Volume 9 • Issue 4 • July / August 2015

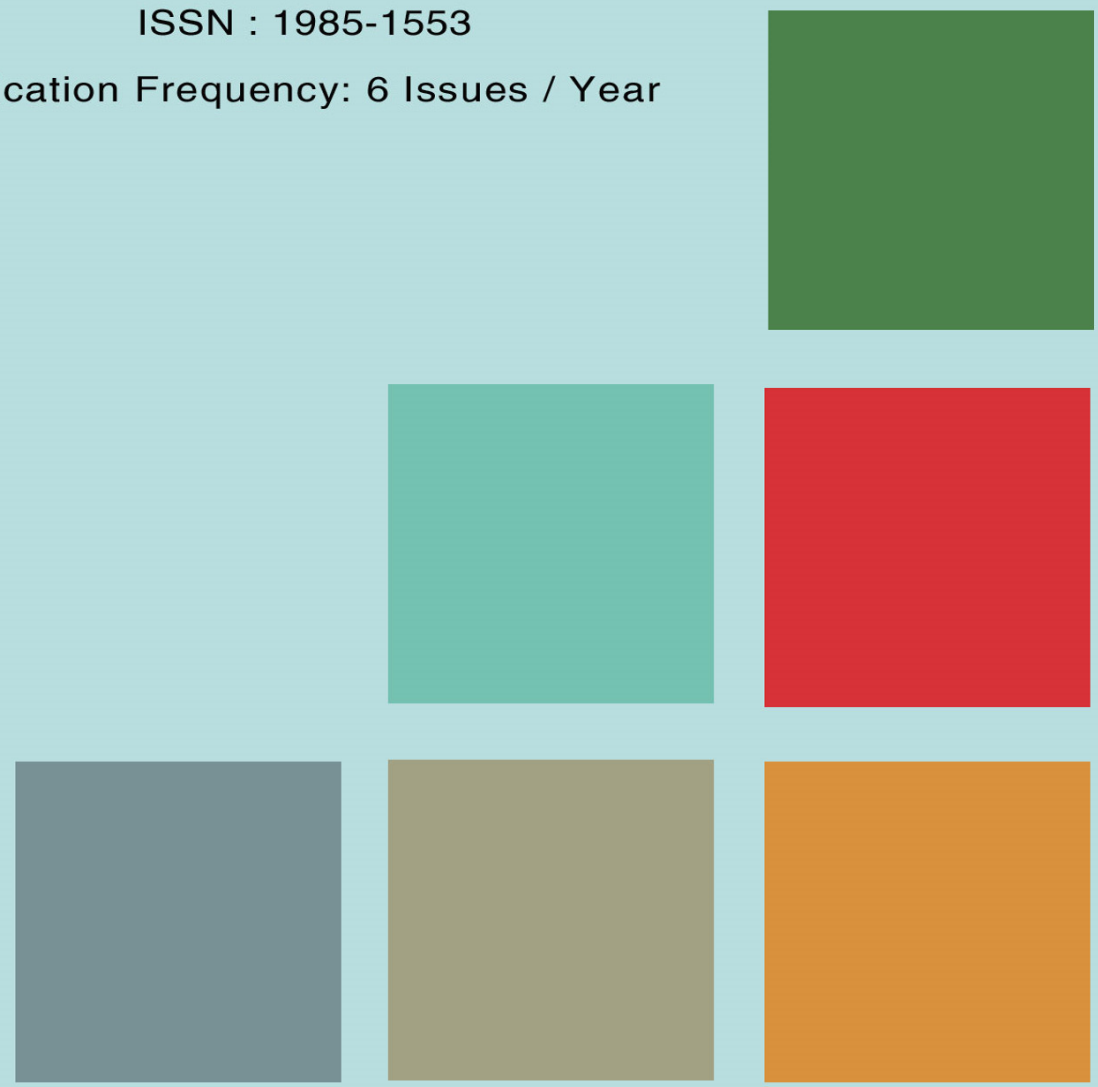
Editor-in-Chief
Dr. Chen-Chi Shing

INTERNATIONAL JOURNAL OF

COMPUTER SCIENCE AND SECURITY (IJCSS)

ISSN : 1985-1553

Publication Frequency: 6 Issues / Year



CSC PUBLISHERS
http://www.cscjournals.org

INTERNATIONAL JOURNAL OF COMPUTER SCIENCE AND SECURITY (IJCSS)

VOLUME 9, ISSUE 4, 2015

EDITED BY
DR. NABEEL TAHIR

ISSN (Online): 1985-1553

International Journal of Computer Science and Security is published both in traditional paper form and in Internet. This journal is published at the website http://www.cscjournals.org, maintained by Computer Science Journals (CSC Journals), Malaysia.

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

INTERNATIONAL JOURNAL OF COMPUTER SCIENCE AND SECURITY (IJCSS)

Book: Volume 9, Issue 4, July / August 2015
Publishing Date: 31-08-2015
ISSN (Online): 1985 -1553

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

Published in Malaysia

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

CSC Publishers, 2015

EDITORIAL PREFACE

This is *Fourth* Issue of Volume *Nine* 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.

The initial efforts helped to shape the editorial policy and to sharpen the focus of the journal. Started with Volume 9, 2015, IJCSS appears with more focused issues. Besides normal publications, IJCSS intend to organized special issues on more focused topics. Each special issue will have a designated editor (editors) – either member of the editorial board or another recognized specialist in the respective field.

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 and Security (IJCSS)

EDITORIAL BOARD

Assistant Professor Vishal Bharti
Maharishi Dayanand University
India

Dr. Parvinder Singh
University of Sc. & Tech
India

Assistant Professor Vishal Bharti
Maharishi Dayanand University,
India

TABLE OF CONTENTS

Volume 9, Issue 4, July / August 2015

Pages

Optimizing Linux Kernel for Real-time Performance On Multi-Core Architecture

P. Bala Subramanyam Raju *bsr3011@gmail.com*
Research Scholar,
S.V University,
Tirupathi Chittoor (Dt) AP, India

P. Govindarajulu *PGovindarajulu@yahoo.com*
Professor, Dept. of Computer Science,
S.V University,
Tirupathi Chittoor (Dt) AP, India

Abstract

Linux kernel developed and distributed in open source doesn't support for Hard Real-time scheduling. The open source Linux kernels are designed in time sharing manner to obtain maximum throughput. With this, Linux Operating System is considered to be an OS, which is not supporting Real-Time Applications, natively it has some features, already included in the mainstream to provide real-time support. There are certain modified Linux kernels like RTLinux, Symbian OS, Nucleus OS, Lynx OS and Fusion RTOS [1] which are explicitly designed for hard Real-Time support [2]. These specially designed Real-Time Linux kernels is mostly targeted for special hardware's like embedded systems, robots, safety critical etc. ,very few kernels for general purpose. Most of these kernels are be available as proprietary or closed, excluding a very few and not suitable for all hardware architecture's.

Now a days Real-Time Performance has become universal requirement for computer games, multimedia systems, household monitoring and controlling appliances. So the general purpose Linux kernel needs to be optimized, to achieve Real-time performance to meet the user expectations. This paper tries to extract real-time performance from general kernel and suggest some techniques to optimize Linux kernel to meet real-time deadlines.

Keywords: Kernel, Embedded Systems, Deadline, Real-time, Scheduler, Hyper Threading.

1. INTRODUCTION

Multi-core processor, delivering high computing power with reduction in hardware cost. This reduction in the hardware cost helps large number of people to purchase high performance computers. A normal user can run special types of applications like robot controller, applications collecting data from physical sensors, which are real time in nature and are not supported by open source Linux. In order to achieve real-time response, the General Purpose Linux kernel has modified by adding two real-time scheduling policies as shown in figure 1.1.

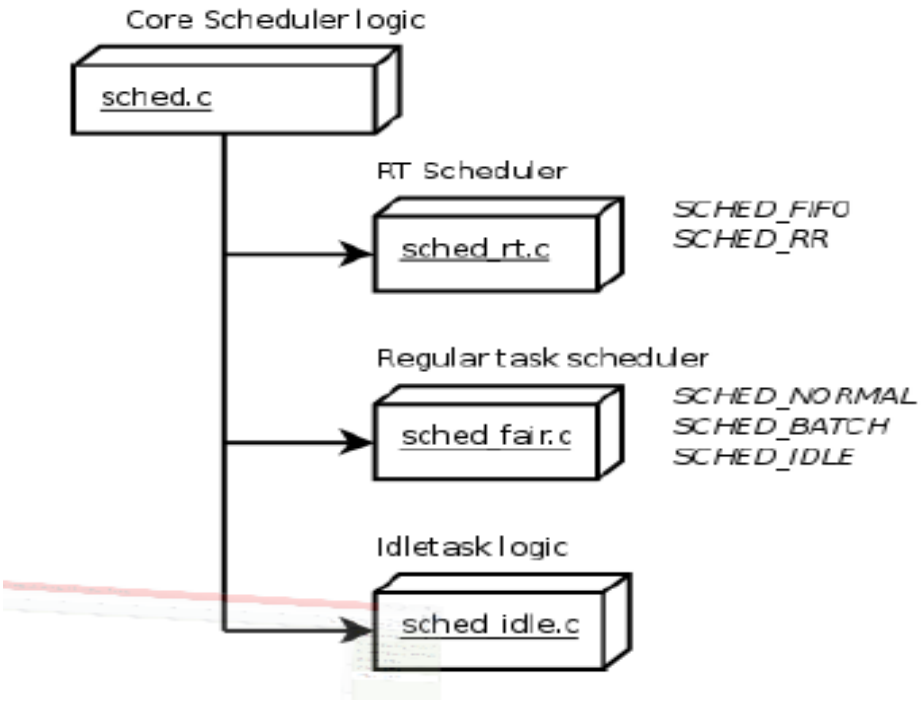


FIGURE1.1: Showing various Scheduling classes and policies in General Purpose Linux Kernel 4.0.

A real time system can be defined [3][4][5][6] as a *"system capable of guaranteeing timing requirements of the processes under its control".*

The following goals should be considered in scheduling a real-time system:

- Meeting the timing constraints of the system
- Obtaining a high degree of System utilization while satisfying the timing constraints
- Reducing the cost of context switches caused by preemption
- Reducing the communication cost in real-time distributed systems

Even after adding new scheduling classes and polices, the kernel needs to be fine-tuned to reach the goals of real-time system from multicore systems.

The remainder of the paper is organized as follows Section 2 discuss the previous work; section; Section 3 presents the hardware and software environment details, Section 4 describes an experimental performance evaluation; Section 5 proposes Optimization technique's. Section 6 gives implementation and experimental Results; Section 7 concludes the paper with future work.

2. PREVIOUS WORK

This section gives the overview of the research work carried out related to the performance improvement of Real-Time Systems.

Chenyang Lu, Xiaorui Wang and Xenofon Koutsoukos [7] proposed an approach to extend Quality of Service (QoS) from single processor to Distributed Real-Time Systems by a model predictive control approach. Utilization control is formulated as a multi-variable constrained optimization problem, second a dynamic model is established to formally characterize the coupling among multiple processor due to end to end tasks and practical constraints. MIMI model predictive controller is designed to control the utilization of multiple processors simultaneously. Finally stability analysis is performed to establish statistical guarantees on desired utilization

despite the uncertainty introduced by variation in task execution times. Simulation results demonstrate it can provide robust utilization guarantees when task execution times are significantly overestimated and change dynamically at run-time.

