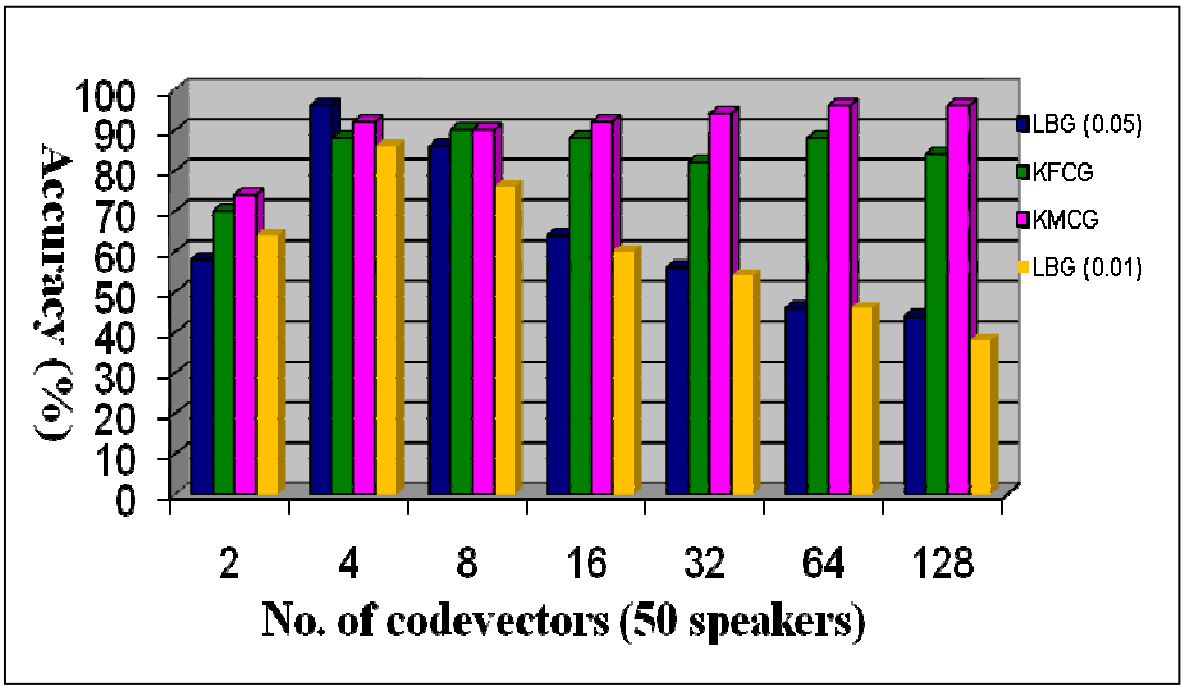


FIGURE 6: Performance Comparison of LBG (0.005), LBG (0.01), KFCG and KMCG (with overlap)

Figure 6 shows the results obtained for text-dependent identification by varying the number of features for a sample set of 50 speakers with overlap. Here LBG (distortion of 0.005) gives maximum accuracy of 96% using only four feature vectors (code vector size). LBG (distortion of 0.01) gives maximum accuracy of 86% using four feature vectors. Again the accuracy decreases as the number of feature vectors are increased for both the cases. Here the performance of KFCG is consistent and it gives maximum accuracy of 90% using 8 feature vectors (code vector size), whereas the performance of KMCG increases as the number of feature vectors are increased. The maximum accuracy is 96% using 128 feature vectors (code vector size). As KFCG and KMCG algorithms for codebook generation are based on comparison they are less complex and very fast compared to LBG which needs Euclidean distance calculations. For LBG the number of calculations required for generating the vectors by Euclidean distance comparison for a 16-dimensional vector (16 additions + 16 Multiplications + 16 comparisons) are much more than KFCG and KMCG (16 comparisons). This reduces computational time by a factor ten.

5. CONCLUSION

Very simple techniques based on the lossy compression using vector quantization have been introduced. The results show that accuracy decreases as the number of feature vectors are increased with or without overlap for LBG. For KFCG, the results are consistent and also accuracy increases with the increase in the number of feature vectors. KMCG gives the best results for with and without overlap and the accuracy increases as the number of feature vectors are increased. Also KFCG and KMCG algorithms for code book generations are simple and faster as only simple comparisons are required as against Euclidean distance calculations for LBG.

6. REFERENCES

1. Lawrence Rabiner, Biing-Hwang Juang and B.Yegnanarayana, “Fundamental of Speech Recognition”, Prentice-Hall, Englewood Cliffs, 2009.

2. S Furui, “50 years of progress in speech and speaker recognition research”, ECTI Transactions on Computer and Information Technology, Vol. 1, No.2, November 2005.

3. D. A. Reynolds, “An overview of automatic speaker recognition technology”, Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP’02), 2002, pp. IV-4072–IV-4075.

4. Joseph P. Campbell, Jr., Senior Member, IEEE, “Speaker Recognition: A Tutorial”, Proceedings of the IEEE, vol. 85, no. 9, pp. 1437-1462, September 1997.

5. F. Bimbot, J.-F. Bonastre, C. Fredouille, G. Gravier, I. Magrin-Chagnolleau, S. Meignier, T. Merlin, J. Ortega-García, D.Petrovska-Delacrétaz, and D. A. Reynolds, “A tutorial on text-independent speaker verification,” EURASIP J. Appl. Signal Process., vol. 2004, no. 1, pp. 430–451, 2004.

6. D. A. Reynolds, “Experimental evaluation of features for robust speaker identification,” IEEE Trans. Speech Audio Process., vol. 2, no. 4, pp. 639–643, Oct. 1994.

7. Tomi Kinnunen, Evgeny Karpov, and Pasi Fr¨anti, “Realtime Speaker Identification”, ICSLP2004.

8. Marco Grimaldi and Fred Cummins, “Speaker Identification using Instantaneous Frequencies”, IEEE Transactions on Audio, Speech, and Language Processing, vol., 16, no. 6, August 2008.

9. Zhong-Xuan, Yuan & Bo-Ling, Xu & Chong-Zhi, Yu. (1999). “Binary Quantization of Feature Vectors for Robust Text-Independent Speaker Identification” in IEEE Transactions on Speech and Audio Processing, Vol. 7, No. 1, January 1999. IEEE, New York, NY, U.S.A.

10. R. M. Gray.: ‘Vector quantization’, IEEE ASSP Marg., pp. 4-29, Apr. 1984.

11. Y. Linde, A. Buzo, and R. M. Gray.: 'An algorithm for vector quantizer design," IEEE Trans. Commun.', vol. COM-28, no. 1, pp. 84-95, 1980.

12. A. Gersho, R.M. Gray.: 'Vector Quantization and Signal Compression', Kluwer Academic Publishers, Boston, MA, 1991.

13. F. K. Soong, et. al., "A vector quantization approach to speaker recognition", At & T Technical Journal, 66, pp. 14-26, 1987.

14. A. E. Rosenberg and F. K. Soong, "Evaluation of a vector quantization talker recognition system in text independent and text dependent models", Computer Speech and Language 22, pp. 143-157, 1987.

15. Jeng-Shyang Pan, Zhe-Ming Lu, and Sheng-He Sun.: 'An Efficient Encoding Algorithm for Vector Quantization Based on Subvector Technique', IEEE Transactions on image processing, vol 12 No. 3 March 2003.

16. F. Soong, E. Rosenberg, B. Juang, and L. Rabiner, "A Vector Quantization Approach to Speaker Recognition", AT&T Technical Journal, vol. 66, March/April 1987, pp. 1426.

17. Md. Rashidul Hasan, Mustafa Jamil, Md. Golam Rabbani Md. Saifur Rahman , "Speaker Identification using Mel Frequency Cepstral Coefficients", 3rd International Conference on Electrical & Computer Engineering ICECE held at Dhaka, Bangladesh , 28-30 December 2004.

18. Poonam Bansal, Amrita Dev, Shail Bala Jain, "Automatic Speaker Identification using Vector Quantization", Asian Journal of Information Technology 6 (9): 938-942, 2007.

19. Jyoti Singhai, "Automatic Speaker Recognition :An Approach using DWT based Feature Extraction and Vector Quantization", IETE Technical Review, vol. 24, No 5, pp 395-402, September-October 2007

20. H. B. Kekre, Tanuja K. Sarode, "Speech Data Compression using Vector Quantization", WASET International Journal of Computer and Information Science and Engineering (IJCISE), Fall 2008, Volume 2, Number 4, pp.: 251-254, 2008. http://www.waset.org/ijcise.

21. H. B. Kekre, Tanuja K. Sarode, "New Fast Improved Codebook Generation Algorithm for Color Images using Vector Quantization," International Journal of Engineering and Technology, vol.1, No.1, pp. 67-77, September 2008.

22. H. B. Kekre, Tanuja K. Sarode, "Fast Codebook Generation Algorithm for Color Images using Vector Quantization," International Journal of Computer Science and Information Technology, Vol. 1, No. 1, pp: 7-12, Jan 2009.

23. H. B. Kekre, Tanuja K. Sarode, "An Efficient Fast Algorithm to Generate Codebook for Vector Quantization," First International Conference on Emerging Trends in Engineering and Technology, ICETET-2008, held at Raisoni College of Engineering, Nagpur, India, 16-18 July 2008, Avaliable at online IEEE Xplore.

24. H B Kekre, Vaishali Kulkarni, "Speaker Identification by using Vector Quantization", International Journal of Engineering Science and Technology, May 2010 edition.

25. H B Kekre, Vaishali Kulkarni, "Performance Comparison of Speaker Recognition using Vector Quantization by LBG and KFCG", International Journal of Computer Applications, vol. 3, July 2010.

26. H B Kekre, Tanuja Sarode, "2-level Vector Quantization Method for Codebook Design using Kekre's Median Codebook Generation Algorithm",International Journal of Advances in Computational

Sciences and Technology Year:2009,Volume:2,Issue:2.

27. H.B. Kekre, Archana Athawale, Tanuja K. Sarode, Kalpana Sagvekar, “Comparative Performance of Information Hiding in Vector Quantized Codebooks using LBG, KPE, KMCG and KFCG”, International Journal of Computer Science and Information Security, 2010 Vol: 8 Issue: 2,pp 89-95.

28. H B Kekre, Archana Athawale, Tanuja Sarode and Kalpana Sagvekar, “Increased Capacity of Information Hiding using Mixed Codebooks of Vector Quantization Algorithms: LBG, KPE and KMCG, International Journal of Advances in Computational Sciences and Technology, Volume 3 Number 2 (2010) pp. 245–256.

A Simple Agent Based Model for Detecting Abnormal Event Patterns in a Distributed Wireless Sensor Networks

Muktikanta Sa alicemukti@gmail.com
Department of Computer Science and Engineering
National Institute of Technology Warangal
AP, India-506004,

Manas Ranjan Nayak manas2nayak@Yahoo.co.in
Department of Computer Science and Application
Regional College of management Bhubaneswar
Orissa, India-751031

Amiya Kumar Rath amiyaamiya@rediffmail.com
Department of Computer Science and Application
College of Engg. Bhubaneswar
Bhubaneswar,Orissa,India-751031

Abstract

Wireless Sensor networks (WSN) is a promising technology for current as well as future. There is vast use of WSN in different fields like military surveillance and target tracking, traffic management, weather forecasting, habitat monitoring, designing smart home, structural and seismic monitoring, etc. For success application of ubiquitous WSN it is important to maintain the basic security, both from external and internal attacks else entire network may collapse. Maintaining security in WSN network is not a simple job just like securing wireless networks because sensor nodes are deployed in randomize manner. Hence major challenges in WSN are security. In this paper we have discussed different attacks in WSN and how these attacks are efficiently detected by using our agent based model. Our model identifies the abnormal event pattern sensor nodes in a largely deployed distributed sensor network under a common anomaly detection framework which will be designed by agent based learning and distributed data mining technique.

Keywords: Wireless Sensor Network, Distributed Data Mining, Machine learning, Anomaly Detection.

1. INTRODUCTION

Wireless sensor networks are developing rapidly in current years and it has vast use in different fields. It is a promising technology in network field. Wireless sensor networks are mainly designed for real time gathering and examination of data in insistent environments. Due to this WSNs are well suitable for [1] military surveillance and target tracking, traffic management, weather forecasting, habitat monitoring, designing smart home, structural and seismic monitoring, etc. WSNs are different from other networks like wired and wireless. In WSN sensor nodes are deployed in open, unsupervised, hostile environment where physical communication is not possible. It operated on an unattended mode area. This leads to a low coherent and physical security level for communication. As a result, the basic communication protocols and algorithms of WSNs have some security problems. So we need stronger algorithm to enhance the security

level. Generally WSN nodes are resource confined in low power embedded processor, memory storage, radio transceiver, sensors, geo positioning system and power source.

For success applications of WSNs it is important to maintain the basic security. Generally security is the level of protection against hazard, harm, defeat, and illegal activity. In the computer science, security refers to a technique which provides guarantee over data stored in a computer or network. And that data cannot be accessed by any others without permission. While communication between nodes we need security over data. In case of WSNs all nodes are independent and they are deployed in randomize manner. So providing security to sensor nodes is not so easy like securing LAN and wireless networks. In this paper we proposed an agent based model which gives more security over data and detect the abnormal events in the network.

The rest of the paper is prepared as follows. In Section 2, we describe the different types of attacks in WSN, categorically represented them in Table 1 and Table 2. In Section 3 we focus on related works so far. In Section 4 we have given our agent based model and architecture of wireless sensor network. We present the experimental result of our proposed model in section 5 and conclude this paper in Section 6.

2. TYPES OF ATTACKS IN WSN

Attacks on WSN can be [2, 3, 5, 11, 12] classified into two main kinds based on interruption of sensor nodes in network: active and passive attacks. In case of passive attack the attacker is outside the network and it watches the communication between client and server [11] and may also passive eavesdropping [12] between them. [5] Whereas in active attack the attacker transmits data to one or both of the nodes, or chunk the data stream in one or both directions in the communication channel. [2] Active attackers can disrupt the normal functionality of the whole network, which means it may change the information, may modify the original data, or can gather falsehood data. The different active attacks in WSN with their behavior are shown in Table1 [2, 6]. The maximum attacks behavior consists of the route updating misbehavior, which sways data transmission between the nodes in the network. Different protocol layer attacks are given in Table 2[1, 2, 4].

Table 1: Different attacks in WSN with their behavior

Attack name	Behaviour and misbehavior
Hello floods	Route updating misbehavior
Node Outage	Route updating misbehavior
Spoofed,	Route updating misbehavior
Sybil	Route updating misbehavior
Sinkhole	Route updating misbehavior
Hello floods	Route updating misbehavior
ACK spoofing	Route updating misbehavior
False Node	Both route updating and data forwarding misbehavior
Message Corruption	Data forwarding misbehavior
Node Malfunction	Data forwarding misbehavior
Denial of Service	Data forwarding misbehavior
Select forward	Data forwarding misbehavior

Table 2: Different protocol layer attacks.

Protocol Layers	Attacks
Application layer	Denial, data bribery
Transport layer	Session hijacking, SYN/ACK flooding
Network layer	Wormhole, flooding, blackhole, Byzantine, resource consumption, location disclosure attacks
Data link layer	Traffic analysis, disruption MAC (802.11), monitoring, WEP weakness
Physical layer	Jamming, interceptions, eavesdropping
Multi-layer attacks	Denial of service, impersonation, replay, man-in-the-middle

3. RELATED WORKS

Security in WSNs is a broad area. As compared to wire and wireless networks, it is a major challenging work. A good discussion about WSNs architecture, applications, key design challenges, sensor network deployment, different localization algorithms, WSN characteristics, medium-access and sleep scheduling algorithms, energy efficiency and robust routing protocols**,** data centric wireless networking, different security mechanism are given by Bhaskar Krishnamachari in [1]. A good summary of present status in sensor network security and research issues is presented by Perrig, J. Stankovic, and D. Wagner al, in [15]. Some of the security concerns include flexible routing, safe communication, and electronic and physical node destructions. Analysis of Sybil attack was given in [19], Newsome et.al, it shows several variants in data aggregation, misbehavior and voting for cluster head. They have given effective security mechanisms against these different attacks for variants. Hu et al. examine the wormhole attack and suggest packet leashes to prevent an attacker from maliciously passageway packets to different areas in a WSN given [20]. In [21], Deng et al. suggest INSENS, intrusion tolerant routing that senses malicious sensors and routes around them. In [16], Karlof and Wagner, review on sensor network routing protocol weakness and defence technique against several electronic attacks. Out of these attacks Sybil attack [18] and the wormhole attack [17] are very harm in nature. In [21] and [22] had discussed about two security protocols, SNEP and µTESLA. These protocols indemnify data discretion, authentication, purity and authenticated broadcast in severely resource constrained background like WSNs. Their model provide defence to sybil, wormhole, eavesdrop attack [23], [15], spoof, respond and message modify attacks [16]. Attackers do traffic examination for determining locations while transmitting messages to the base station is discussed in [24]. In [24], J. Deng et al. have discussed for the protection of the base station from different attacks. Protection from Denial-of-service (DoS) attacks is a key challenge for researchers in WSNs. In [25], Wood and Stankovic study the attacks at different protocol layers in the network [25]. They have designed a time factors constraint which reduces network defencelessness to DoS attacks. In [26], A. D. Wood et al. have discussed about the radio frequency jamming DoS attack and presented a method to route around the jammed area of the network.

4. OUR AGENT BASED MODEL

Figure 1 shows the distributed wireless sensor network architecture. It is a two-tier hierarchical cluster topology [1]. We used this topology for deployment of nodes because it is easy for the multiple nodes of their local region to report to cluster head. Each local region is called a cluster and cluster head is a data gathering node which is discussed later in this section. Another reason for using this topology is that the network deployment becomes attractive in heterogeneous settings when the cluster-head nodes are more powerful in terms of computation and communication. The main advantage of this two-tier hierarchical cluster based approach is that it usually crumbles a large network into separate zones within which data processing and aggregation can be carried out locally. This topology consists of two types of sensor nodes:

(a).Forwarding nodes or simple sensor nodes which sense the activity and forward data to base station.
(b). Cluster head (CH) or simple data gathering point node, where all sensed data from the nodes are collected. As shown in Figure 1, we have four clusters. Each cluster selects a cluster head which is responsible for collection of data from the sensor nodes and send to base station (BS) or sink. CH is not a special node; it is one like other sensor node. A clustering based routing protocol called the base station controlled dynamic clustering protocol[9], which uses a high energy base station to set up cluster heads and achieve other energy rigorous tasks. It can enhance the lifetime of a network. United voting dynamic cluster routing algorithm based on lingering energy in wireless sensor networks [8], which periodically selects cluster heads according to lingering energy among the nodes located in the incident area.

Our approach is completely based on agent based model for classifier to identify the abnormal event pattern sensor nodes in the respective clusters. This classifier model tackle the security problems related to attacks in a distributed wireless sensor networks. In this model we used new system such as distributed data mining and agents for providing solution against wireless sensor network. Figure 2 depicts how we have embedded our agent base model for classifier in the distributed wireless sensor network. Figure 3 shows internal architecture of our proposed model.

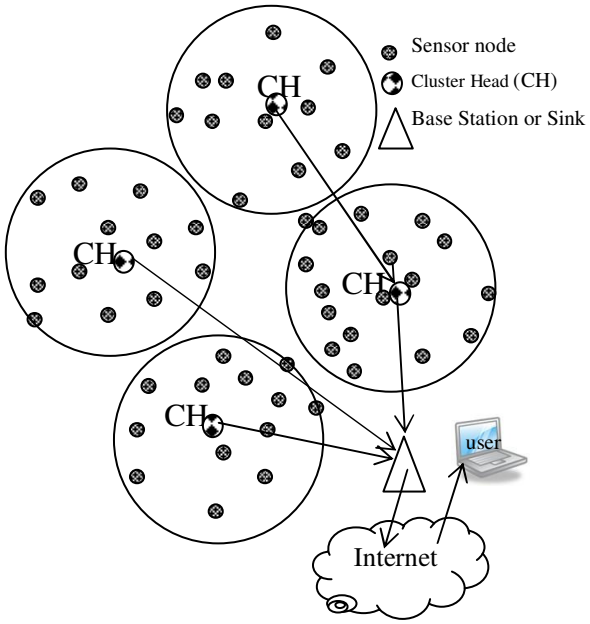


Figure 1: A two-tier hierarchical four cluster based distributed wireless sensor network architecture. It consists of sensor nodes and cluster heads. All cluster heads gather the sense data send to the sink or base station..

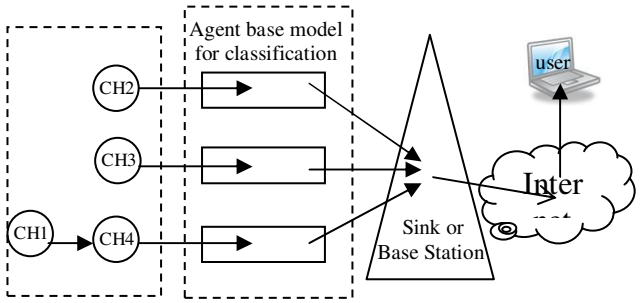


FIGURE 2: Embedded position of our agent base model for classification in the distributed wireless sensor network.

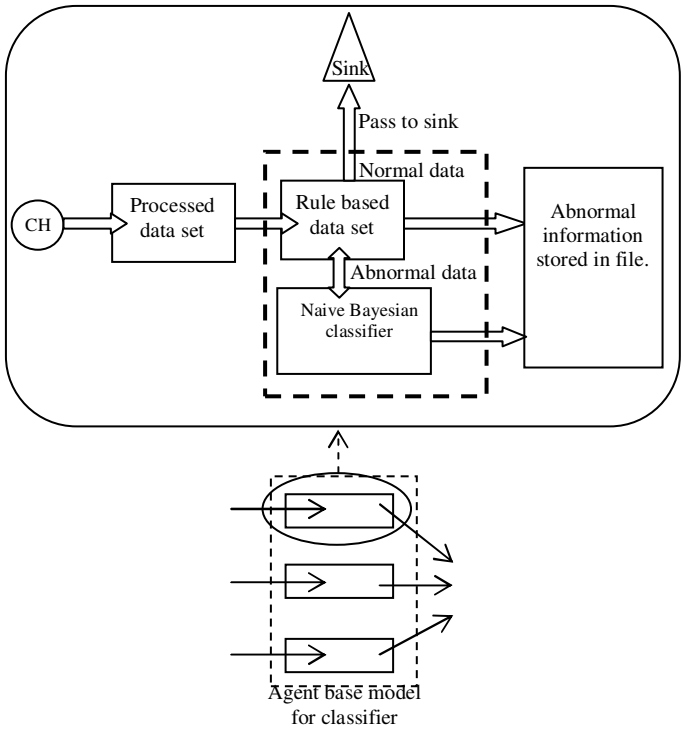


FIGURE 3: Internal architecture of our proposed model. Here we have given the architecture for one CH, similarly all CHs have their own model.

In wireless sensor network, initially, sensor nodes sense the action, and then report to their corresponding cluster head. All information is processed at cluster heads. Then, cluster heads send sensed data file to base station. While gathering data file at cluster head, it may collect some erroneous data, or it is possible that some sensor nodes may send wrong information to CH. These data is called anomaly. Before sending data file to base station cluster heads need to detect anomalies and remove them. In the data file, we will detect the abnormal event information. For that we have embedded our agent base classifier model in-between cluster head and base station. Cluster heads ensures all anomalies present in the data to be removed before it sends to base station.

Each cluster head have its classifier model for training the data. At first our model takes the information of all sensor nodes (for a cluster) in a processed file. The file is then processed using agent based rule and naive Bayesian classifier model. If all processed trained data and processed test data are normal then it will pass the file to base station otherwise the data is

abnormal and the file is not to be sent to the base station. We then apply naive Bayesian classifier to find the anomalies and rectify them to become normal data to be sent to base station. In each time the number of anomalies detected and stored in a file. This process will continue until all anomalies are detected. At the end we calculate the percentage of abnormal events detected and subsequently the percentage of false positive. Again if an abnormal node sends any erroneous data to CH, our proposed model calls the classifier construction to find out the abnormal nodes. If an abnormal node is detected, it will filter the individual node from the global networks. Algorithm 1 illustrates how our proposed agent based model works.