David Beal [8] an engineer of Freescale semiconductor, has specified that standard Linux kernel include enhanced schedulers, virtual memory, shared memory, POSIX Timers, Real-time Signals, POSIX IO,POSIX threads, Low Latency and many features that make Linux suitable for challenging real-time products and applications. Nat Hillary [9] of Freescale Semiconductor, presented methods for designing, measuring, improving the performance of real-time systems. Finally he concluded that real-time software is not something that can be done in a single step. Meeting required performance criteria can be obtained by careful consideration between the system and its environment needs. High fidelity software performance measurements may be achieved by using a combination of source code instrumentation and hardware.

Suresh Siddha,Venkatesh Pallipadi and Asit Mallick [10] proposed new scheduler optimization for Linux Kernel 2.6 for Chip Multi Processing (CMP). They discussed about generic OS Scheduler optimization opportunities that are appropriate in CMP environment. Henrik Austad, of Norwegian University of Science and Technology has introduced a new pfair algorithm for real-time tasks in the linux kernel on multi-core system. This scheduler will handle real-time task that cannon miss a deadline and is planned to be placed on top of the RT preemption patch. Due to problems faced during integration process, a fully functional scheduler has not been implemented.

Swati Pandit and Rajashree Shedge [11] has presented no of real-time scheduling algorithms that are suitable for simple uniprocessor and highly sophisticated multi-core processor. This paper also discusses the static, dynamic and hybrid priorities of a process. Finally conclusion shows that Instantaneous utilization factor scheduling algorithm gives better result in uniprocessor scheduling algorithms and Modified Instantaneous utilization factor scheduling algorithm gives better context switching, response time and CPU utilization as compared to previous scheduling algorithms.

Rohan R. Kabugade, S. S Dhotre, S H Patil [12] presents a modified algorithm named MOFRT (Modify O (1) For Real-Time) and Just-In-Time (JIT) based on the Linux kernel 3.2 to improve the Queue Management for Real time Tasks. Though, some of these algorithms have not been implemented since it is very hard to support new scheduling algorithms on nearly every operating system. However the previous works does not try to explore and fine tune the existing real-time supporting features included in the Linux Kernel Scheduler.

3. HARDWARE & SOFTWARE DETAILS

Processor	Intel® Core™ i7-4770k[5]
No of Cores	4
No of Threads	8
Base Frequency	3.5 GHz
Turbo Frequency	3.9 GHz
Intel® Smart Cache	8 MB
RAM	8 GB/1600 MHz

TABLE 1.1: Showing the hardware details.

- LINUX KERNEL 4.0-generic [14]
- LINUX MINT OS [15]
- TERMINATOR

- GCC COMPILER
- HTOP
- STRACE

4. EXPERIMENTAL PERFORMANCE EVALUATION

The goal of this experiment is to evaluate the performance of real-time scheduler included in the kernel 4.0. The program is designed to create a load to a multicore processor; and evaluate how far the existing scheduler supports for real-time programs in heavy load and normal situations, irrespective of other delays like data transfers, IO read & writes Network issues etc.

The algorithm of **"Load.C"** is as follows:

Step 1 : Start
Step 2 : repeat the following until true
step 2.1 : Stime=Read System time
step 2.2 : print 'Start time is:" stime
step 2.3 : j=0;
step 2.3 : Repeat the following steps until j<100
step 2.3.1 :i=0
step 2.3.2 :Repeat the following steps until i<1000000
step 2.3.2.1 : sum =sum +i;
step 2.3.2.2 : i=i+1
step 2.3.3 :i=0
step 2.3.4 :Repeat the following steps until i<1000000
step 2.3.4.1 : sum =sum +i;
step 2.3.4.2 : i=i+1
step 2.3.5 :i=0
step 2.3.6 :Repeat the following steps until i<1000000
step 2.3.6.1 : sum =sum +i;
step 2.3.6.2 : i=i+1
step 2.3.7 :i=0
step 2.3.8 :Repeat the following steps until i<1000000
step 2.3.8.1 : sum =sum +i;
step 2.3.8.2 : i=i+1
step 2.3.9 :i=0
step 2.3.10 :Repeat the following steps until i<1000000
step 2.3.10.1 : sum =sum +i;
step 2.3.10.2 : i=i+1
step 2.3.11 :i=0
step 2.3.12 :Repeat the following steps until i<1000000
step 2.3.12.1 : sum =sum +i;
step 2.3.12.2 : i=i+1
step 2.3.13 :i=0
step 2.3.14 :Repeat the following steps until i<1000000
step 2.3.14.1 : sum =sum +i;
step 2.3.14.2 : i=i+1
step 2.3.15 :i=0
step 2.3.16 :Repeat the following steps until i<1000000
step 2.3.16.1 : sum =sum +i;

step 2.3.16.2 : i=i+1
step 2.3.17 :i=0
step 2.3.18 :Repeat the following steps until i<1000000
step 2.3.18.1 : sum =sum +i;
step 2.3.18.2 : i=i+1
step 2.4 : etime=read system time
step 2.5 : print 'End time is:"etime
step 2.6 : 'The Loop used :' etime-stime 'seconds'
step 3 move to step 2
step 4 stop

The designed program has been executed with two different schedulers and priority. One with RT class, FIFO scheduler and highest priority of 99 using the command “chrt –f 99. /FIFO”. Other has executed with Fair Scheduler class and normal priority using the command “. /a.out”. The results are tabulated below:

Real –Time Threads	Normal Threads	Total No of Threads	Real-time Thread Failures/Sec
1	0	1	0
1	1	2	0
1	2	3	0
1	3	4	0
1	4	5	2
1	5	6	5
1	6	7	5
1	7	8	5

TABLE 1.2: showing execution results of Load.C.

The maximum time taken by the specified system to complete the loop execution is 2 seconds. The above results show that Linux Kernel supports real-time performance by default, until no of threads is equal to no of physical cores in the system without any deadline failure .After the no of threads increases than physical cores the system performance starts degrading.

The figure 1.2 shows the execution of “Load.c” program using terminator.

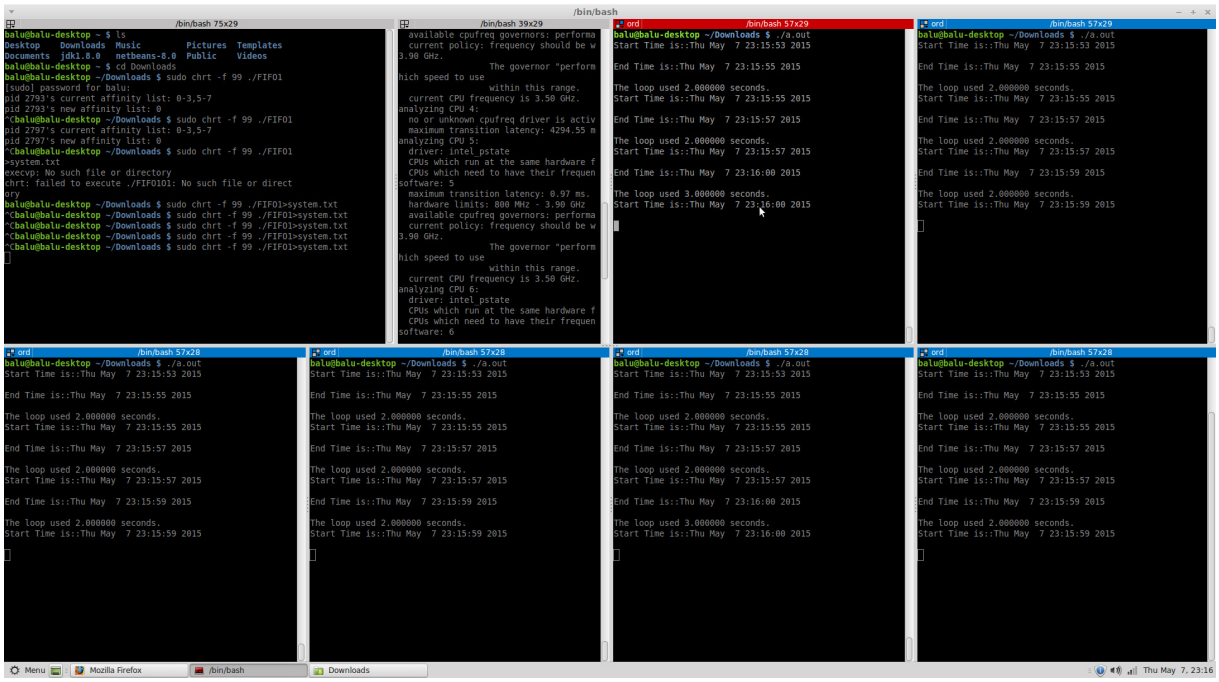


FIGURE 1.2: showing real-time thread and 7 normal thread execution.

5. PROPOSED OPTIMIZATION TECHNIQUE’S AND IMPLEMENTATION

There Linux kernel needs to be fine-tuned to support real-time environment in heavy load situations ,because the default kernel fully supports multitasking to increase the overall throughput buy using completely fair scheduling class the optimization techniques are specified below.

5.1 No Force Preemption

Preemption is one of the bottlenecks to real-time performance, because the kernel will preempt the thread non-voluntarily irrespective of program requirements and priorities, in order to support completely fair scheduling class. The default Linux kernel available in General Public License is built in way that the kernel can preempt the thread non-voluntarily. To improve real-time performance the Linux kennel needs to rebuild and update boot loader in a way that the kernel should not force thread to preempt. The procedure is explained below.

1. Download latest kernel from www.kernel.org[16]
2. Install git-core,libncurses5-dev tools required to build Linux kernel
3. Configure the options required using the $ make menuconfig
4. In menu configuration options select processor type and features-> preemption model Select “**No forced preemption (server)** “model shown in fig 1.3
5. Then build using make command
6. Install modules using command **$sudo make modules-install**
7. Install kernel using **$sudo make**
8. Update system configuration using **$ sudo update-initramfs-c-k 4.0**
9. Update boot loader **$ sudo update-grub**

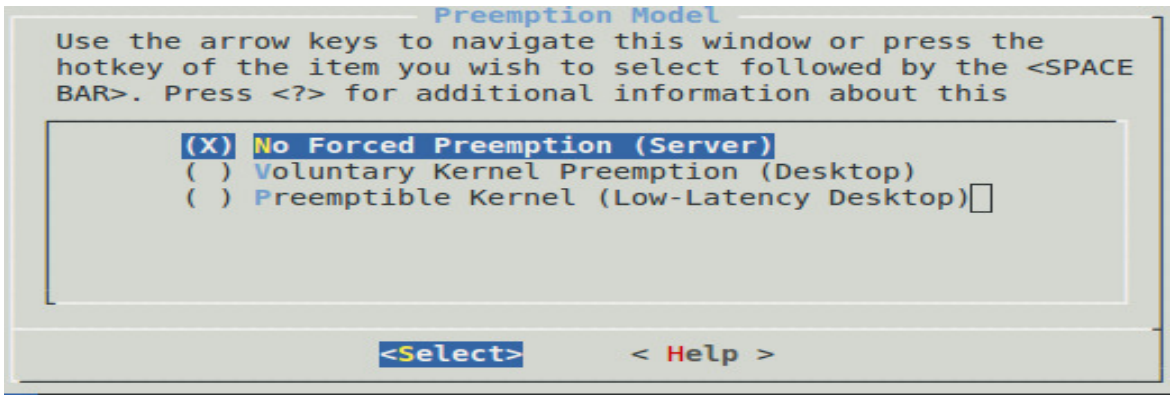


FIGURE 1.3: Showing the setting the preemption model during kernel building.

5.2 Stop Non-Voluntary Context Switching

The default Linux kernel will move thread from one core to other, in order to load balance the execution cores. This context switching takes CPU time to move executing thread from one core to other, by blocking the execution, which results in the execution delay leads to real-time failure. This can be avoided by setting the allowed CPU list to one specific core using the command **"taskset"**. The figures 1.4 and 1.5 shows the thread allowed CPU's list before and after setting CPU affinity. This can be viewed by using a command **"cat /proc/pid/Status"**,where pid is processID of a thread that needs to bind to a processor[17][18][19].

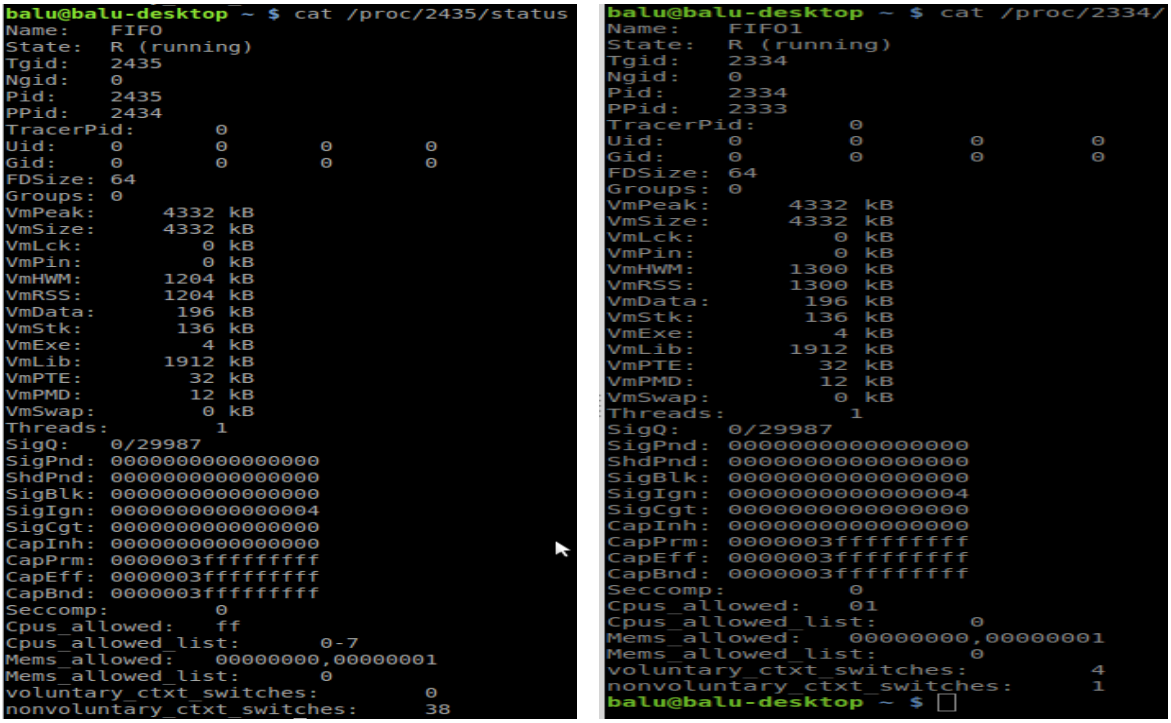


FIGURE 1.4 & 1.5: Showing process details before and after setting CPU affinities list and reduced context switches.

5.3 Hyper Threading (HT)

Intel Hyper-Threading Technology (Intel® HT Technology) is a technology used by some Intel microprocessors that allows a single microprocessor to act like two separate processors to the operating system and the application programs that use it. HT Technology utilizes resources

more efficiently .As a performance feature, it also increases processor throughput, improving overall performance on threaded software. Figure 1.6 showing the difference between Multiprocessor and Hyper threading processor [22][23][24].

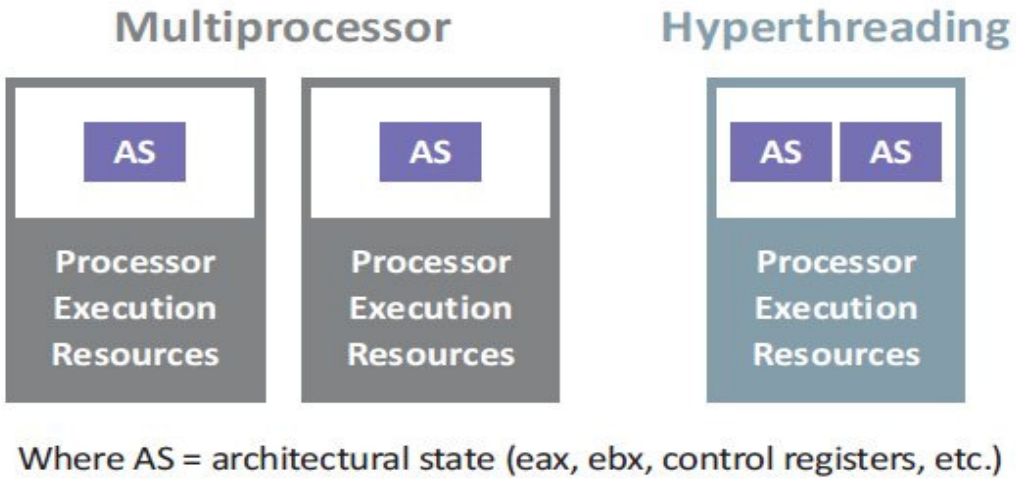


FIGURE 1.6: showing technological difference between multiprocessor and HT enabled processor.

This HT Technology becomes a major drawback for real-time performance because, the two threads compete for execution resource on single execution unit, and this results in sharing the CPU cycles between the two threads. Sharing real-time thread CPU cycles results in deadline failure, due to less execution time. To utilize 100 percent CPU cycles for real-time thread, the other thread on the core needs to be disabled. This can be achieved using the command **"echo 0 | sudo tee /sys/devices/system/cpu/cpu4/online",** here core no 4 has disabled. The figure1.7 showing the cpu4 disabled on quad core 8 threaded processor.

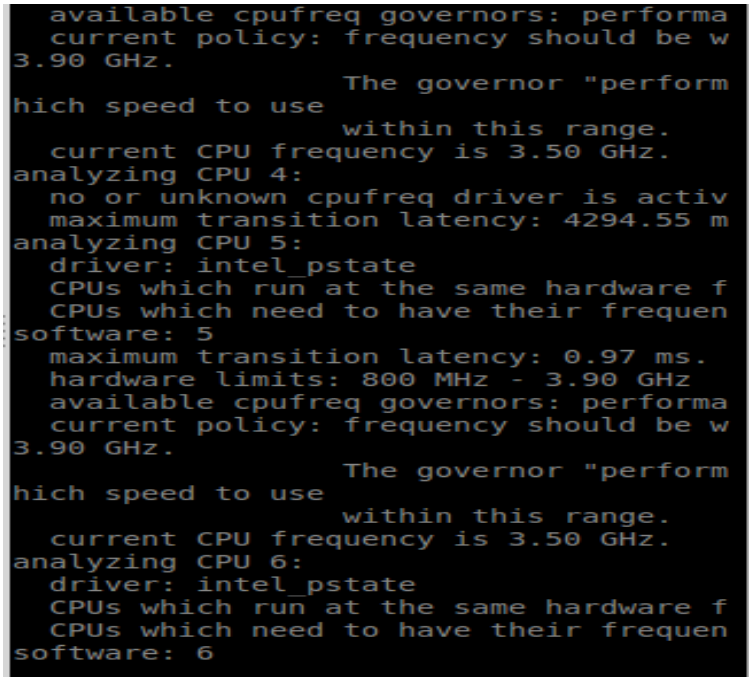


FIGURE 1.7: showing the cpu4 disabled on quad core 8 threaded processor.

6. EXPERIMENTAL RESULTS AFTER OPTIMIZATION

After fine tuning the operating system and hardware, the program also needs to change according to the requirements for real-time. The changed algorithm is shown below.

Step 1 : Start
Step 2 : Read current process ID from kernel
tid= getpid();
Step 3 : Set the current thread to execute on CPU0 using taskset –cp 0 tid
Step 4: Disable core 4 by executing the command as super user
"echo 0 |sudo tee /sys/devices/system/cpu/cpu4/online
Super user password
Step 5 : repeat the following until true
step 5.1 : Stime=Read System time
step 5.2 : print 'Start time is:" stime
step 5.3 : j=0;
step 5.3 : Repeat the following steps until j<100
step 5.3.1 :i=0
step 5.3.2 :Repeat the following steps until i<1000000
step 5.3.2.1 : sum =sum +i;
step 5.3.2.2 : i=i+1
step 5.3.3 :i=0
step 5.3.4 :Repeat the following steps until i<1000000
step 5.3.4.1 : sum =sum +i;
step 5.3.4.2 : i=i+1
step 5.3.5 :i=0
step 5.3.6 :Repeat the following steps until i<1000000
step 5.3.6.1 : sum =sum +i;
step 5.3.6.2 : i=i+1
step 5.3.7 :i=0
step 5.3.8 :Repeat the following steps until i<1000000
step 5.3.8.1 : sum =sum +i;
step 5.3.8.2 : i=i+1
step 5.3.9 :i=0
step 5.3.10 :Repeat the following steps until i<1000000
step 5.3.10.1 : sum =sum +i;
step 5.3.10.2 : i=i+1
step 5.3.11 :i=0

step 5.3.12 :Repeat the following steps until i<1000000
step 5.3.12.1 : sum =sum +i;
step 5.3.12.2 : i=i+1
step 5.3.13 :i=0
step 5.3.14 :Repeat the following steps until i<1000000
step 5.3.14.1 : sum =sum +i;
step 5.3.14.2 : i=i+1
step 5.3.15 :i=0
step 5.3.16 :Repeat the following steps until i<1000000
step 5.3.16.1 : sum =sum +i;
step 5.3.16.2 : i=i+1

step 5.3.17 :i=0
step 5.3.18 :Repeat the following steps until i<1000000
step 5.3.18.1 : sum =sum +i;
step 5.3.18.2 : i=i+1
step 5.4 : etime=read system time
step 5.5 : print 'End time is:"etime
step 5.6 : 'The Loop used :' etime-stime 'seconds'
step 6 move to step 2
step 7 stop

The table 1.3 shows the execution results after fine tuning the system for real-time performance.

Real –Time Threads	Normal Threads	Total No of Threads	Real-time Thread Failures/Sec
1	0	1	0
1	1	2	0
1	2	3	0
1	3	4	0
1	4	5	0
1	5	6	0
1	6	7	0
1	7	8	0

TABLE 1.3: showing execution results of real-time and normal threads.