Algorithm 1
Input: Processed n data set files.
Output: percentage of abnormal data stored in a file

```
1: abnormalEventDetectionUsingBayesian()
2: {
3:  for( each file )
4:   {
5: read processed data set file for each cluster head;
6: call naive Bayesian classifier program for training the classifier for abnormal event detection,
   store that information into a test data file;
7: Test this file with the classifier model and write them to an output file;
8: Calculate the percentage of abnormal data;
9:  }
10:}
```

Analysis. Algorithm 1 takes input as n number of processed data set files. Each file is independent for each cluster head. So each file need to be processed which is specified in line 5. After reading a file we will call the naive Bayesian classifier program to train the classifier for detection of abnormal event, after this the result stored in a test file, which is specified in line 6. The output from the line 6 is tested with the classifier model and the output is written to a file, which is specified in line 7. At the end we calculate the percentage of abnormal event which is calculated as

Percentage of abnormal event = (total number of abnormal event $\times$ 100)/ (total number of traces data).

Normal data set is created using the threshold value and a decision threshold value 0 is learned from the training data set. If the probability of abnormal event is greater than threshold value it is labeled as normal data set, otherwise it is labeled as abnormal. Therefore using this agent based model we can able to detect an abnormal event pattern in a distributed WSN.

5. EXPERIMENTAL RESULTS

Our simulation was based on the sensor network running NS2 (version 2.33). We used 200 sensor nodes, four clusters. Each cluster head was elected using united voting dynamic cluster routing algorithm based on lingering energy in wireless sensor networks [8]. All sensor nodes are constant bit rate transport protocol; we used Ad hoc On-Demand Distance Vector (AODV) as routing protocol. The movement of all sensor nodes was randomly generated oven a 1000m $\times$1000m field, with a maximum speed of 75m/s and an average pause of 10ms. Each simulation runs for a time period of 10,000 simulation seconds.

We run this simulation for many times and detected different commonly attacks. We have successfully detected maximum abnormal events. Using this model we calculate the percentage of abnormal events. The experimental result was shown in Table 3 and 4.

Table 3 : Detection abnormal events.

Types of attacks	Percentage(%) of detection rate for			
	CH2	CH3	CH4	Avg
Hello floods	99.12	99.23	98.14	98.83
Node Outage	98.32	99.42	98.21	98.65
Sybil	99.22	98.23	98.19	98.55
Sinkhole	99.12	99.56	99.01	99.23
Hello floods	98.34	98.75	98.76	98.62
False Node	99.15	97.22	97.87	98.08

Table 4: Detection of false positive rate

Types of attacks	Percentage(%) of false positive for			
	CH2	CH3	CH4	Avg
Hello floods	0.34	0.53	0.74	0.54
Node Outage	1.12	0.32	0.65	0.7
Sybil	0.23	0.64	0.84	0.57
Sinkhole	0.45	0.23	0.65	0.44
Hello floods	1.54	0.72	0.65	0.97
False Node	0.33	0.24	1.82	0.79

This system was tested with large number of attacks present in a highly deployed wireless sensor networks. It shows the good results to support the proposed system.

6. CONSLUSION & FUTURE WORK

In this paper, we evaluate the performance of the agent based abnormal detection model, which is implemented by rule base and naive Bayesian technique. Throughout experiment the simulation results shows the performance of our proposed agent based model. The average detection rate of the wireless network is 98.66% and the average false positive rate is 0.67%. Hence the accuracy what we achieve was high and it was much better than the result obtained [4] R. Nakkeeran et al. in adhoc network. Hence this is a well approached model for detection of abnormal events. While doing experiment we found that individual detection rate is very small when the training sample is not substantial. So to achieve high accuracy rate we apply the classifier to a perfect training set of data with known classifications. Experimental results show that average detection rate is increased and average false positive rate is reduced by using this model. In the future work, we will test how to detect data forwarding misbehavior types of attacks using this model. This model can reconfigure using BPN and SVM classifier algorithms.

7. REFERENCES

[1] Bhaskar Krishnamachari “Networking Wireless Sensors”. published in the USA by Cambridge Unversity Press, New York in 2005.

[2] T. G. LUPU “Main Types of Attacks in Wireless Sensor Networks” International Conference in "Recent Advances in Signals and Systems". in 2009, ISSN: 1790-5109, ISBN: 978-960-474-114-4.

[3] Xiaojiang Du Hsiao-Hwa Chen North Dakota State Univ., Fargo, "Security in wireless sensor networks". Wireless Communications, IEEE Publication Date: Aug. 2008 Volume: 15 , Issue: 4 On page(s): 60 – 66.

[4] R. Nakkeeran, T. Aruldoss Albert and R.Ezumalai "Agent Based Efficient Anomaly Intrusion Detection System in Adhoc networks". IACSIT International Journal of Engineering and Technology Vol. 2, No.1, February, 2010, ISSN: 1793-8236.

[5] Dr. G. Padmavathi and Mrs. D. Shanmugapriya, "A Survey of Attacks, Security Mechanisms and Challenges in Wireless Sensor Networks". (IJCSIS) International Journal of Computer Science and Information Security, Vol. 4, No. 1 & 2, 2009.

[6] K.Q. Yan, S.C. Wang, C.W. Liu, "A Hybrid Intrusion Detection System of Cluster-based Wireless Sensor Networks". Proceedings of the International MultiConference of Engineers and Computer Scientists 2009 Vol I, IMECS 2009, March 18 - 20, 2009, Hong Kong.

[7] P C Kishore Raja , Dr.M.Suganthi.M, R.Sunder, "Wireless node behavior based intrusion detection using genetic algorithm". Ubiquitous Computing and Communication Journal, June 2010.

[8] Guo Bin,Li Zhe, "United voting dynamic cluster routing algorithm based on residual-energy in wireless sensor networks". Journal of Electronics & Information Technology. 2007,29(12).pp:3006-3010.

[9] Muruganathan S D, Ma DCF, Bhasin PI, et al. "A centralized energy-efficient routing protocol for wireless sensor networks". IEEE Communications Magazine, 2005,43(3): 8 – 13.

[10] Sooyeon Shin, Taekyoung Kwon, Member, IEEE, Gil-Yong Jo, Youngman Park, and Haekyu Rhy, "An Experimental Study of Hierarchical Intrusion Detection for Wireless Industrial Sensor Networks". IEEE Transactions on Industrial Informatics, Vol. 6, No. 4, Nov. 2010.

[11] Snakar, K. Sundaraliga, S. Nalinsky, A. and Miller, D.(2005) "Cisco wireless LAN security". Expert guidenace for securing year 802:11 networks Cisco Press:U.S.A.

[12] Mohteshim Hussain, "passive and active attcaks against wireless lan's". Unversity of Hertfordshire, England,U.K.

[13] Sutharshan Rajasegarar, Christopher Leckie, James C. Bezdek and Marimuthu Palaniswami, "Centered Hyperspherical and Hyperellipsoidal One-Class Support Vector Machines for Anomaly Detection in Sensor Networks". IEEE Transactions on Information Forensics and Security. Vol. 5, No. 3, Sept. 2010.

[14] Masud Moshtaghi, TimothyC.Havens, JamesC.Bezdek, LaurencePark, ChristopherLeckie, Sutharshan Rajasegarar, JamesM.Keller, Marimuthu Palaniswami", "Clustering ellipses for anomaly detection". Pattern Recognition 44,page 55–69,July 2010.

[15] A. Perrig, J. Stankovic, and D. Wagner, "Security in wireless sensor networks," in Communications of the ACM, Vol. 47, No. 6, June 2004, pp. 53–75.

[16] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: Attacks and countermeasures," in Proc. of 1st IEEE International Workshop on Sensor Network Protocols and Applications, May 2003.

[17] Y. C. Hu, A. Perrig, and D. B. Johnson, "Wormhole detection in wireless ad hoc networks," in Tech. Rep. TR01-384, Department of Computer Science, Rice University, June 2002.

[18] J. R. Douceur, "The sybil attack," in Proc. of 1st International Workshop on Peer-to-Peer Systems, March 2002.

[19] J. Newsome, E. Shi, D. Song, and A. Perrig, "The sybil attack in sensor networks: Analysis and defenses," in Proc. of 3rd International Symposium on Information Processing in Sensor Networks, April 2004.

[20] Y. C. Hu, A. Perrig, and D. B. Johnson, "Packet leashes: A defense against wormhole attacks in wireless ad hod networks," in Proc. of 21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), April 2003.

[21] J. Deng, R. Han, and S. Mishra, "Insens: Intrusion-tolerant routing in wireless sensor networks," in University of Colorado, Department of Computer Science Technical Report CU-CS-9393-02, 2002.

[22] A. Perrig, R. Szewczyk, J. D. Tygar, V. Wen, and D. E. Culler, "Spins: Security protocols for sensor networks," in Proc. of 7th Annual International Conference on Mobile Computing and Networking (MobiCom), July 2001.

[23] W. Du, J. Deng, Y. S. Han, S. Chen, and P. K. Varshney, "A key management scheme for wireless sensor networks using deployment knowledge," in Proc. of INFOCOM, March 2004.

[24] J. Deng, R. Han, and S. Mishra, "Intrusion tolerance and anti-traffic analysis strategies for wireless sensor networks," in Proc. of the 2004 IEEE International Conference on Dependable Systems and Networks (DSN), June 2004.

[25] A. D. Wood and J. A. Stankovic, "Denial of service in sensor networks," in IEEE Computer, October 2002, pp. 54–62.

[26] A. D. Wood, J. A. Stankovic, and S. H. Son, "Jam: A jammed-area mapping service for sensor networks," in Communications of the ACM, Vol. 47, No. 6, June 2004, pp. 53–75.

A New System for Clustering and Classification of Intrusion Detection System Alerts Using Self-Organizing Maps

Amir Azimi Alasti Ahrabi amir.azimi.alasti@gmail.com
Department of Computer
Islamic Azad University, Shabestar Branch
Tabriz, East Azerbaijan, Iran

Ahmad Habibizad Navin ah_habibi@iaut.ac.ir
Department of Computer
Islamic Azad University, Science and Research Branch
Tabriz, East Azerbaijan, Iran

Hadi Bahrbegi hadi.bahrbegi@gmail.com
Department of Computer
Islamic Azad University, Shabestar Branch
Tabriz, East Azerbaijan, Iran

Mir Kamal Mirnia mirnia-kam@tabrizu.ac.ir
Department of Computer
Islamic Azad University, Science and Research Branch
Tabriz, East Azerbaijan, Iran

Mehdi Bahrbegi m.bahribayli@gmail.com
Department of Computer
Islamic Azad University, Shabestar Branch
Tabriz, East Azerbaijan, Iran

Elnaz Safarzadeh elnaz_safarzadeh@yahoo.com
Department of Computer
Islamic Azad University, Shabestar Branch
Tabriz, East Azerbaijan, Iran

Ali Ebrahimi ali.ebrahimi1781@gmail.com
Department of Computer
Islamic Azad University, Shabestar Branch
Tabriz, East Azerbaijan, Iran

Abstract

Intrusion Detection Systems (IDS) allow to protect systems used by organizations against threats that emerges network connectivity by increasing. The main drawbacks of IDS are the number of alerts generated and failing. By using Self-Organizing Map (SOM), a system is proposed to be able to classify IDS alerts and to reduce false positives alerts. Also some alert filtering and cluster merging algorithm are introduce to improve the accuracy of the proposed system. By the experimental results on DARPA KDD cup 98 the system is able to cluster and classify alerts and causes reducing false positive alerts considerably.

Keywords: IDS, alert clustering, SOM, false positive alert reduction, alert classification.

1. INTRODUCTION

An IDS is a device that monitors system and/or network activities for malicious activities or policy violations and produces alerts to a Management Station. IDSs are generally divided into two categories based on detection method: Signature based IDSs and anomaly based IDSs [1]. In IDS there are two major problems namely generating many alerts and high rate of false positive alerts. Alert management methods are used to manage with these problems. One of the methods of alert management is clustering the alerts. According to recent researches, clustering the alerts is an NP-Complete problem [21].

In this paper by using SOM the proposed system classifies and clusters the alerts and also detects false positive alerts. Two algorithms are used in this system to filter alerts to train the SOM better and to merge generated clusters to reduce the number of clusters depending on the types of the attacks. Moreover to obtain a better result from SOM a preprocessing process is applied to the alerts during train and test phases.

The subject is introduced briefly in Section 1. Section 2reviews related works, section 3 explains the suggested system for classifying and clustering the alerts, the experimental results are shown in section 4 and finally section 5 is a conclusion and feature works.

2. RELATED WORKS

K. Julisch [2] proposed a clustering technique witch based on forming a generalized view of false alerts. This technique is for discovering root causes of false positive alerts. Julisch notice that a small number of root causes implies 90% of alerts. by identifying and removing this root causes total number of alerts come down to 82%.