7. CONCLUSION AND FUTURE WORK

The results show that after fine tuning the Linux kernel in consideration with hardware, it is possible to extract real-time performance from generally available open source Linux kernel.

This paper doesn't on concentrate on implementation of fully non-preemptible kernel. This experiment results in delay for normal priority process. If the number of real-time process increases and starts disabling the cores then only real-time process will execute until the hardware supports and normal process will block execution leads to imbalance execution.

8. REFERENCES

[1] Wikipedia Internet:www.en.wikipedia.org/wiki/List_of_real-time_operating_systems [May 10, 2015].

[2] Embedded Internet: www.embedded.com/design/operating-systems/4371651/9/Comparing-the-real-time-scheduling-policies-of-the-Linux-kernel-and-an- RTOS [May 10, 2015].

[3] Wikipedia Internet: www.en.wikipedia.org/wiki/Real-time_operating_system [May 10, 2015].

[4] Peter wurmsdobler "Real Time Linux Foundation, Inc.". Internet : www.realtimelinuxfoundation.org/ [May 10, 2015].

[5] Fernando S. Schlindwein “Real-time DSP” Internet:www.le.ac.uk/eg/fss1/real%20time.htm. [May 10, 2015].

[6] Kanaka Juvva “Real-Time Systems”Internet: www.users.ece.cmu.edu/~koopman /des_s99/real_time/. [May 10, 2015].

[7] Chenyang Lu, Xiaorui Wang, Xenofon Koutsoukos,” End-to-End Utilization Control in Distributed Real-Time Systems”, Distributed Computing Systems, 2004. Proceedings. 24th International Conference, 2004.

[8] David Beal,”Linux® As a Real-Time Operating System” , Freescale Semiconductor, Document Number: SWVERIFICATIONWP Rev. 0 11/2005.

[9] Nat Hillary,” Measuring Performance for Real-Time Systems” , Freescale Semiconductor, Document Number: GRNTEEPFRMNCWP Rev. 0 11/2005.

[10] Suresh Siddha, Venkatesh Pallipadi,” Chip Multi Processing aware Linux Kernel Scheduler”, Linux Symposium, Volume 2, 2006.

[11] Swati Pandit and Rajashree Shedge,” Survey of Real Time Scheduling Algorithms ” IOSR Journal of Computer Engineering e-ISSN: 2278-0661, p- ISSN: 2278-8727 Volume 13, Issue 2 (Jul. - Aug. 2013), pp 44-51,

[12] Rohan R. Kabugade, S. S Dhotre, S H Patil,” A Study of Modified O(1) Algorithm for Real Time Task in Operating System”, Sinhgad Institute of Management and Computer Application NCI2TM: 2014 ISBN: 978-81-927230-0-6.

[13] www.ark.intel.com/products/75123/Intel-Core-i7-4770K-Processor-8M-Cache-up-to-3_90-GHz/[May.10, 2015].

[14] Internet: www.kernel.org,[May.10, 2015].

[15] Linux Mint Internet:www.linuxmint.com//[May.10, 2015].

[16] Lakshmanan Ganapathy,” How to Compile Linux Kernel from Source to Build Custom Kernel” Internet: www.thegeekstuff.com/2013/06/compile-linux-kernel/ June 13, 2013 [May. 10, 2015].

[17] Robert Love,”CPU Affinity” Internet: www.linuxjournal.com/article/6799 [May.10, 2015].

[18] Internet: www.gnu.org/software/libc/manual/html_node/CPU-Affinity.html[May.10, 2015].

[19] Internet: www.linux.die.net/man/1/taskset[May.10, 2015].

[20] Alexander Sandler, April 15, 2008 “SMP affinity and proper interrupt handling in Linux” Internet: www.alexonlinux.com/smp-affinity-and-proper-interrupt-handling-in-linux[May.10, 2015].

[21] Sandeep Krishnan on January 27, 2014,” Introduction to Linux Interrupts and CPU SMP Affinity”, Internet: www.thegeekstuff.com/2014/01/linux interrupts/ [May.10, 2015].

[22] www.intel.in/content/www/in/en/architecture-and-technology/hyper-threading/hyper-threading-technology.html [May.10, 2015].

[23] http://whatis.techtarget.com/definition/Hyper-Threading[May.10, 2015]

[24] Internet : www.doc.opensuse.org/products/draft/SLES/SLES-tuning_sd_draft/cha.tuning.taskscheduler.html[May.10, 2015].

Design for A Network Centric Enterprise Forensic System

Hongye Zhong *hzhong@our.ecu.edu.au*
School of Computer and Security Science
Edith Cowan University
WA 6050, Australia

Jitian Xiao *j.xiao@ecu.edu.au*
School of Computer and Security Science
Edith Cowan University
WA 6050, Australia

Abstract

Increased profitability and exposure of enterprise's information incite more attackers to attempt exploitation on enterprise network, while striving not to leave any evidences. Although the area of digital forensic analysis is evolving to become more mature in the modern criminology, the scope of network and computer forensics in the large-scale commercial environment is still vague. The conventional forensic techniques, consisting of large proportion of manual operations and isolated processes, are not adequately compatible in modern enterprise context. Data volume of enterprise is usually overwhelming and the interference to business operation during the investigation is unwelcomed. To evidence and monitor these increasing and evolving cyber offences and criminals, forensic investigators are calling for more comprehensive forensic methodology. For comprehension of current insufficiencies, this paper starts from the probes for the weaknesses of various preliminary forensic techniques. Then it proposes an approach to design an enhanced forensic system that integrates the network distributed system concept and information fusion theory as a remedy to the drawbacks of existing forensic techniques.

Keywords: Network, Forensic, Information Security, Enterprise.

1. INTRODUCTION

With the evolution of networking technology and mobile computing, portable devices and communication vehicles such as mobile phones, laptops email and social networks are pervasively participated in our daily lives and production environment. When individuals and enterprises happen to encounter legal or corporate issues, evidences are required to be collected from relevant electronic devices to support legal or business decisions [1]. Moreover, auditing and examining digital trails are usually enabled on computer devices to discover or assure whether the information is secure or has been tempted. Apart from these scenarios, various cases entail computer forensics including collecting reliable digital evidences for a law case in a court of justice. The increasing requirements and complexity of investigation becomes a spur for the study and application of computer forensic science.

The primary purpose of computer forensics is to dig up data to expose or assure what and when something has been done, and by whom, where, why and how. With the improvement of forensic theory and the accumulation of forensic practices, various forensic process, workflow and techniques have been introduced to ensure this fundamental purpose of computer forensics can be accomplished, while the essential procedures of forensic process almost remain unchanged.

Forensic process is a mechanism that uses scientific methods to discover digital evidences. The evidences found will be utilized to support or disprove a hypothesis or reveal or verify works done by others. Forensic process generally is consisted of the five linear procedures in terms of Plan,

Acquire, Extract, Analyze, and Report [2]. In practice, the forensic process can be organized and interpreted into a Model of Digital Forensic Analysis (MDFA, Fig. 1), which eases the implementation of the process and enhances its controllability in a clear and systematic manner.

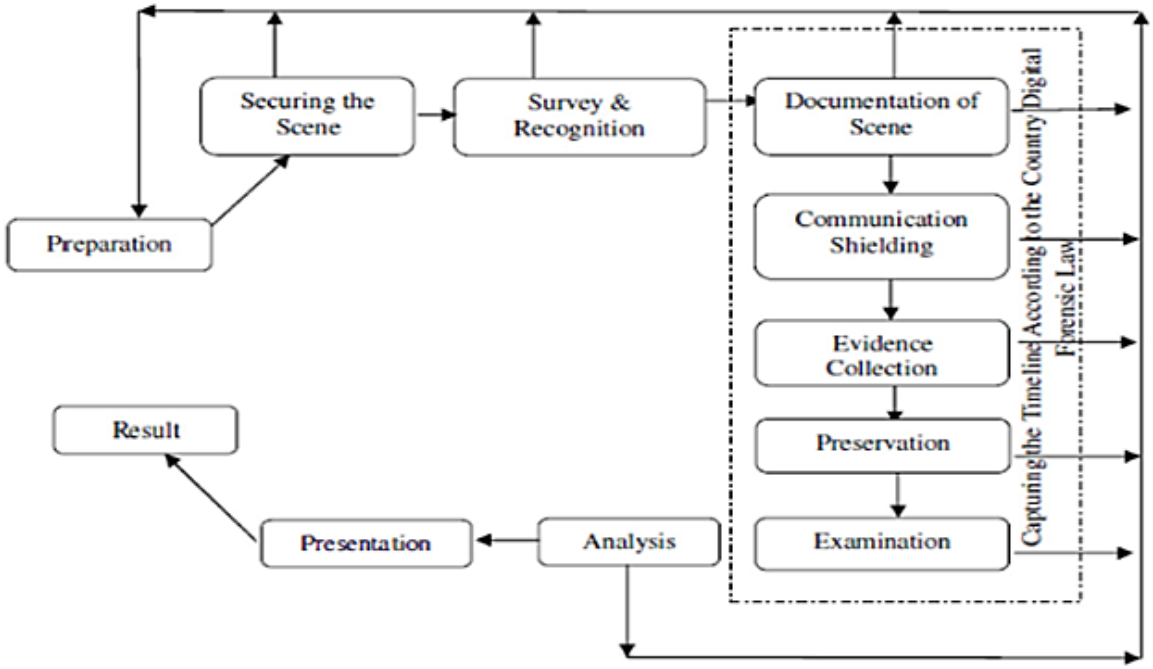


FIGURE 1: Model of Digital Forensic Analysis.

2. PRELIMINLARY FORENSIC TECHNIQUES

Forensic techniques are the means employed by the investigators with the goal to discover and retrieve the evidences stored in devices. Although the investigation methods might vary from case to case, general methods to achieve the goal of digital forensics could be summarized in two categories: Locating Data and Capturing Data [3].

Locating Data is the process of discovering sensible or relevant data from the storage component of devices, which usually involves investigations on components, such as file systems and disk cluster, memory and process image, and history and temporary files. Capturing Data is a more active way for digital investigation. Instead of searching evidences left on the devices, it creates evidences. When the investigator estimates that a typical device might contain data closely related to an incident, he/she might covertly implant some monitoring mechanism onto the targeted device/s to intercept the data processed on the device/s, with the hope to attain useful information that can become digital evidences. The techniques usually used in Capturing Data include keystroke logger, wiretapping, and spyware [3].

In recent years, large number of electronic devices has been adopted by enterprise, both in its operation and production. Meanwhile, more vulnerabilities and security issues have also been introduced, which increases the occurrence of crimes and internal incidents related to information security. These security issues sometimes come from the negligence or fraudulence of its staff, or the attacks from outsiders. Most of the cases require investigations on the electronic devices to collect evidences for making decisions. In some other cases, when the enterprise is under some security agreements, digital investigations are also needed to assure the enterprise's compliance. For example, if an enterprise has adopted the ISO 27001 Security Standard, evaluation needs to be taken annually to assure the enterprise has been obeying the codes in the standard.

Investigators usually find the difficulty is severely increased in collecting evidences with the conventional forensic techniques in enterprise context, rather than in civil cases. In the modern enterprise environment, especially in large organizations, the amount of data and information is usually overwhelming. In some cases, the data is even geographical distributed [1]. Frequently, the suspected users who are targeting the enterprise's sensitive information are usually trained and skilled. In contrast, the inadequate computer training of employees may damage the evidence before the arrival of the investigator/s. The difficulty might be increased, if the enterprise might not want the intervention to production while the data is being collected for investigation. All of these hurdles arouse the need to design a more sophisticated and centralized process for computer forensics in enterprise.

3. ENHANCEMENT ON FORENSIC PROCESS

The logical and geographical scattering of data and devices increases the difficulty of evidence collection and hinders the investigation. As most enterprises have different running patterns, we need to consider a strategy that can fit most situations. For digital investigation, the procedures of forensic process are basically mandatory for all investigations. We should break down the forensic process to several operational tasks so as to be more easily adapted them into the information systems of different enterprises [4].

3.1 Enhanced Forensic Workflow

In accordance with enterprise forensic convention, MDFA can be interpreted into more executable and measurable steps. These steps are shown and linked up in the Enhanced Forensic Workflow (EFW, Fig. 2), which generally consists of the following stages.

Collection Planning - targeting device, execution time and search strategy should be determined for the data collection process.

Physical Media Identification - the targeted device are identified and labelled, so that they can be easily located and identified later.

Media Identifier Creation - the media is tagged with date/time and the disk images are acquired for future reference and research purpose.

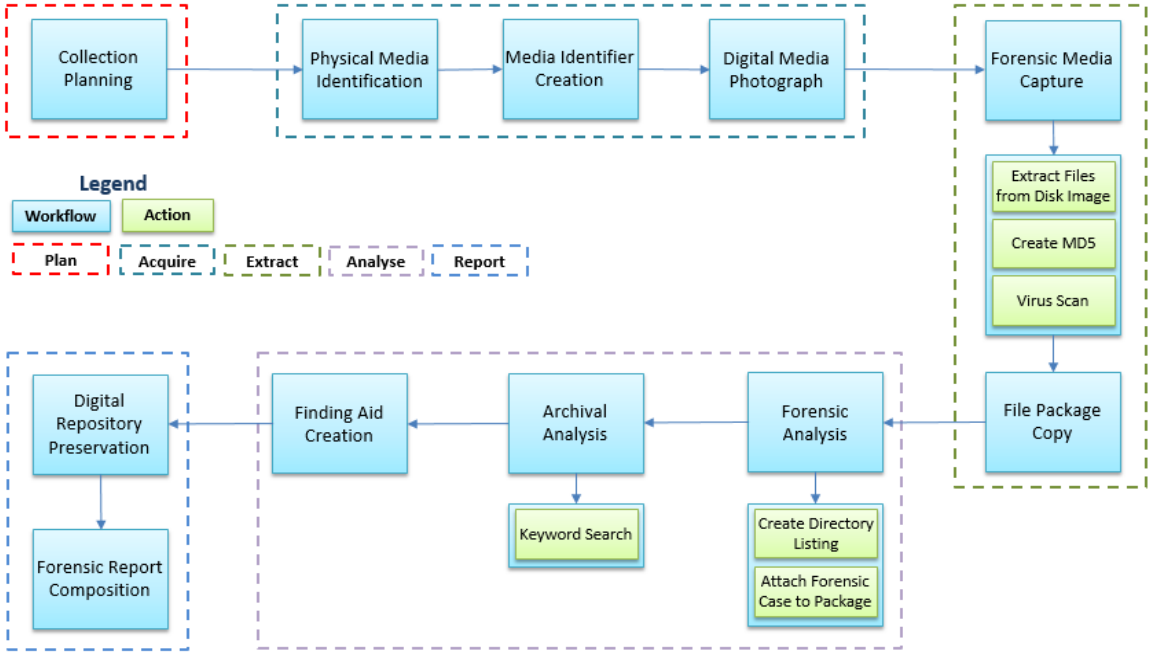


FIGURE 2: Enhanced Forensic Workflow.

Digital Media Photograph - the photo of the device should be recorded and kept in digital form, which helps to identify whether the device has been physically altered. It also helps to easier identify the device.

Forensic Media Capture - the file contained in the disk images acquired above will be extracted and properly tagged.

File Package Copy - captured disk images, metadata information and extracted files will be transferred to a server computer where analysis will be performed.

Forensic Analysis - captured disk images, metadata information and extracted files will be resorted, indexed and organized into an understandable hierarchy, so that the relationship between the evidences can be easily identified.

Archival Analysis - analysis will be performed by searching the collected information and file with selected searching heuristics. The analysis is aimed to identify any breaches to the enterprise security regulations.

Finding Aid Creation - the analysis results will be sorted and linked with original data. Hints and evidences discovered should be annotated so that the evidences are more comprehensible.

Digital Repository Preservation - the original data and findings are preserved and indexed in an isolated location where the evidences cannot be contaminated. These findings are likely to be referred in reporting and assist decision makings.

Forensic Report Composition - by drawing links between original data and the findings and explaining the relationships between them and what they are leading to, an investigation report will be composed to give an overall picture of the investigation.

According to the workflow, the initial tasks involve acquisition of the information from all other information systems in the enterprise. The collected information is assembled into a database for further analysis. In most cases, analysis and report will necessarily be performed based on the data stored in the evidence database. The forensic tasks are planned to be executed periodically, so that it can cover all incidents and detect them in a timely manner. This forensic methodology ensures the efficiency and adequate coverage of the information collection. To ensure the recorded evidences can reflect the most detailed and honest facts about the incidents, the tasks of the workflow are expected to be executed adequately in investigations.

3.2 Automation of the Forensic Workflow

With the objective of reducing the process mistakes and shortening the process duration, the investigation should be executed in a systematic and automatic manner. Since the forensic process has been broken down to smaller executable units of tasks, the automation becomes easier to implement. In the workflow, certain manual processes are unavoidable such as labelling or photographing the device. To maximize the automation processing of the workflow, such processes are required to be handled beforehand. Due to the nature of the processes, the original forensic workflow can be divided into two corresponding workflows: Preliminary Forensic Workflow (PWF), and Automated Forensic Workflow (AWF).

The PFW (Fig. 3) is consisted of a group of processes to be performed to ensure that the devices of the enterprise are prepared for automated data collection. This workflow needs to be executed once for each device only when the device is procured into the enterprise.

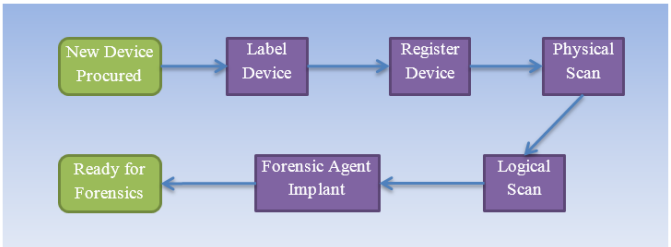


FIGURE 3: Preliminary Forensic Workflow.

The main purpose of PFW is to register the physical and logical identity information of the device for future referencing purpose. This kind of processes have to be done manually. For asset management purposes, normally enterprise has already registered every new device when they were procured. Hence it will be convenient to simply embed the original device register procedure into our workflow. For such purpose, whenever a new device is procured, the following steps must be followed.

Label Device - when a new device is procured by the enterprise, a unique identity number will be generated and labeled on to the device. This identity number is going to be the key to uniquely distinguish between devices.

Register Device - after the device is labelled, its relevant information will be recorded into the forensic system. The register information will be stored in the database of the forensic system along with the forensic information collected in future, and indexed by its unique identity number.

Physical Scan - after the register information of the device has been created, the 3D model of device should be scanned and saved as a part of the register information. Unless the device suffers physical damage, this property of the device does not need to be modified.

Logical Scan - this step is to make an initial record of the disk image of the imported device. In future, disk image might be acquired multiple times for forensic purposes. For example, the record of the disk image will be acquired periodically for comparison analysis. The previous image data will not be removed. The serials of disk image records will be maintained in the forensic database and kept in a hierarchy pattern, and labelled with time stamps. As the result, the image records can be compared and referenced for forensic analysis.

Implant Forensic Agent - in order to transmit forensic data in response to the request of forensic server, a client service needs to be installed and kept running on the device. This service will act as the coordinator between the operating system of the device and the forensic server to perform the task in correspondence with the server requests.

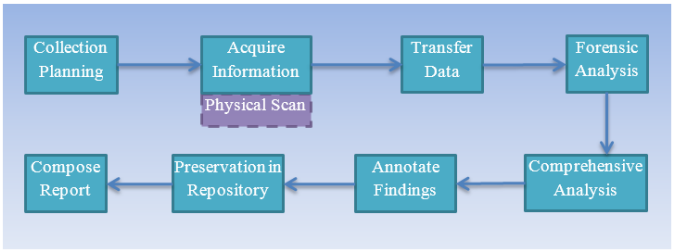


FIGURE 4: Automated Forensic Workflow.

The AFW (Fig. 4) is a set of processes that collect and analyze forensic data from the device, and generates a report to illustrate the findings. AFW assumes the newly procured devices have been properly handled in the preliminary workflow, so that all devices should have been kept

connected with the forensic server via the Forensic Agent implanted in the devices. Through a remote request, forensic server can instruct the devices to carry out the forensic workflow periodically.

The preliminary condition of the automatic workflow is that all devices of the enterprise have been labelled and registered, and the original manual processes of labelling and photographing devices has been removed. AFW continues the incomplete investigations of PFW by working on the recorded devices and corresponding information. Such preliminary condition eliminates the necessity to identify and to label the device again when the forensic investigation is performed on the devices. In most cases, the digital evidences are contained in the logical components of the devices when incidents happen. Unless the incidents involve physical damages to the devices, physical scan on device will not be necessary. Accordingly, the tasks for each stage of the forensic workflow are slightly different, depending on whether there are needs for physical scanning etc.

Collection Planning - in the planning stage, the forensic server schedules and configures the data collection, and dispatches the forensic requests to forensic agents of all devices of the enterprise.

Acquire Information - unlike the original forensic workflow, the Acquire stage is not going to take the whole dump of disk image every time. According the needs and nature of the investigation, the Forensic Agent will selectively acquire the disk image, partial dump, or the log files etc.

Transfer Data - when the Forensic Agent finishes acquiring relevant data, the device will transfer the collected data to the forensic server.

Forensic Analysis - when the forensic server receives the collected data from the devices, it will extract data and form files from the raw data. Then the extracted files will be sorted and indexed in a more sensible and analyzable form.

Comprehensive Analysis - the sorted files will be further analyzed with the customized analysis strategy and the predefined searching heuristics, in order to discovers any hints or evidences from the collected information.

Annotate Findings - analysis results from the previous stages will be sorted and indexed, and the reasoning and other interesting information related to the analysis results will also be annotated.

Preservation in Repository - the collected information and the corresponding findings will be sorted and stored in the database of the forensic server.

Compose Report - after finishing the above steps, a summary about the current investigation will be generated. In the summary, the brief of collected information, findings and the investigation process itself will be documented. The summaries will be saved to the forensic database, which can be exported and printed in a more representable form when they are needed to be referenced in future.