Cuppens in his Mirador project used an expert system for clustering. By this method the alerts are entered to database with XML format and then to decide whether these be merged into a cluster the expert system algorithm is used [4, 5]. Jianxin Wang, et al, have used genetic algorithm for clustering alerts from IDS [10]. Also two clustering algorithms, based on GA and IGA are compared together [11]. Wang applied GA and IGA instead of Julisch's algorithm for "root cause" clustering. To distinguish malicious traffic from normal traffic the SOM is used [7, 8]. Also it has been proved [7, 8] that SOM-based IDS can handle two situations by discovering new attacks. Hayoung, et al. used SOM for real time detection of attacks in IDSs [9].

Maheyzah Md Siraj, et al. compared the clustering algorithms, EM, SOM, K-means and FCM on Darpa 2000 dataset [3]. The results show that EM algorithm is the best for clustering. Since the alerts received by SOM are not filtered thus the result for the SOM could be in doubt.

The main features of the proposed system are obtaining results with high accuracy which is due to filtering alerts and merging the generated clusters and also to reduce the number of false positive alerts considerably.

3. CLUSTERING AND CLUSSIFICATION SYSTEM

The proposed system is shows in Figure 1. schematically. The system depends on produced alerts directly by IDS. To generate alerts, Snort tool [12] with Darpa 98 dataset [13] is used. Snort is an open source signature based IDS which gets Darpa 98 online traffic and then generates alert log files. The alert log files are used as the inputs of the system.

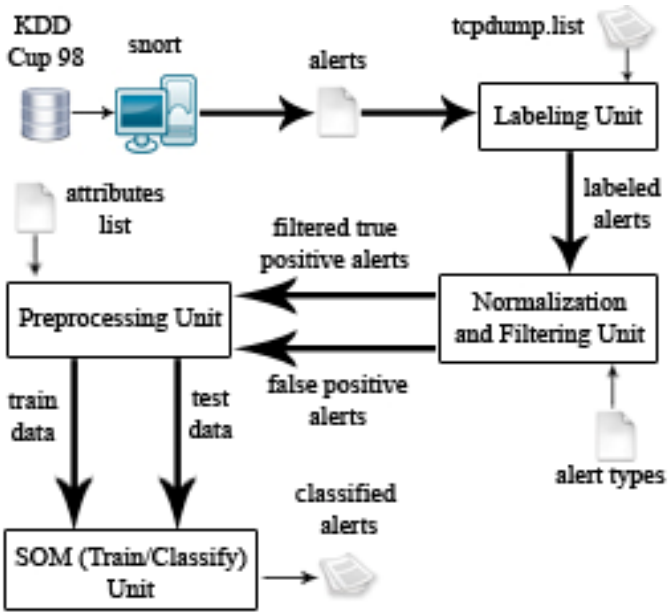


FIGURE 1: Alert clustering and classification system.

2.1 Labeling Unit

In Darpa 98 dataset, there are some tcpdump.list files for each online traffic flow which contains information of attacks. Labeling unit gets the alert and tcpdump.list files returning a list of labeled alerts involving type of attack for each alert.

An algorithm is proposed to generate map between alerts and attacks by using a unique key. This key consists source IP, destination IP, source port, destination port and ICMP code/type. The algorithm is shown in Figure 2.

1. Input TCPDUMP list files.
2. Input alert log files.
3. Create an empty AttackList set.
4. Create an empty AlertList set.
5. For each row in TCPDUMP list files:
   5.1. If the row is a labeled attack then add the row to the AttackList set.
6. For each row in alert log files:
   6.1.Create key with the five attributes: source ip, destination ip, source port, destination port, ICMP code/type.
   6.2. If the key exists in AttackList set then label the selected row with the type of found attack from AttackList set.
   Else
   Label the selected row with the False Positive attack type.
   6.3. Add the selected row to the AlertList set.
8. Return the AlertList set.

FIGURE 2: The algorithm of alerts labeling.

2.2 Normalization and Filtering Unit

Since Snort is a signature based IDS, it can't detect some of attacks like Pod and Smurf. It means that among the available attack type in Darpa 98 dataset, it can detect only eight cases with high accuracy [22]. So this unit takes the list of acceptable attacks, selected attributes and labeled alerts and then produces the list of filtered false and true positive alerts (Figure 1.). In normalization process eight attributes are chosen among the collection of alert attributes [14] stored in a vector named alert vector. The chosen attributes are: Signature ID, Signature Rev,

Source IP, Destination IP, Source Port, Destination Port, Datagram length and Protocol. One of the similar alerts based on values of the alert vector is selected in filtering process. Experiment shows that filtering the similar alerts wouldn't remove two alerts with two different types of attack.

2.3 Preprocessing Unit

Some attributes in the alert vector are string type and some numerical type. In this unit the string values are converted into numerical values and the range of the whole attributes is reduced. This unit takes the list of false positive and filtered true positive alerts and produces train and test data. (Figure 1)

By using (1) and (2) the string values are converted into numerical values.

$$IP = X_1.X_2.X_3.X_4, \qquad (1)$$
$$IP\_VAL = (((X_1 \times 255) + X_2) \times 255 + X_3) \times 255 + X_4$$

$$protocol\_val = \begin{cases} 0, protocol = None \\ 4, protocol = ICMP \\ 10, protocol = TCP \\ 17, protocol = UDP \end{cases} \qquad (2)$$

Since the differences in the range of the values will lose the accuracy of result, so the values of alert vector should be normalized. By using (3, 4), we convert [X_{min}, X_{max}] into [0, 1] (Unit Range) and into [0.1, 0.9] (Improved Unit Range).

$$UR = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \qquad (3)$$

$$IUR = 0.8 \times \frac{x - x_{\min}}{x_{\max} - x_{\min}} + 0.1 \qquad (4)$$

2.4 SOM (Train/Classify) unit

- **SOM**

Self-organizing map [6] is a type of artificial neural networks that is trained using unsupervised learning. SOM describes a mapping from a higher dimensional input space to a lower dimensional map space. When a training example is fed to the network, its Euclidean distance to all weight vectors are computed (5). The neuron with weight vector which is similar to the input mostly is called the best matching unit (BMU). The weights of the BMU and neurons close to it in the SOM lattice are adjusted toward the input vector. The magnitude of the change decreases with time and with distance from the BMU.

$$\|x - m\|^2 = \sum_{k \in K} w_k |x_k - m_k|^2 \qquad (5)$$

where K is the set of known (not missing) variables of sample vector **x**, x_k and m_k are *kth* components of the sample and prototype vectors and w_k is the *k*th weight value. In this paper, the neighbour function

$$\exp\left(-\frac{\|r_c - r_i\|^2}{2\sigma^2(t)}\right) \qquad (6)$$

is Gaussian, where r_c is the location of unit c on the map grid and the $\sigma(t)$ is the neighbourhood radius at time t. And the learning algorithm

$$m_i(t+1) = \frac{\sum_{j=1}^{n} h_{ic(j)}(t) x_j}{\sum_{j=1}^{n} h_{ic(j)}(t)}, \quad (7)$$

is batch, where *c(j)* is the BMU of sample vector $\mathbf{x}_j$, $h_{i,c(j)}$ the neighbourhood function (the weighting factor), and *n* is the number of sample vectors.

- **Training the SOM**

Test data and train data are used as the input for this unit. For each feature, SOM makes the corresponding maps and then construct U-matrix (unified matrix) based on all feature maps [9]. U-matrix method allows to get a more suitable information of the vector distribution. This method is capable to classify all artificially generated data correctly [16]. The algorithm of SOM [17, 18, 19] is described in Figure 3 and can build U-matrix for normalized filtered alerts data in Figure 4. In U-matrix the lighter color neurons mean the borders of clusters.

Since feature maps and U-matrix obtained through two normalization methods (UR and IUR) are similar (the only difference is the range of corresponding feature maps and U-matrixes), thus diagrams of UR are shown.

1. Initialize the network.
   For each node *i* set the initial weight vector $w_i(0)$ to be random.
   Set the initial neighborhood $N_i(0)$ to be large.
2. Present the input.
   Present *x(t)*, the input pattern vector *x* at time *t* (*0<t<n* where *n* is the number of iterations defined by the user) to all nodes in the network simultaneously.
   x may be chosen at random or cyclically from the training data set.
   3. Calculate the winning node.
   Calculate node *c* with smallest distance between the weight vector and the input vector
   $$\|x(t) - w_c(t)\| = \min_i \{\|x(t) - w_i(t)\|\}$$

Hence

$$c = \arg\min \{\|x(t) - w_i(t)\|\}$$

4. Update the weights.
   Update weights for *c* and nodes with neighborhoods $N_c(t)$.
   $$m_i(t+1) = \frac{\sum_{j=1}^{n} h_{ic(j)}(t) x_j}{\sum_{j=1}^{n} h_{ic(j)}(t)},$$
   Where *c(j)* is the BMU of sample vector $\mathbf{x}_j$, *n* is the number of sample vectors and $h_{i,c(j)}$ the neighborhood function (the weighting factor)
   $$h_{ci}(t) = \alpha(t).\exp(-\frac{\|r_c - r_i\|^2}{2\sigma^2(t)})$$
5. Present the next input.
   Decrease h_{ci} so that $h_{ci}(t+1) < h_{ci}(t)$
   Reduce the neighborhood set so that $N_i(t+1) \subset N_{ci}(t)\ \forall_i$
   Repeat from step 2 choosing a new unique input vector $x(t+1) \neq x(j)$, $j \leq t$
   Until all iterations have been made ($t=n$).

FIGURE 3: SOM Algorithm to Construct Maps quoted from [17, 18, 19].

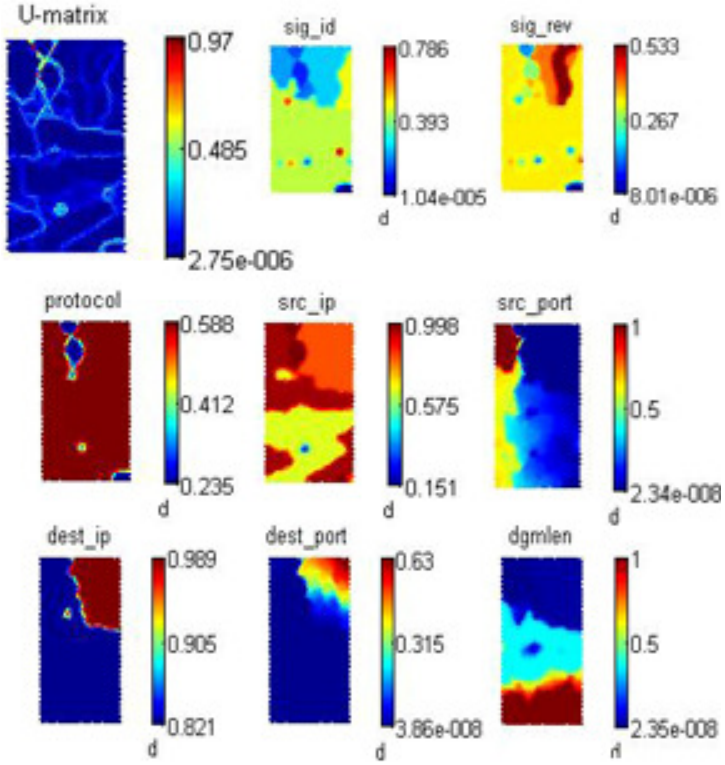


FIGURE 4: U-matrix and 8 Feature Maps.

- **Clustering of U-matrix**

For clustering of U-matrix, distance based U-matrix algorithm with 3 neuron neighbourhoods has been used [20] (Figure. 6. b). To label the clusters, the number of corresponding labels is calculated for each data vector in the cluster and the label with the higher density is supposed to be the label of the cluster. Because there is too many clusters, clusters merging algorithm is executed over clusters (Figure 5.). In this algorithm, the clusters with blank label are supposed to be unknown. Unknown clusters are the borders of various parts of U-matrix. Figure 6. c. shows merged clusters after execution of algorithm.

1. Input clustered U-matrix, *U*.
2. Creating of the initial clusters.
   Let *C* an empty set.
   For each attack type in *U*:
   Add $C_{AttackType}$ set to *C*.
3. Finding correspond clusters for each attack type in *U*.
   For each cluster in *U*:
   Let *SomeClusterOfU* = select a cluster from *U*.
   AttackType= find the name of the *SomeClusterOfU*.
   If *AttackType* is empty then *AttackType* = unknown.
   Add *SomeClusterOfU* to $C_{AttackType}$.
4. Merging of the clusters.
   For each cluster in *C*:
   Merge all of the clusters in *U* correspond to the $C_{AttackType}$ set.

FIGURE 5: The cluster merging algorithm.

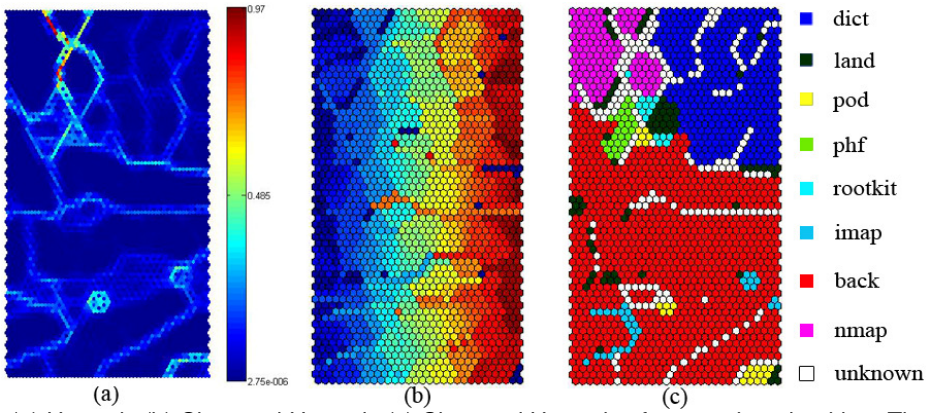


FIGURE 6: (a) U-matrix (b) Clustered U-matrix (c) Clustered U-matrix after merging algorithm. The number of clusters in (b) was 253 and after merging algorithm in (c) reduced to 9 clusters.

- **Classifyin**
- **g of Test Data**

Test data is given to SOM in this unit. It is expected that all the given data vectors from alerts with attack label are placed in the corresponding clusters. Data vectors from false positive alerts should be placed in unknown cluster, otherwise network error should be high. This error shows the distance between entered data vectors and found BMUs. If the error is more than threshold α (α is a constant value) the corresponding data vector is supposed to be as a false positive alert and classified as an unknown cluster.

4. EXPERIMENTAL RESULTS

Matlab software is used to implement the system and SOM toolbox is used to simulate SOM. The map size is 30*60 with grid topology and neighborhood is hexagon. Training data contains 6053 data vectors or 70% of total filtered alert data vectors. In training phase only the data vectors of true positive alerts are used. Since the SOM should be trained with the data vectors extracted from true positive alerts, the data vectors of the false positive alerts are ignored in the training phase. Test data includes 30% of the data vectors of labeled alerts; it means 2591 data vectors, and 2591 data vectors of false positive alerts. The reason for adding the false positive alerts to the test dataset is that always IDSs produce this type of alerts alongside true positive alerts. The number of clusters is 9. Here we let α= 0.1 as a threshold value.

To evaluate the performance of the system, 8 criteria were used. (Table 1)

1-Classification Error (ClaE) is the number of alerts that are wrongly classified. 2-Classification Error Rate (ClaER) is the percentage of wrongly classified alerts (8). 3-Classification Accuracy Rate (ClaAR) is percentage of alerts that are accurately classified as they should be (9). 4-Clustering Error (CluE) is the number of alerts from training data that are wrongly clustered. 5-Clustering Error Rate (CluER) is the percentage of wrongly clustered alerts from train data (10). 6-False Positive Reduction Rate (FPRR) is percentage of false positive alerts that accurately identified and reduced (11). 7-Average Network Hit Error (ANHE) is the average of BUMs error in SOM for all test data (12). 8-Average Network Hit Error for True Positive (ANHETP) is the average of BUMs error in SOM only with the true positive alerts (13).

ClaER=(ClaE ÷Total Number of Alerts Observed)×100 (8)
ClaAR=100–ClaER (9)
ClaER=(CluE÷Total Number of Alerts Observed From Train Data)×100 (10)
FPRR=100–(The Number of False Positive Alerts that Accurately Identified ÷ Total Number of False Positive Alerts Observed)×100 (11)
ANHE=Sum of All BMUs Error÷Total Number of Alerts Observed (12)

ANHETP=Sum of BMUs Error for All True Positive Alerts÷Total Number of Alerts Observed (13)

	ClaE	ClaER	ClaAR	CluE	CluER	FPRR	ANHE	ANHETP
UR	672	12.97	87.03	23	0.38	87.34	0.073	0.011
IUR	33	0.64	99.36	23	0.38	99.71	0.091	0.003

TABLE 1: Proposed system performance metrics.

	ClaE	FPRR	ClaAR
UR	2109	59.70	59.30
IUR	2516	51.87	51.04

TABLE 2: Proposed system performance metrics without considering α threshold value.

As Table 1 shows, the best result is obtained by IUR scaling method which has 99.36% accuracy and its false positive alerts reduction ratio is 99.71%. Because of the IUR method distributes values of attributes of alert vectors as a uniform manner in a mentioned range, the acquired results are better than other unit range method. When Table 1 and Table 2 are compared, you can see the FPRR metric is improved after using α threshold in Table 2. If the result of classification without using α threshold is accepted then gather may exist some incorrectly classified false positive alert vectors. If α threshold is used as a maximum value of network error and classified results have upper value of this threshold value then classification operation is corrected for these alert vectors. Because of dependency between FPRR and ClaAR by improving the result of FPRR metric, the ClaAR metric result is improved.

5. CONSLUSION & FUTURE WORK

In this paper a SOM based system is presented which is able to cluster and classify the alerts with high accuracy. This system is also able to reduce number of false positive alerts considerably.

If the SOM is trained by alerts for various types of attacks with proper filtering process and preprocessing on the alerts, SOM becomes a suitable tool to classify alerts and reduce the false positive alerts in the alert management systems for IDS.
In using SOM for a wide range of alerts with various types of attacks, ratio of error may be high thus using hierarchal SOMs on each trained SOM of a special type of attack can be a solution of this problem.

SOM can be used to find the correlations between alerts. Thus all of the alert vectors with several common attributes placed on one or several neighbour neurons are supposed to be as the related alerts.

6. REFERENCES

1. H. Debar, M. Dacier, and A. Wespi. Towards a taxonomy of intrusion-detection systems. COMPUT. NETWORKS, 31(8):805-822, 1999. 60 Conclusion And Future Work 61.

2. K. Julisch, "Clustering intrusion detection alarms to support root cause analysis", ACM Transactions on Information and System Security (TISSEC) , 2003, Volume 6 , Issue 4, Pages: 443 – 471.

3. Maheyzah, M. S., Mohd Aizaini, M., and Siti Zaiton, M. H. (2009), " Intelligent Alert Clustering Model for Network Intrusion Analysis.", Int. Jurnal in Advances Soft Computing and Its Applications (IJASCA), Vol. 1, No. 1, July 2009, ISSN 2074-8523. pp. 33 - 48.

4. F. Cuppens. Managing alerts in a multi-intrusion detection environment. Proceedings of the 17th Annual Computer Security Applications Conference, 32, 2001.

5. E. MIRADOR. Mirador: a cooperative approach of IDS. Poster present au me European Symposium on Research in Computer Security (ESORICS). Toulouse, France, octobre, 2000.

6. Kohonen, T, "Self-Organized Maps", Springer series in information. Science Berlin Heidelberg:1997.

7. Kiziloren, Tevfik, "Network traffic classification with Self-Organized Maps", Computer and information sciences, 2007, page(s): 1-5.

8. Pachghare,V. K., "Intrusion Detection System Using Self Organized Maps", Intelligent Agent & Multi-Agent Systems, 2009, page(s): 1-5.

9. Hayoung Oh, Kijoon Chae, "Real-Time Intrusion Detection System Based on Self-Organized Maps and Feature Correlations", Third International Conference on Convergence and Hybrid Information Technology, IEEE, 2008, vol. 2, Pages.1154-1158.

10. Wang, J., Wang, H., Zhao, G. 2006. A GA-based Solution to an NP-hard Problem of Clustering Security Events. IEEE 2093- 2097.

11. Jianxin Wang, Baojiang Cui, " Clustering IDS Alarms with an IGA-based Approach", ICCCAS 2009, pp586-591.

12. Snort: The open source network intrusion detection system. Available: http://www.snort.org/.

13. MIT Lincoln Lab. (1998). DARPA 1998 Intrusion Detection Evaluation Datasets. Available: http://www.ll.mit.edu/mission/communications/ist/corpora/ideval/data/1998data.html

14. Snort Manual, www.snort.org/assets/82/snort_manual.pdf

15. Matlab Software, http://www.mathworks.com.

16. A. Ultsch, H. P. Siemon, " Kohonen's Self Organizing Feature Maps for Exploratory Data Analysis", Proceedings of International Neural Networks Conference (INNC) (1990), pp. 305-308.

17. Binh Viet Nguyen, "Self-Organizing Map for anomaly detection", Available in http://www.cs.umd.edu/~bnguyen/papers/papers.html

18. SOM Toolbox for Matlab, Available in http://www.cis.hut.fi/projects/somtoolbox/.

19. Juha Vesanto, John Himberg, Esa Alhoniemi, and Juha Parhankangas, "SOM Toolbox for Matlab 5", SOM Toolbox Team, Helsinki University of Technology, 2000.

20. Juha Vesanto and Esa Alhoniemi. Clustering of the Self-Organizing Map. IEEE Transactions on Neural Networks, 11(2):586–600, March 2000.

21. Julisch, K.: Mining alarm clusters to improve alarm handling efficiency. In: Proceeding of the 17th Annual Computer Security Applications Conference, New Orleans, pp. 12–21 (2001)

22. S Terry Brugger and Jedidiah Chow, " An Assessment of the DARPA IDS Evaluation Dataset Using Snort", UC Davis Technical Report CSE-2007-1, Davis, CA, 6 January 2007

Enhanced Mobile IP Handover Using Link Layer Information

Mohamed Alnas M.J.R.Alnas@bradford.ac.uk
School of Informatics, Mobile Computing
Network and Security Research Group
University of Bradford
Bradford, UK

Mahmud Mansour mmmansour@ieee.org
College of Engineering
Department of Electrical Engineering
Al-Imam Muhammad Ibn Saud University
Riyadh, Saudi Arabia

Abstract

The main source of the problem in Mobile handover is the latency and packet loss introduced by the lengthy registration processes. The registration messages must traverse all the way to the home agent (HA) and back. In addition, the packets sent by the corresponding node (CNs) are lost until they receive the binding update (BU) indicating the new care-of-address (nCoA) of the mobile node (MN). To reduce the number of lost packets during this time, the MN can request the old access router (oAR) to forward all its incoming packets to the new access router (nAR)
Mobile IP handovers can be improved through link layer information to reduce packet loss during handovers. It avoids link disruption during Mobile IP handovers and reduces packet loss. Therefore, link layer information allows an MN to predict the loss of connectivity more quickly than the L3 advertisement based algorithm. It is the best choice used to predict a breakdown wireless link before the link is broken. This facilitates the execution of the handover and eliminates the time to detect handover.

Keywords- Mobile IP Handover; Link Layer Information; Fast Handover; Handover Latency; Packet Loss

1. NTRODUCTION

The IP is expected to become the main carrier of traffic to mobile and wireless nodes; this includes ordinary data traffic like HTTP, FTP and e-mail, as well as voice, video and other time-sensitive data. The goal is to allow applications on a Mobile IP node to keep on communicating with other nodes while roaming between different IP networks. Roaming typically occurs when the MN physically moves from its location to a new location and decides to make use of a different access link technology; this can result in the node disappearing from one point on the Internet, and, topologically at least, re-appearing at another point.

Mobile IP is an Internet standards protocol, proposed by the Internet Engineering Task Force (IETF), which enhances the existing IP to accommodate mobility [1, 2]. Mobile IP in wireless networks is intended to be a direct extension for existing fixed/wireline networks with uniform end-to-end Quality-of-Service (QoS) guarantees. In addition, using Mobile IP, seamless roaming and access to applications will make users more productive, and they will be able to access applications on the fly, perhaps giving them an edge on the competition.

Mobile IP handover defined as the process for redirecting IP packet flow destined to the MN's old location to the MN's current attachment point. When MN moves to a new subnetwork, packets are not delivered to the MN at the new location until the Care-of-Address (CoA) registration to HA is complete. Mobile IP doesn't buffer packets sent to the MN during handovers. Therefore, these packets may be lost and need to be retransmitted [3, 4].
During the handover procedure, there is a time period in which a MN cannot send or receive packets, because of the link switching delay. This period of time known as handover latency; Moreover; there is a high Mobile IP handover delay because of the agent discovery and registration periods, eventually Mobile IP handover can cause significant performance degradation, especially in large scale mobility environments

Mobile IP use link layer information to force a handover to a new access network before any mobility at the network layer can be detected [2]. In this paper we propose the use of link-layer information, and the link-layer trigger to enhance the overall performance of the enhanced Mobile IP handover.