4. DESIGN OF NETWORK CENTRIC FORENSIC SYSTEM

The automated forensic workflow introduced above can be implemented in an enterprise as a comprehensive system by combining the existing enterprise networks with modern intelligent computing approaches. This system is called Network Centric Enterprise Forensic System (NCEFS). NCEFS is consisted of two parts: the client side and the server side applications. The client side application is named Forensic Agent (FA), and the forensic server is named Centralized Forensic Processor (CFP).

4.1 Forensic Agent

The FA is a client side application installed on devices of the enterprise. It acts as a coordinator that communicates between CFP and the devices (see Fig. 5). Once FA is installed on a device, it installs itself as a daemon service that starts running once OS starts. Once FA starts, it will establish a TCP connection with the CFP. Since TCP is a stateful network protocol, the connection is always being listened by the server, by which CFP can monitor the running status of the devices and transmit control requests. As FA needs to execute some sensitive processes such as disk imaging, memory imaging and data transmission, special execution privileges should be granted to FA. When FA receives forensic request from CFP, it will interpret the request and ask the hosting OS to complete the request. The requests usually contain various forensic tasks such as collecting certain information about the device or transferring the collected data to CFP.

With FA, the conventional device information such as disk image and files is able to be collected remotely as long as the device is remained connected with CFP. Moreover, as many modern computer devices have been equipped with some advanced input components such as light sensor and camera, they can be used to capture extra evidences remotely when such evidences are needed in certain investigations. Through the operation of FA, the devices are also remotely controllable by CFP. For example, when an incident happens, CFP can lock up the system of a specific device to protect the evidences contained on the device from being contaminated.

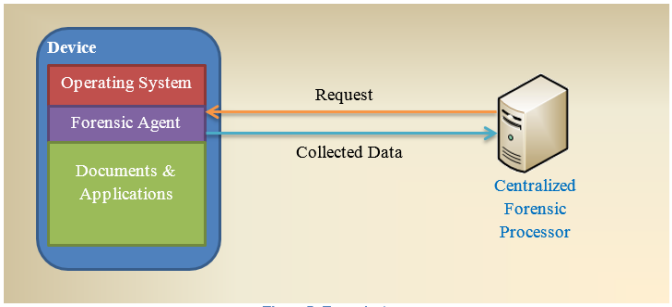


FIGURE 5: Functionality of Forensic Agent.

4.2 Centralized Forensic Processor

The Centralized Forensic Processor (CFP, Fig. 6) is a dedicated server that automatically organizes and controls the processing of digital forensics on the devices of the enterprise. CFP is closely corresponding with the forensic workflow. The role of CFP is to guarantee the steps in forensic workflow to be executed in a timely manner. As the workflow is performed automatically, the collected evidences and generated reports can be stored and managed more appropriately with labelling and sorting processes, so that the retrieved information are analyzable and can be easily referenced when they are needed in the future. CFP is mainly consisted of five components: Planner, Collector, Storer, Analyzer, and Reporter.

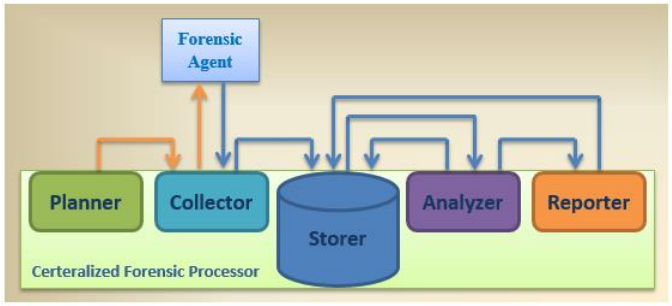


FIGURE 6: Centralized Forensic Processor.

CFP Planner - it is the component that systematically schedules and organizes information collecting in accordance with the devices status, network situation, and investigation requirements. As CFP is able to query the status of the devices, Planner can schedule the devices to transfer the collected information when the devices are idle, with the purpose of avoiding interference to the normal usage of the devices. As there might be considerable number of devices connected with CFP, to avoid network conjunction, the Planner attempts to request the devices to transfer data when the network traffic is idle. The devices under investigation should not be required to upload full dump of disk image every time the information is collected. Instead, Planner considers the natures and aims of the investigations, and requests the device to collect only necessary data accordingly. For instance, for investigating whether there are devices in the enterprise breached the Application Installation Agreement of Apple Inc., CFP only needs to request all the iOS devices to submit the lists of their installed applications. By doing so, the side effects of forensic investigation to the enterprise normal operation can be greatly reduced.

CFP Collector - the information collection schedules made by CFP Planner will be passed to Collector to be executed. When Collector receives an information collection request, it will interpret and dispatch the request to FA of the targeted device via the TCP tunnel established. When the collection request has been completed, or failed, FA will inform Collector to incept the collected information.

CFP Storer - it is functionally an internal database that stores the collected data, discovered findings and summarized reports. The information stored in the database should be indexed with the unique identity number of the devices, so that they can be rapidly referenced and queried. The database should be encrypted and the creation time and modification of all the data should be time stamped and recorded to ensure the collected evidences will not be contaminated. Moreover, for security purpose, the database is only accessible to the other components of CFP, and it must not be visited from the outside of the server.

CFP Analyzer - it is an intelligent facility that can refine the raw data into more understandable form and conduct customized searching and heuristic searching on the collected information to discovery the hints and patterns hidden in the data. After analysis, the findings will be stored into the database and forwarded to Reporter to generate a summary about the current investigation.
Reporter - all the collected information, discovered findings, and the investigation itself such as process duration, involved devices etc. will be assembled and summarized into a brief and comprehensible conclusion. The generated summaries will then be saved in Storer for future references and analysis. The saved reports can be exported from the database and print out into a representable form.

4.3 Collection and Analysis Strategy

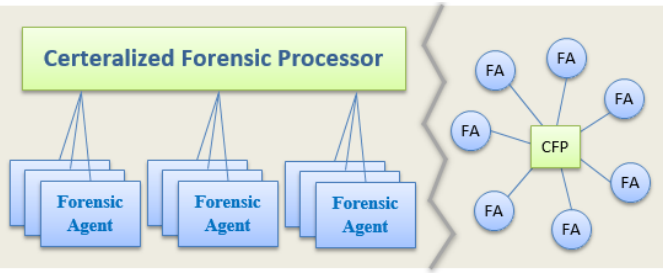


FIGURE 7: Communication Topology of CFP with Multi-agents.

Generally, there might be hundreds or even thousands of computer devices in a signal enterprise (as Fig. 7). The CFP Collector needs to simultaneously perform the information collections for many agents. This causes a multi-agent planning problem, in which the task performed to combine information from several sources, called Information Integration [5]. When multiple

agents working in the environment to collect information, it raises a need to ensure each agent can achieve its own goal with the help or hindrance of others [6]. When communication constrains exist, although the planning phase is a centralized process, the execution phase may need to be decentralized or at least partial decoupled. In such a case, explicit communicative instructions are needed for organizing the operations of multiple agents to achieve the goal.

For the security purpose that the evidence of an agent should not be contaminated by another, the communication requirement between the agents are prohibited. In such a case, the communication between CFP and the agents form a Star network topology. However, as the enterprise will recruit new computer devices from time to time, changing of the edges causes the topology to be dynamic. The edge set of the network is time varying in which edges may disappear and reappear in accordance with the changing state o network agents [7]. Such situation can be represented as:

$$V = \{V_1, V_2, \ldots, V_n\}$$

In the representation, V denotes vertex set in underlying context and the set is consisted of n elements. An analysis strategy is needed to ensure the information collecting can go through without interruption when new elements join the set. Lyapunov theory is an intuitive framework for the analyzing asymptotic properties of dynamical systems, which provides a viable solution to such kind of problems [7]. The theory treats the system as a graph. When the edges of the system change, it rebuilds the graph without interrupting analysis of the graph. The Edge Agreement Protocol proves that a connected graph with changed edges can steer the edge states to the origin. Such graph has the following relationship：

$$X_e(t) = -L_e(G)X_e(t)$$

In above representation, G denotes a graph with n nodes and m edges, $X_e(t) \in R^m$ represents the internode states, and $L_e(G)$ is a lieu of the vertex-to-edge transformation induced by the incidence matrix of G. As our system uses a design of Star topology, we do not need to worry about that circles in the graph will prevent the agreement state to be reached [7]. The integrated state information for all the network nodes can be calculated by repeatedly applying the Lyapunov equation on each element node. When there is a new node appended to the network, the state information for the network can be updated by applying the Lyapunov equation on the additional node.

After the information has been integrated, the uncertain information needs to be further processed, aiming to analyze uncertainty and derive the meaning of the information. The task to achieve this purpose is called Data Fusion [8]. As the information of collected evidence is subject to the analysis difficulties of incompleteness, imprecision, and uncertainty, the core of the analysis is to find the probabilities behind the information. These output probabilities provide a support to decision makings and court judgments, and thus for such purposes, the generation of these probabilities is required to be representable. Some handling approaches are building probabilities based on belief measures. These approaches cause the analysis results difficult to be represented. For instance, the inference of Bayesian Theory under the assumption that a and b are disjoint propositions can be expressed as:

$$P(a + b) = P(a) + P(b) + P(a \cup b)$$

Such inference cannot distinguish between lack of belief and disbelief, and disallows to withhold belief from a proposition before the negation of the proposition belief [9]. In contrast, the approaches based on plausibility measures can easily represent the analysis result, and hence they are encouraged to be used in the forensic context.

Possibilistic Logic (PL) is a viable approach developed from possibility theory that handles uncertainty in a logical setting. PL measures probability by classical logic formulae associated with weights of necessity degrees and such measuring is inconsistency-tolerant [10]. A first-order PL is basically a pair (P,α) made of a classical first order logic formula (P) and a weight expressing (α) certainty or priority. The inference rules can be expressed as:

$$(\neg P \vee Q, \alpha): (P, \beta) \vdash (Q, \min(\alpha, \beta))$$

Such inference can be improved to deal with inconsistency and be applied to derive implicit relationships from PL knowledge bases [11]. Such knowledge bases are expected to be used to generate reports by the Reporter of CFP and retained as supporting evidences for the reports.

5. USING NETWORK CENTRIC FORENSIC SYSTEM

NCEFS can be utilized for various investigation purposes including security compliance assurance, employee behavior monitoring and incident investigation, etc. The GTMC, a large motor vehicle manufacturer in China, is establishing and expanding its internal information control and forensic infrastructure referring to this concept. Based on the nature of the investigations, the usages of NCEFS can be divided into the following two categories.

5.1 Regular Investigation

Regular Investigation is a circling process needed for assuring certain activities have happened or have not happened in a timely manner. For example, if an enterprise has established some security agreement that prohibits employees from installing social applications on the office computers, regular investigations are required to examine the office computers to assure the employees are complying with the security agreement. The regular investigation usually is only interested in certain aspects of the devices, such as network logs or installed software list etc. However, if all investigations are treated equally and for every single investigation, the devices acquire the full dump of disk and memory, it will heavily increase the burden of the systems and networks of the enterprise, and make the investigations inefficient. For this type of investigations, not all available information of the devices is required to be resented. CFP Planner of NCEFS will request the CFP Collector only to acquire relevant information to avoid redundant processing and shorten the analysis time.

5.2 Incident Response Investigation

When an information security incident happens and have been aware by the enterprise, investigations must be carried out to probe out when, where, why, and with whom did the incident occur [12]. In this situation, as the scale and influence is unknown at the beginning when an incident is just detected, investigator must collect as much information as available to discover the facts and consequences about the incident, then seek for evidences that can prove the crime and the wrecker. Under this circumstance, in addition to the permanent data storage of the victim device, live response in terms of information on current network connections, running processes, open files and other artefacts must be collected immediately. When an incident happens, NCEFS will request Forensic Agent to lock down the victim device and cease its normal operation, and start to collect data on its permanent storage components and information of live response. NCEFS will not release the lock of the device until the data collection has been finished, in order to guarantee the digital evidences not to be dropped or contaminated.

6. CONCLUSION

This research explored the potential value of integrating conventional forensic tools and manual processes into a systematic and automated forensic system in enterprise context. The automated forensic processing is introduced to reduce the operational mistakes and increase the efficiency. An enhanced forensic workflow has been proposed to prevent the negligence and ignorance upon the essential procedures of investigation during the planning, acquiring, analyzing and reporting stages.

This research promotes the utilization of modern network distributed system concept and information fusion theory in the implementation of forensic. By the integration of network distributed system with forensic techniques, the investigation can become more efficient and agile in response to incidents. The idea of implementing forensic system with network distribution and information fusion concept encourages innovative forensic practices in terms of intelligent forensic planning, remote evidence collection, and comprehensive information analysis. Such innovative forensic approaches are expected to overcome the weaknesses of the traditional forensic techniques, reduce redundant processing, and render more robust forensic processes. In future, research effort will be spent on improving the incident analytic algorithm, so that more accurate forensic reports can be produced.

7. REFERENCES

[1] Naqvi, S., Dallons, G., & Ponsard, C. (2010). "Protecting Corporate ICT Infrastructures by using Digital Forensics". IEEE, 255-258.

[2] "The Computer Forensic Process an Overview". (n.d.). (Gobal Digital Forensics) Retrieved from Gobal Digital Forensics: http://evestigate.com/the-computer-forensic-process-an-overview

[3] Sivaprasad, A., & Jangale, S. (2012). "A Complete Study on Tools and Techniques for Digital Forensic Analysis". IEEE, 881-886.

[4] Edwards, G., & Chan, P. (2010). "First Draft of our Forensic Workflow". Retrieved from Born-Digital Program @ Stanford University Libraries: http://lib.stanford.edu/digital-forensics-stanford-university-libraries/first-draft-our-forensic-workflow

[5] Torra, V., & Narukawa, Y. (1998). Modeling Decisions - Information Fusion and Aggregation Operators. Springer.

[6] Russell, S. J., & Norvig, P. (2009). AI: A Modern Approach 3rd. Prentice Hall.

[7] Mesbahi, M., & Egerstedt, M. (2010). "Graph Theoretic Methods in Multiagent Networks". Princeton University Press.

[8] Shahbazian, E., Rogova, G., & Valin, P. (2005). Data Fusion for Situation Monitoring, Incident Detection, Alert and Response Management. IOS Press.

[9] Klein, L. A. (2004). Sensor and Data Fusion - A Tool for Information Assessment and Decision Making. SPIE.

[10] Das, S. (2008). High-Level Data Fusion. Artech House Inc.

[11] Dubois, D., & Prade, H. (2003). "Possibilistic Logic: a Retrospective and Prospective View". Elsevier, 3-22.

[12] Nolan, R., O'Sullivan, C., & Branson, J. (2005). "First Responders Guide to Computer Forensics". CMU.

[13] "Digital Data Acquisition Tool Test Assertions and Test Plan". (2005). NIST, 1-47.

[14] EC-Council. (2009). Computer Forensics Investigating Data and Image Files. EC-Council Press.

[15] EC-Council. (2009). Computer Forensics Investigating Network Intrusions and Cyber Crime. EC-Council Press.

[16] EC-Council. (2009). Computer Forensics Investigating Wireless Networks and Devices. EC-Council Press.

[17] Hunt, R., & Slay, J. (2010). "Achieving Critical Infrastructure Protection through the Interaction of Computer Security and Network Forensics". IEEE, 23-30.

[18] Hunt, R., & Slay, J. (2010). "The Design of Real-Time Adaptive Forensically Sound Secure Critical Infrastructure". IEEE, 328-333.

[19] Kubi, A. K., Saleem, S., & Popov, O. (2011). "Evaluation of Some Tools for Extracting e-Evidence from mobile Devices". IEEE, 1-6.

[20] Marturana, F., Me, G., Berte, R., & Tacconi, S. (2011). "A Quantitative Approach to Triaging in Mobile Forensics". IEEE, 582-588.

[21] Meghanathan, N., Allam, S. R., & Moore, L. A. (2009). "Tools and Techniques for Network Forensics". IJNSA, 1004.0570.

[22] Naqvi, S., Dallons, G., & Ponsard, C. (2010). "Applying Digital Forensics in the Future Internet Enterprise Systems - European SMEs' Perspective". IEEE, 89-93.

[23] Philipp, A., Cowen, D., & Davis, C. (2009). Hacking Expose Computer Forensics. McGraw Hill.

[24] Pladna, B. (2008). "Computer Forensics Procedures, Tools, and Digital Evidence Bags: What They Are and Who Should Use Them". East Carolina University.

[25] Thing, V. L., Chua, T.-W., & Cheong, M.-L. (2011). "Design of a Digital Forensics Evidence Reconstruction System for Complex and Obscure Fragmented File Carving". IEEE, 793-797.

[26] Vacca, J. R. (2005). Computer Forensics Computer Crime Scene Investigation 2ed. Charles River Media.

Cryptography Based MSLDIP Watermarking Algorithm

Abdelmgeid A. Ali *abdelmgeid@yahoo.com*
Faculty of Science, Computer Science Department
Minia University
Minia, 61519, Egypt

Ahmed H. Ismail *ahamdycs2012@gmail.com*
Faculty of Science, Computer Science Department
Minia University
Minia, 61519, Egypt

Abstract

In recent years, internet revolution resulted in an explosive growth in multimedia applications. The rapid advancement of internet has made it easier to send the data accurate and faster to the destination. Aside to this, it is easier to modify and misuse the valuable information through hacking at the same time. Digital watermarking is one of the proposed solutions for copyright protection of multimedia data. In this paper cryptography based MSLDIP watermarking method (Modified Substitute Last Digit in Pixel) is proposed. The main goal of this method is to increase the security of the MSLDIP technique besides to hiding the watermark in the pixels of digital image in such a manner that the human visual system is not able to differentiate between the cover image and the watermarked image. Also the experimental results showed that this method can be used effectively in the field of watermarking.

Keywords: Cryptography, Encryption, Decryption, Watermarking, Spatial Domain, MSLDIP (Modified Substitute Last Digit in Pixel), Security.

1. INTRODUCTION

Information hiding techniques have recently become important in a number of application areas [1]. The term hiding here can refer to either making the information imperceptible (as in watermarking) or keeping the existence of the information secret (as in steganography) [2]. Information hiding means communication of information by hiding in and retrieving from any digital media. The digital media can be an image, an audio, a video or simply a plain text file. Information hiding is a general term encompassing many sub disciplines. However, generally it encompasses three disciplines: cryptography, watermarking, and steganography [3, 4]. It is graphically shown in (Figure 1.1), Watermarking can be robust or fragile depending upon the application domain.

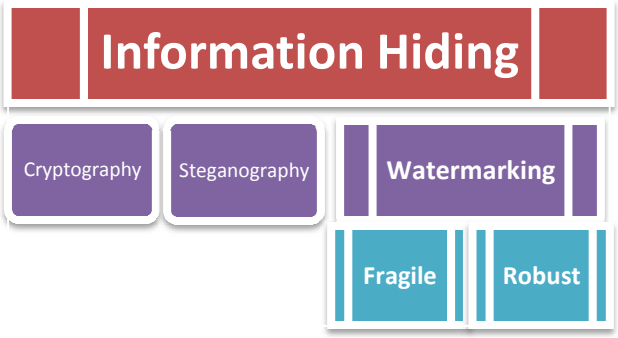


FIGURE 1.1: Information Hiding Disciplines.

Cryptography is an area within the field of cryptology. The name cryptology is a combination of the Greek (cruptos = hidden and logos = study, science). Therefore, the word cryptology literally implies the science of concealing. The cryptography can be divided into two areas: cryptography and cryptanalysis [5]. Cryptanalysis is the area within cryptology which is concerned with techniques for deciphering encrypted data without prior knowledge of which key has been used. This more commonly known as 'Hacking'. The cryptanalyst is the person who tries to find weaknesses in encryption schemes. He will often figure out how to break the cryptography scheme, and then the developer of the scheme will use that information to make it stronger [6].

When people initially tried to communicate over distances, they tried to ensure the secrecy of their communications. The technology of steganography is developed for this goal [7]. The word steganography comes from two Greek words stegauw (steganos) and grafein (graphein) meaning covered writing. It is basically about embedding a secret message in a cover file [8] which looks innocuous. This cover file could be an image file, video file, audio file, text file, or any computer code [4, 9]. Steganography is comprised of two algorithms, one for embedding and one for extraction [10]. A great deal of attention is paid to ensuring that the secret message goes unnoticed if a third party were to intercept the cover file [11].

Watermarking is a technique used to hide data or identifying information within digital multimedia. The discussion will focus primarily on the watermarking of digital images, though digital video, audio, and documents are also routinely watermarked. Digital watermarking is becoming popular, especially for adding undetectable identifying marks, such as author or copyright information. The digital watermarking process embeds a signal into the media without significantly degrading its visual quality. Digital watermarking is a process to embed some information called watermark into different kinds of media called Cover Work [3, 12]. Digital watermarking is used to hide the information inside a signal, which cannot be easily extracted by the third party. Watermarking is used for following reasons, Proof of Ownership (copyrights and IP protection), Copying Prevention, Broadcast Monitoring, Authentication, Data Hiding. Digital watermark is an important research direction for the technique of information hiding, mainly including the characteristics (capacity, invisibility, security, robustness).

2. RELATED WORK

In this section, the MSLDIP Watermarking algorithm will be presented which works on the spatial domain of the cover image. At the first the (SLDIP) will be presented before the (MSLDIP) method. SLDIP method takes the cover image and the watermark as input, convert the blue layer of the cover image into one row, and divide the row into blocks each of which contains 9 values, then consider the watermark is color image then each pixel will be represented in 3 bytes, according to the color image representation (which each pixel is specified by three values one each for red, blue, and green components of the pixel's color and each value represented by one byte, so each pixel will be represented in three bytes) [13].

Each byte in the watermark image will be ranges from 0 to 255, and make each byte value's length equal to 3 digits, for example we have byte of value 15, this value equal to 015 which has length of 3 digits, finally substitute each 9 digits of each pixel with the last digit of each pixel in the current block, so each pixel of the watermark image will be embedded in only one block, and output the watermarked image and 2 keys which be required in the extraction process (Figure 2.1) and (Figure 2.2) [14].