2. MOBILE IP HANDOVER LATENCY

The handover time can be defined as the time between reception of the last packet through the old FA (oFA) and reception of the first packet through the new FA (nFA). Throughout the time between the MN leaving the old foreign network and HA receiving the MN registration message, HA does not know the MN's latest CoA and, therefore, it still forwards the packets destined for MN to the old foreign network [5, 6, 7]. These packets will be discarded and lost. The packet losses could cause impossible disruptions for real-time services, degrade the QoS and lead to severe performance deteriorations of upper layer protocols, especially when the handover is frequent and the distance between MN and the HA is great [8,9,10].

3. LINK LAYER INFORMATION

Link layer information allows an MN to predict the loss of connectivity more quickly than L3 advertisement-based algorithms. It is used to predict a breakdown wireless link before the link is broken. This facilitates the execution of the handover, and the elimination of the time to detect handover [11,12].

MN monitors any advertisements, records the lifetime and updates the expiration time when a new advertisement is received from a new network. When the advertisement lifetime of the current Mobile IP's FA expires, the MN assumes that it has lost connectivity and attempts to execute a new registration with another FA [5]. Although the MN might already be informed about the availability of the nFA, the mobile agent defers switching until the advertisement lifetime of the oFA is expired [27].. Mobile IP handovers that are based on movement detection being handled by the network layer are not appropriate to provide seamless and lossless connectivity of MNs, such movement detection causes packet loss and detects movements after the previous link has been broken [13.14].

4. RELATED WORKS

There are present two techniques in [15]. to support delay-sensitive and real-time applications in Mobile IPv4: pre-registration and post-registration. The pre-registration is based on link layer triggers, proxy agent advertisements and agent solicitations. In advance of a handover, the neighbouring FAs exchange agent advertisement messages with each other. A link layer trigger at the MN, or the old or the new FA triggers the handover.

Note that this happens before the actual L2 handover. The MN receives a proxy agent advertisement from the nFA relayed by the old foreign agent (oFA). Now, the MN can issue a Registration Request *(ReReq)* to the nFA via the oFA. This allows the MN to use the oFA until the registration completes. If the L2 handover is scheduled to the same moment that the registration completes, the overall handover latency will be reduced only to the L2 handover latency.

The post-registration method forms tunnels between FAs to forward arriving packets to the current location of the MN. Packets from the HA are first received by an anchor FA. The anchor FA is the FA that relayed the last Registration Request/Reply pair to the HA, it forwards packets to the subnet of the FA that currently serves the MN. There is no need for

further registrations until the anchor FA has to be changed. The MN can postpone the registration to a time that it sees the most appropriate. Tunnels between FAs are established by a Handover Request *(HReq)* and a Handover Reply *(HRep),* the messages that are exchanged upon the L2 trigger that indicate the handover [16]..

Development of the link layer hints that are used as an input to the handover decision process [9]. An algorithm for handover initiation and decision has been developed based on the policy-based handover framework introduced by the IETF. A cost function was designed to allow networks to judge handover targets based on a variety of user and network valued metrics. These metrics include link layer hint parameters, as well as other QoS metrics. Evaluation of this method considered only the network controlled side, while mobile control was not mentioned, which in fact makes a difference between both of them.

The proposed a scheme in [17]. involving a Layer 3 handover which is able to reduce packet loss and provide more seamless handover without buffering. The proposed scheme uses L2 triggers and applies the tunnel mechanism and pre-registration method of the low latency handover scheme in Mobile IPv4.

However, the direction of the established tunnel of this scheme is opposite to that of the low latency handover of post-registration, in that data traffic arriving at the nFA is tunnelled to the oFA. When an L2 trigger is issued, the MN sends a seamless fast registration request (*SF_RegReq*) to the nFA, this message should carry the oCoA to create a tunnel between the nFA and the oFA.

Upon receiving *SF_RegReq*, the nFA forwards it to the HA, creates a *HRqst* message and sends it to the oFA to make the oFA is ready to create a tunnel with the nFA. Then, the data traffic for the MN is able to be transferred to the nFA. After receiving a seamless fast registration reply (*SF_RegReply*), which is replied to by the HA, the nFA encapsulates the data traffic received from HA and transmits it to the oFA. Therefore, the MN, which is still connected to the oFA, can receive the data traffic. When an L2 handover to the nFA completes, a link-up trigger is generated at both mobile and network sides. Then the nFA removes the tunnel and starts swapped these round data traffic to the MN [18].

5. PROPOSED ALGORITHM

Link layer information, such as signal strength, is continuously available, providing important information about the accurate link's quality. Therefore, link layer information allows an MN to predict the loss of connectivity more quickly than the L3 advertisement based algorithm. It is the best choice used to predict a breakdown wireless link before the link is broken. This facilitates the execution of the handover and eliminates the time to detect handover.

We propose Enhanced Mobile IP (E-Mobile IP) handover using link layer information, such as signal strength, network prefix, bandwidths and link indicator, which is continuously available, providing important information about the availability of new links [18]. Therefore, E-Mobile IP uses link layer information to allow an MN to predict the loss of connectivity more quickly than network layer advertisement based algorithms. Figure 1 describes the overall E-Mobile IP protocol message flow.

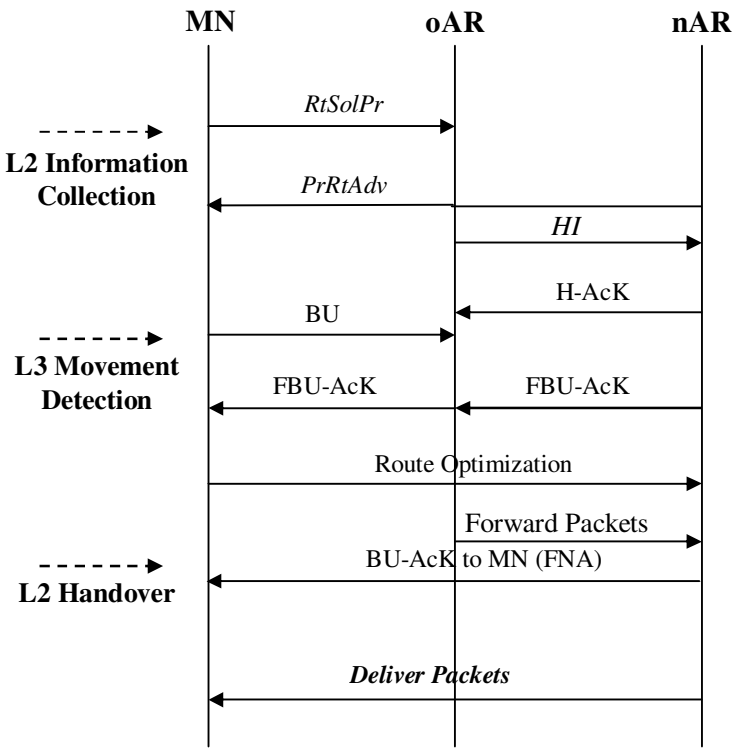


FIGURE 1: E-Mobile IP Protocol Message Flow

The π-Calculus E-Mobile IP handover system is described in terms of the following actions.

5.1 Handover system

The handover system is made up of the MN and access routers as follows:

$$System \overset{def}{\equiv} \text{MN}\langle CoA \rangle \mid \text{oAR} \mid \text{nAR} \mid$$

5.2 Mobile Node (MN)

The MN will receive a link from the nAR, which is used to communicate with it. Then, the MN sends *RtSolPr* to inform the oAR that it is going to handover to the nAR.

$$MN(CoA) \overset{def}{\equiv} \overline{RtSolPr}\langle CoA \rangle . PRtAdv(nCoA, Link\ Information, LinkIdentifier)$$
$$\overline{BU}\langle new \rangle . FBU\text{-}AcK. FNA. MN\langle nCoA \rangle$$

MN will initiate L3 handover by sending an *RtSolPr* message to the oAR, if the L2 trigger is received at the mobile-initiated handover. On the contrary, the oAR will send *PRtAdv* to the MN, if the L2 trigger is received at the network controlled handover [12, 19].
Then; MN checks the neighbour cache to determine the link layer address of the next nodes, a neighbour is considered reachable if it has recently received confirmation that packets sent to the neighbour have been received [16,20].

An MN obtains an nCoA while it is still connected to the oAR; it performs this by receiving the RA included in the visited network information from the nAR.

5.3 Old Access Router (oAR)

The oAR is made up of the following components:
RtSolPr: a process utilized by the MN, sent to its current AR to request information about likely candidate APs and handle the MN initial request for the handover.
Forward: a process which passes both new and old CoAs.
HI: a request message sent to the nAR to make the handover process.

The oAR first receives the handover request from the MN, and then sends it directly to the nAR:

$$oAR \overset{def}{\equiv} \mathrm{RtSolPr}(\mathrm{oCoA}).\overline{\mathrm{Forward}}\langle \mathrm{oCoA}\rangle.PRtAdv(nCoA, Link\ Information, LinkIdentifier).$$
$$\overline{PRtAdv}(nCoA, Link\ Information, LinkIdentifier).\overline{\mathrm{HI}}.\ \mathrm{HAcK}$$
$$BU.\ FBU\text{-}Ack.\ \overline{FBU\text{-}AcK}.\langle\ Forward\ Packets\ \rangle.\ oAR$$

The oAR will validate the nCoA and send a Handover Initiation (HI) message to the nAR to establish the bi-directional tunnel process between oAR and nAR [16].
After the oAR receives the BU, it must verify that the requested handover is accepted as it was indicated in the H-AcK message.

The oAR starts forwarding packets addressed for the oCoA to the nAR and sending a Binding Update Acknowledgement (BU-AcK) with a Fast Neighbour Advertisement (FNA) to the MN.

5.4 New Access Router (nAR)

The nAR is made up of the following components:
Forward: a process which passes both new and old CoAs.
PRtAdv: the response by the present AR, containing the neighbouring router's advertisement for the link information and network prefix.
H-Ack: a confirmation sent back to the oAR to make the handover to the nAR.

$$nAR \overset{def}{\equiv} Forward\ (oCoA).\overline{PRtAdv}\langle nCoA, Link\ Information, LinkIdentifier\rangle.$$
$$HI.\ H\text{-}AcK.\ \overline{FBU\text{-}Ack}\ .\ \langle Forward\ Packets\rangle.\ \overline{FNA}.\ nAR$$

The nAR will respond with the Handover Acknowledgment (H-AcK) message. Then the MN sends a BU to the oAR to update its binding cache with the MN's nCoA.

When MN receives a *PrRtAdv*, it has to send a BU message prior to disconnecting its link.
Upon verification of the variables, nAR will send the Acknowledgment (*ACK*) to confirm its acceptance; then the oAR will start sending the buffered packets to the nAR distend to the MN.

6. SIMULATION SCENARIO and CONFIGURATION

The simulations are carried out using network simulator ns-2 version ns-allinone-2.31, implementations of the E-Mobile IP handovers [21, 22]. The simulator is modified to emulate IEEE 802.11 infra-structured behaviours with multiple disjoint channels. This modification forces L2 handover operations, where stations only receive data packets via one AP at a time. The domain contains eight ARs, each one managing a separate IEEE 802.11 cell.

The network features three MNs connected to it; the first will move sequentially from AR to AR, starting at AR1, performing handovers at a rate of a 30 handovers/min. In each test, the MN1 will be the receiver of a CBR or FTP traffic source, generating either UDP or TCP packets. This traffic originates from the CN1 outside the network, or inside the domain from CN2. All presented results are taken as the average of multiple independent runs, coupled with a 95% confidence interval.
For simplicity we assume that there is no change in direction while the MN moves inside the overlapping area. The best possible handover point occurs at position A, as shown in Figure 2.

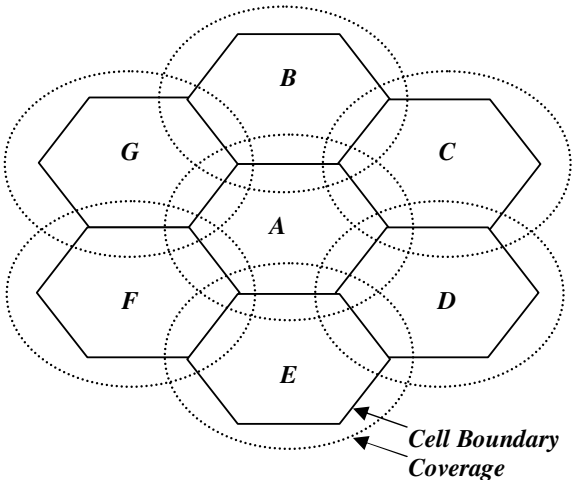


FIGURE 2: Overlapping Coverage Area

The coverage area can be defined in terms of signal strength; the effective coverage is the area in which MNs can establish a link with acceptable signal quality with the AP. The coverage radius is defined as the distance from an AP to its coverage boundary. The cell radius is the distance from an AP to its cell boundary.

7. PERFORMANCE ANALYSIS and EVALUATION

In our simulation, we use a 600m × 600m and a 1000m × 1000m area with a 3 to 7 MNs [5, 11]. The network bandwidth is 2 Mbps and the medium access control (MAC) layer protocol is IEEE 802.11 [23]. The packet size is 10p/s which will generate enough traffic when we increase the number of connections for example at 40 connections of source-destination pairs, it will generate 400 packets per second for whole scenario. Other simulation parameters are shown in Table1. These parameters have been widely used in the literature [24, 25.26].

Simulation parameter	Value
Simulator	Ns-allinone-2.31
Network range	600m×600m and 1000m×1000m
Transmission range	25m
Mobile nodes	3 and 5
Traffic generator	Constant bit rate
Bandwidth	2Mbps
Packet size	512 bytes
Packet rate	10 packet per second
Simulation time	900s and 1200s

TABLE 1: Simulation Parameters

The main purpose behind the proposed approach is to reduce the handover latency and the number of packets loss. As a result, end-to-end delay can also be reduced and the throughput can be improved.

The TCP throughput between MN and CN was measured for the E-Mobile IP, Standard Mobile (S-Mobile IP) and previous study of Mobile IP [24, 25.27]. Some of these results are shown in Figure 3, values are averages over several measurements made on the receiving process for CN to MN down stream traffic.

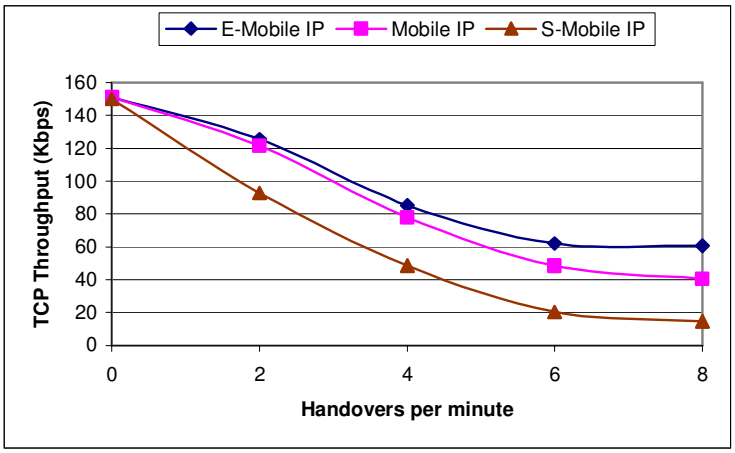


FIGURE 3: TCP Throughput for Data Transfer from CN to MN

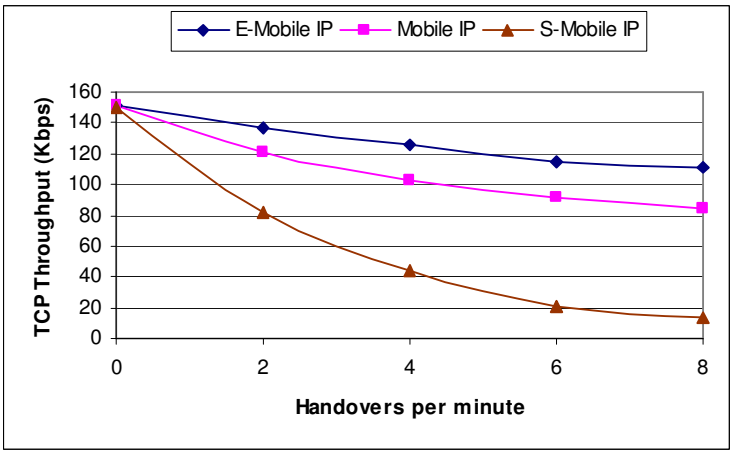


FIGURE 4: TCP Throughput for Data Transfer from MN to CN

Measurements of TCP throughput in Figure 4 are also made for upstream traffic from the MN to the CN. The throughput degradations using Mobile IP with an increasing number of handovers per minute are similar in this case to the previous case. These are not unreasonable values for Mobile IP, where HA and FA could be very far away from each other, the resulting degradation in performance is evident.

However, E-Mobile IP performs better in upstream direction, because, even before the crossover node is aware of the handovers, data packets following the handover message are already taking the right path for transferring packets, because due to the route update (TCP acknowledgments may be lost during this time though, accounting for the slight degradation in throughput as the handover rate increases).

In Figure 5 and Figure 6 the expected number of lost packets is shown as a function of the buffer size at the oFA. Figure 5 shows that the results for link delays are equal to 5ms on every scheme of the simulated Mobile IP, whereas in Figure 6 the three schemes on the nFA path are increased to 10ms each.

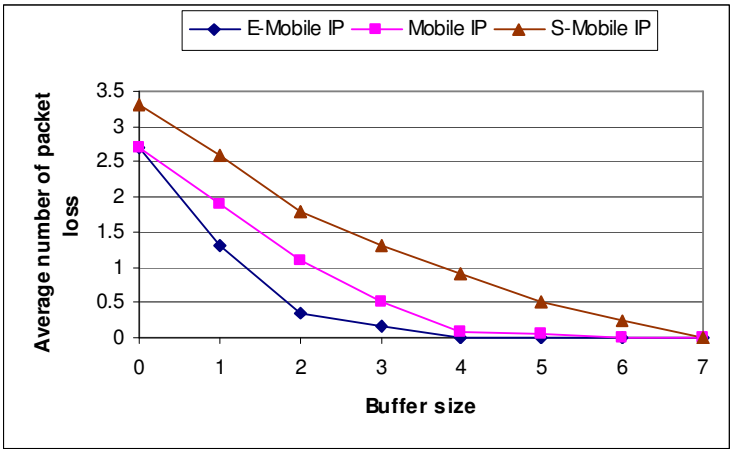


FIGURE 5: Packet Loss as a Function of the Buffer Size

Obviously, the loss in the buffer at the oFA decreases when the buffer size is increased. The packets that are lost in the case of a very small buffer size do go through the buffer when this buffer size increases.

However, it possibly contributes, in the latter case, to the number of lost packets at the nFA. This is especially true for the case of the 10ms-link delay on the nFA-oFA path, because then the whole buffer is likely to arrive too early at the nFA (that is, before the registration reply (*ReRep)* message from the nFA has arrived). In other words, in the case of a long delay on the nFA path, if a packet is not dropped at the oFA, it will most likely be lost at the nFA.

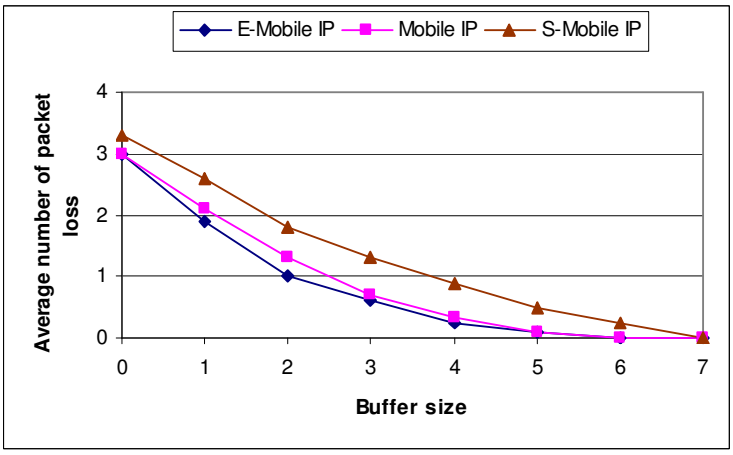


FIGURE 6: Packet Loss as a Function of the Buffer Size

In order to avoid packet loss at the oFA, the dimensions of the forwarding buffer need to be such that it can store packets in the order of the product bit rate of the stream times delay (MN; nFA; oFA). The loss at the nFA, on the other hand, depends on the difference between the distance (nFA; HA) (nFA; oFA). If the latter is smaller than the former, then packets may get lost. A possible solution would be to provide the nFA with a buffer to store temporarily unauthorized traffic until the *ReRep* from the oFA arrives at the nFA.

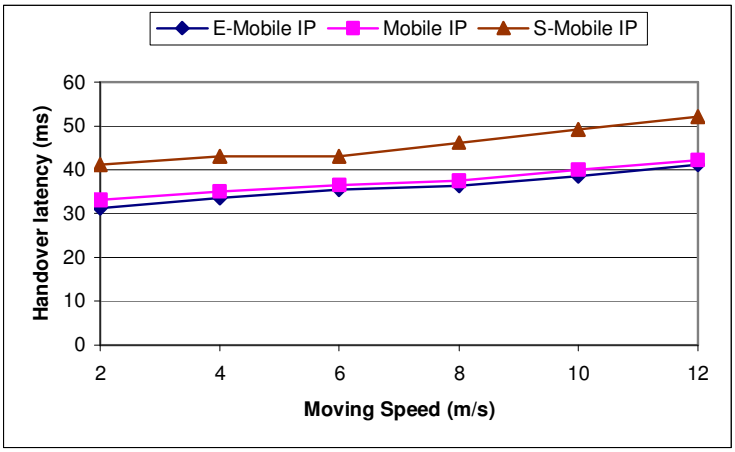


FIGURE 7: Impact of Moving Speed and Layer 2 Beacons

Next we vary the movement speed of MN from 2m/s up to 12m/s, and vary the L2 beacon period from 20ms to 60ms. As shown in Figure 7, when MN's moving speed is less than 5m/s, the impact of moving speed is not obvious. When MN moves faster, the whole Mobile IP scheme will experience higher handover latency due to MN having insufficient time to prepare for the handover. Therefore, there is a higher possibility that the packets are forwarded to the outdated path and are lost. The time instance during which MN can receive packets from a new path will be postponed and the handover latency increases accordingly.

Comparing the curves of different L2 beacon periods in Figure 7, we can see S-Mobile IP generates the highest handover latency at low moving speeds (under 50m/s). This is because of too small a beacon period (for example, 12ms) produces a high volume of beacons. The packet loss rates for the signalling packets thus increase, and require additional retransmission time to deliver them successfully. The handover latency will, therefore, increase. However, at higher speeds (more than 5m/s), the small L2 beacon period can help the MN to detect the nAP and begin the L2 connection setup earlier, thus reducing the possibility that packets are forwarded to the outdated path, resulting in a decrease in the handover latency.

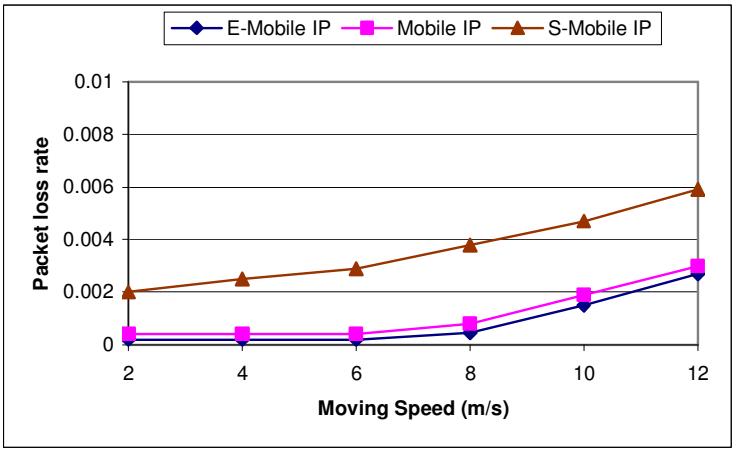


FIGURE 8: Impact of Moving Speed and Beacon Period on Packet Loss Rate

When the MN moves faster than 6m/s, Mobile IP experiences a higher packet loss rate in Figure 8 and decreased throughput in Figure 9 when compared with those of a low moving speed. This is because the possibility of packets being forwarded to an outdated path increases with an increase in the speed

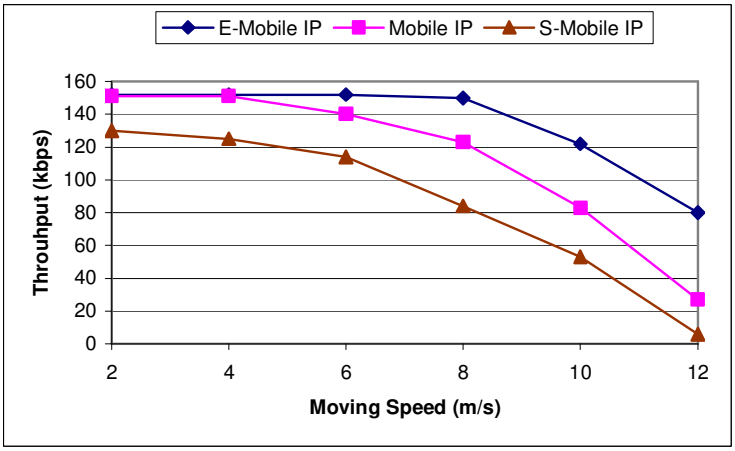


FIGURE 9: Impact of Moving Speed and Beacon Period on Throughput

These packets are dropped by AR1/AR2, either because they are not aware of MN's current location or because the buffer space is full. We can also notice that reducing the L2 beacon period somewhat offsets the impact of high speed by detecting the nAP and beginning the L2 connection setup earlier. Therefore, there will be a smaller probability that the packets are sent to an outdated location and get dropped by the AR.

8. CONCLUSION

In this paper we developed and analyzed the proposed scheme of the E-Mobile IP handover using link layer information scheme, we then compared the experimental results with the results of the Mobile IP and S-Mobile IP. The performance study in this chapter indicates that the use of link layer information with location information helps to minimize packet loss and improve the throughput of Mobile IP handover.

We have seen that the starting point for packet loss could happen in two ways: first, packets may get lost in the oFA when the forwarding buffer overflows and secondly, packets may get lost in the nFA when, upon their arrival, the *ReRep* from the HA has not arrived in the nFA. The first reason for loss may be avoided by appropriately dimensioning the forwarding buffer. This buffer should be able to store arriving packets at least during a time equal to the delay on the nFA and oFA path. The second loss is more difficult to deal with. It is determined by the difference between the delays of the paths oFA, nFA and nFA, HA.

In addition, we evaluated the impact of L2 setup on different performance measures of Mobile IP, together with handover latency, packet loss and throughput. The simulation results show that E-Mobile IP handover latency is not too sensitive to L2 setup latency and beacon periods compared to the other schemes of Mobile IP. Moreover, E-Mobile IP can achieve a fast and seamless handover if MN's moving speed is not too high, but is within reasonable limits.

The reasons for the improved performance of the proposed scheme include the exploitation of location information and the use of the powerful entity RA for complex tasks. In the proposed scheme the powerful RA was used for most of the decision processes necessary for handover. Simulation results in this chapter demonstrate that in most cases the link layer information handover scheme improves the TCP and UDP performances.

REFERENCES

[1] C. E. Perkins, "*Mobile Networking through Mobile IP*", *Internet Computing, IEEE,* vol. 2, pp. 58–69, 1998.

[2] P. Bhagwat, C. Perkins and S. Tripathi, "*Network Layer Mobility: an Architecture and Survey*", *IEEE Personal Communications,* vol. 3, pp. 54–64, 1996.

[3] H. Balakrishnan, V. N. Padmanabhan, S. Seshan and R. H. Katz, "*A Comparison of Mechanisms for Improving TCP Performance over Wireless Links*", *IEEE/ACM Transactions on Networking,* vol. 5, pp. 756–769, 1997.

[4] C. Perkins (ed.), *"IPv4 Mobility Support*", *RFC 2002*, October 1996.

[5] S. Mohanty and I. F. Akyildiz, "*A Cross-layer (Layer 2 + 3) Handoff Management Protocol for Next-Generation Wireless Systems", Transactions on Mobile Computing,* vol. 5, pp. 1347–1360, 2006.

[6] I. F. Akyildiz, X. Jiang and S. Mohanty, "*A Survey of Mobility Management in Next-generation All-IP-Based Wireless Systems*", *IEEE Wireless Communications*, vol. 11, pp. 16–28, 2004.

[7] P. Venkataram, R. Rajavelsamy and S. Laxmaiah, *"A Method of Data Transfer Control during Handoffs in Mobile-IP Based Multimedia Networks", ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 5, no. 2, pp 27–36, April 2001.

[8] I. F. Akyildiz, *"Mobility Management for Next Generation Wireless Systems", Proceedings of the IEEE*, vol. 87, no. 8, pp. 1347–84, August 1999.

[9] M.Mansour, J.Mellor and I.Awan. *"Improving handoff HMIP Scheme Using Location Information"* The 3 International Conference on Information and Communication Technologies: ICCTA' 08, Dmascus, Syria April 2008

[10] M.Mansour, J.Mellor and I.Awan "*Fast handoff Scheme Using Location Information*" The 10th International Conference on Computer Modelling and Simulation, Cambridge, England April 2008

[11] J. Puttonen, *"Using Link Layer Information for Improving Vertical Handovers", 16th International Symposium on Personal, Indoor and Mobile Radio Communications IEEE*, 2005.

[12] H. Chung-Ming, C. Meng-Shu and L. Jin-Wei, "*A Link Layer Assisted Fast Handoff Scheme Using the Alternative Path Approach", 20th International Conference on Advanced Information Networking and Applications*, 2006.

[13] F. Fang and D. S. Reeves, *"Explicit Proactive Handoff with Motion Prediction for Mobile IP"*, *2004 IEEE Wireless Communications and Networking Conference, WCNC 2004*, vol. 2, pp. 855–860, 2004.

[14] S. Oh, H. Song and Y. Kim, *"Seamless Fast Handover in Mobile IPv4 Using Layer-2 Triggers", 2nd International Conference on Systems and Networks Communications, ICSNC 2007*, pp. 16–16. 2007.

[15] S. Thalanany, *"Low Latency Handoffs in Mobile IPv4", draft-ietf-mobileip-lowlatency-handoffs-v4-04.txt*, June 2002.

[16] K. El-Malki and H. Soliman, "*Fast Handoffs in Mobile IPv4", Internet draft, draft-emalki-mobileip-fast-handoffs-03.txt,* September 2000.

[17] S. Oh, H. Song and Y. Kim, "*Seamless Fast Handover in Mobile IPv4 Using Layer-2 Triggers," in Systems and Networks Communications, ICSNC 2007, 2nd International Conference*, pp. 16-16, 2007.

[18] R. Koodli and C. E. Perkins, *"Mobile IPv4 Fast Handovers"*, Internet draft, *Internet Engineering Task Force*, *draft-ietf-mip4-fmip*, February 2006.

[19] R. Hsiehet, *"S-MIP: a Seamless Handoff Architecture for Mobile IP", Proceedings of INFOCOM 2003*, March 2003.

[20] G. Dommety and T. Ye, *"Local and Indirect Registration for Anchoring Handoffs"*, draft-dommety-mobileip-anchorhandoff-01.txt, July 2000.

[21] Columbia University, *Columbia IP Micro-Mobility Software*, http://www.comet.columbia.edu/micromobility/.index.html.

[22] G. Pollini, *"Trends in Handover Design", IEEE Communications Magazine*, 34, 3, 80–90, March 1996.

[23] S. Goswami, *"Simultaneous Handoff of Mobile-IPv4 and 802.11",* Internet Draft, *IETF, draft-goswami-mobileip-simultaneous-handoff-v4- 02.txt*, February 2003

[24] H. Chung-Ming, C. Meng-Shu and L. Jin-Wei, *"A link layer assisted fast handoff scheme using the alternative path approach,"* in *Advanced Information Networking and Applications: 20th International Conference*, pp. 5, 2006.

[25] S. Oh, H. Song and Y. Kim, *"Seamless Fast Handover in Mobile IPv4 Using Layer-2 Triggers,"* in *Systems and Networks Communications, ICSNC 2007, 2nd International Conference*, pp. 16-16, 2007.

[26] S. Aust, D. Proetel, N. A. Fikouras, C. Pampu and C. Görg, *"Policy Based Mobile IP Handoff Decision (POLIMAND) Using Generic Link Layer Information", 5th IEEE International Conference on Mobile and Wireless Communication Networks (MWCN 2003*), Singapore, October 2003.

[27] C. Hsia, *"Low-latency Mobile IP Handover Based on Active-scan Link Layer Assisted",* 2007.

CALL FOR PAPERS

Journal: International Journal of Computer Science and Security (IJCSS)
Volume: 5 **Issue:** 1
ISSN:1985-1553
URL: http://www.cscjournals.org/csc/description.php?JCode=IJCSS

About IJCSS

The International Journal of Computer Science and Security (IJCSS) is a refereed online journal which is a forum for publication of current research in computer science and computer security technologies. It considers any material dealing primarily with the technological aspects of computer science and computer security. The journal is targeted to be read by academics, scholars, advanced students, practitioners, and those seeking an update on current experience and future prospects in relation to all aspects computer science in general but specific to computer security themes. Subjects covered include: access control, computer security, cryptography, communications and data security, databases, electronic commerce, multimedia, bioinformatics, signal processing and image processing etc.

To build its International reputation, we are disseminating the publication information through Google Books, Google Scholar, Directory of Open Access Journals (DOAJ), Open J Gate, ScientificCommons, Docstoc and many more. Our International Editors are working on establishing ISI listing and a good impact factor for IJCSS.

IJCSS List of Topics

The realm of International Journal of Computer Science and Security (IJCSS) extends, but not limited, to the following:

- Authentication and authorization models
- Communications and data security
- Computer Engineering
- Bioinformatics
- Computer Networks
- Computer graphics
- Cryptography
- Computer security
- Databases
- Data mining
- Image processing
- Electronic commerce
- Operating systems
- Object Orientation
- Programming languages
- Parallel and distributed processing
- Signal processing
- Robotics
- Theory
- Software engineering

IMPORTANT DATES

Volume: 5
Issue: 1
Paper Submission: January 31, 2011
Author Notification: March 01, 2011
Issue Publication: March /April 2011

CALL FOR EDITORS/REVIEWERS

CSC Journals is in process of appointing Editorial Board Members for ***International Journal of Computer Science and Security (IJCSS)***. CSC Journals would like to invite interested candidates to join **IJCSS** network of professionals/researchers for the positions of Editor-in-Chief, Associate Editor-in-Chief, Editorial Board Members and Reviewers.

The invitation encourages interested professionals to contribute into CSC research network by joining as a part of editorial board members and reviewers for scientific peer-reviewed journals. All journals use an online, electronic submission process. The Editor is responsible for the timely and substantive output of the journal, including the solicitation of manuscripts, supervision of the peer review process and the final selection of articles for publication. Responsibilities also include implementing the journal's editorial policies, maintaining high professional standards for published content, ensuring the integrity of the journal, guiding manuscripts through the review process, overseeing revisions, and planning special issues along with the editorial team.

A complete list of journals can be found at http://www.cscjournals.org/csc/byjournal.php. Interested candidates may apply for the following positions through http://www.cscjournals.org/csc/login.php.

Please remember that it is through the effort of volunteers such as yourself that CSC Journals continues to grow and flourish. Your help with reviewing the issues written by prospective authors would be very much appreciated.

Feel free to contact us at coordinator@cscjournals.org if you have any queries.

Contact Information

Computer Science Journals Sdn BhD
M-3-19, Plaza Damas Sri Hartamas
50480, Kuala Lumpur MALAYSIA

Phone: +603 6207 1607
+603 2782 6991
Fax: +603 6207 1697

BRANCH OFFICE 1
Suite 5.04 Level 5, 365 Little Collins Street,
MELBOURNE 3000, Victoria, AUSTRALIA

Fax: +613 8677 1132

BRANCH OFFICE 2
Office no. 8, Saad Arcad, DHA Main Bulevard
Lahore, PAKISTAN

EMAIL SUPPORT
Head CSC Press: coordinator@cscjournals.org
CSC Press: cscpress@cscjournals.org
Info: info@cscjournals.org

COMPUTER SCIENCE JOURNALS SDN BHD
M-3-19, PLAZA DAMAS
SRI HARTAMAS
50480, KUALA LUMPUR
MALAYSIA

Copyrights © 2008 Computer Science Journals. All rights reserved.