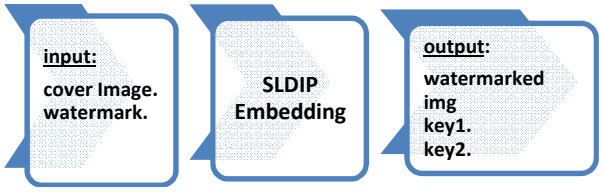


FIGURE 2.1: SLDIP Watermarking Embedding Process [14].

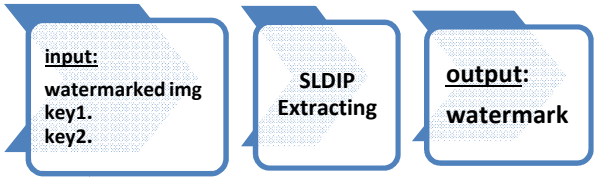


FIGURE 2.2: SLDIP Watermarking Extraction Process [14].

Assuming that watermark image of height 1 pixels and width 1 pixels, and cover image of height 3 pixels and width 3 pixels. The SLDIP will represent the cover image in one row which contains one block of 9 values (125, 255, 086, 192, 145, 210, 035, 099, and 004), and the watermark image will be represented as (230, 105, and 080), the SLDIP will substitute 5 (last digit in cover image) in 125 with 2 (first digit in watermark image) in 230, the result is 122 and also substitutions done until reaching the last digit in the last value of the watermark. The watermarked image will be (122, 253, 080, 195, 140, 215, 030, 098, and 000) [13].

By using this method capacity of embedding has been increased, the maximum area of watermark image that can be embedded in any cover image can be calculated by using this formula [14]:

$$\frac{ImageWidth \times ImageHeight}{9} = watermark_{area}\ (1) \quad [14]$$

If the watermark image is grayscale image this formula can be used:

$$\frac{ImageWidth \times ImageHeight}{3} = watermark_{area}\ (2) \quad [14]$$

Supposing a (8 x 8) cover image, by using equation 1, we can embed colored watermark image of area 7 pixel2 which approximately equals to (2 x 3) colored watermark image, and by using equation 2, a grayscale watermark image of area 21 pixel2 which approximately equals to (4 x 5) grayscale watermark image can be embedded. Notice that SLDIP uses only one layer of the color image neither two nor three layers. This means that we can use this method in color and grayscale images [14].

MSLDIP is a modification on SLDIP by update the substitution step to decrease the difference between the original pixel and the substituted pixel, for example embedding value digit 9 in pixel 100, by using SLDIP the pixel will be 109, but by MSLDIP two possible values can be taken for each substitution and choose the value that has the smallest difference, so the two values will be 109 and 99, then the value with the smallest difference must be chosen, so the pixel value will be 99, the difference will be 1 instead of 9 and this increases the PSNR value of the image [14].

3. PROPSED METHOD

In this section the proposed method will be presented, at the first the proposed method will be divided into two algorithms which are Watermark embedding algorithm and watermark extraction algorithm.

3.1 Watermark Embedding Algorithm

Algorithm: Secured MSLDIP Embedding Algorithm.
Input: Watermark W; Cover Image C; Secret Key K.
Output: Encrypted Watermark W', Secured Watermarked Image SWI.
Steps: (Figure 3.1)

1. Take W and encrypt it by performing RC4 Encryption algorithm with K, the output of this step is called W'.
2. Apply MSLDIP Watermarking Embedding procedure to embed W' in C, the output of this step is called secured watermarked image SWI.

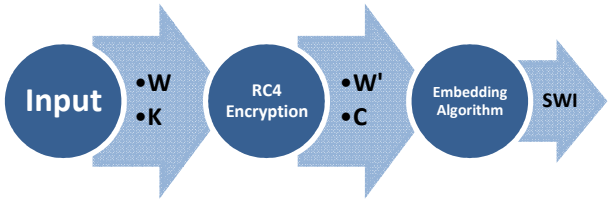


FIGURE 3.1: Modified MSLDIP Embedding Process.

3.2 Watermark Extraction Algorithm

Algorithm: Secured MSLDIP Extraction Algorithm.
Input: Secured Watermarked Image SWI; Secret Key K.
Output: Encrypted Watermark W', Watermark W.
Steps: (Figure 3.2)

1. Apply procedure MSLDIP extraction to extract the encrypted watermark from SWI, the output of this step is called W'.
2. Take W' and decrypt it by performing RC4 Decryption algorithm using K, the output of this step is called W.

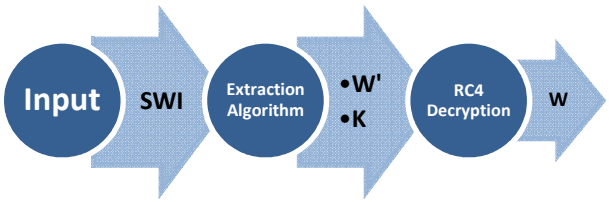


FIGURE 3.2: Modified MSLDIP Extraction Process.

4. EXPERIMENTAL RESULTS

In order to evaluate the performance of the watermarked images, there are some quality measures such as PSNR and MSE.

The **MSE (Mean Square Root)** is defined as an average squared difference between a reference image and a distorted image. It can be calculated by the formula given below

$$MSE = \frac{1}{XY}\left[\sum_{i=1}^{X}\sum_{j=1}^{Y}(c(i,\ j) - e(i,\ j))^2\right] \quad (3)$$

Where X and Y are height and width respectively of the cover image, the c(i, j) is the pixel value of the cover image and e(i, j) is the pixel value of the watermarked image.

The **PSNR (Peak Signal to Noise Ratio)** is an expression for the ratio between the maximum possible value (power) of a signal and the power of distorting noise that affects the quality of its representation. It can be calculated by the formula as

$$PSNR = 10\ log_{10}\left(\frac{L \times L}{MSE}\right) \quad (4)$$

Where L is the peak signal value of the cover image which is equal to 255 for 8 bit images [15].

In order to make the watermarking algorithm more secured, RC4 Encryption Algorithm is merged with the MSLDIP Embedding Algorithm (Figure 4.1), and in the other hand RC4 Decryption Algorithm is merged with the MSLDIP Extraction Algorithm as anyone can know the MSLDIP watermarking algorithm and do the reverse of embedding algorithm and so the watermark can be

known, but by using an encryption algorithm the user who doing the reverse of the MSLDIP embedding algorithm must know the key to extract the correct watermark, if the key has been entered is incorrect the watermark will be fake (Figure 4.2).

Modified MSLDIP have been implemented in MATLAB 2014 platform and the experiment has been conducted on various images.

FIGURE 4.1: Embedding a, b using Modified MSLDIP Watermarking to output the watermarked image c.

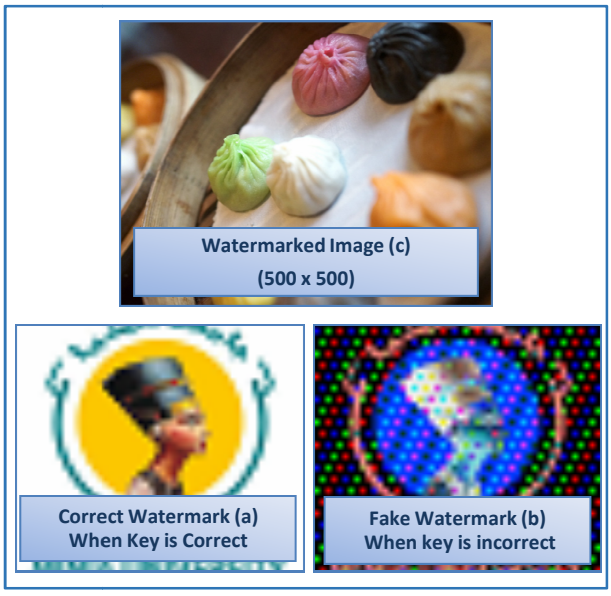


FIGURE 4.2: Watermark Extraction using Modified MSLDIP when input can be correct key and incorrect key.

Modified MSLDIP Watermarking has been applied on set of images different in sizes and the Peak Signal to Noise Ratio (PSNR), and Mean Square Root (MSE) have been calculated, all results recorded in (Table 1).

Cover Image	PSNR	MSE	Watermark Image
(150 x 150)	43.21	3.11	(50 x 50)
(200 x 200)	45.87	1.68	(50 x 50)
(500 x 500)	53.29	0.31	(50 x 50)
(700 x 700)	56.22	0.16	(50 x 50)
(1000 x 1000)	59.82	0.07	(50 x 50)

TABLE 1: Results of applying Modified MSLDIP watermarking on various sizes images.

In (Figure 4.3), Chart showing the results of MSE and PSNR between the cover image and watermarked image in the Modified MSLDIP Algorithm.

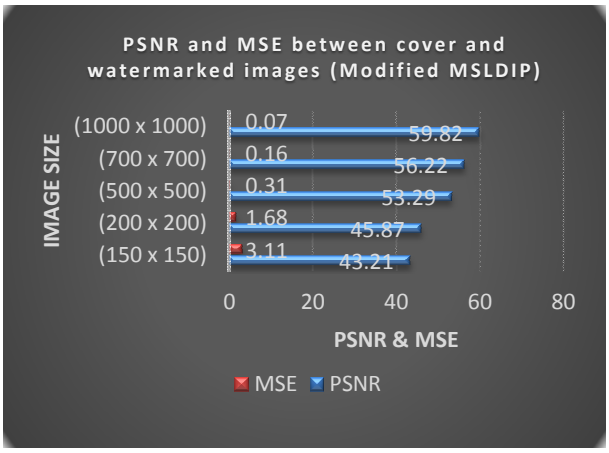


FIGURE 4.3: Chart showing the results of Modified MSLDIP.

Modified MSLDIP has been compared with [16] (Table 2), supposed four cover image with squared sizes 128, 256, 512, and 1024, and a watermark with full capacity with cover images according to [16], thus the full watermark capacity can be calculated using cover image sizes according to [16] by using formula

$$Round\ up\left(\sqrt{\frac{ImageWidth\ \times ImageHeight}{24}}\right) = watermark\ side\ length \qquad (5)$$

Then Modified MSLDIP has been compared with [14] (the previous version method) (Table 3), the results have been approximately equal to each other but in the modified MSLDIP the user who extracting the watermark from the watermarked image must have the key to get the correct watermark if not the watermark which has been extracted from the watermarked image will be fake, thus the Modified MSLDIP can be better than MSLDIP as the first one is more secured than the other.

Finally the Modified MSLDIP has been compared with [17], supposed the grayscale baboon.bmp as a cover image and, the grayscales lena.bmp, and barbara.bmp as watermark and the full capacities of the embedded watermark according to each algorithm have been calculated using formulas

$$Round\ up\left(\sqrt{\frac{CoverImageWidth \times CoverImageHeight}{3}}\right) = watermark\ side\ length\ (Modified\ MSLDIP) \quad (6)$$

$$Round\ up\left(\sqrt{\frac{CoverImageWidth \times CoverImageHeight}{8}}\right) = watermark\ side\ length\ of\ [14] \quad (7)$$

Cover Image	Watermark FC [10]	[16] 3rd Bit		Modified MSLDIP	
		PSNR	MSE	PSNR	MSE
Baboon 128	(27 x 27)	31.68 dB	44.50	46.90 dB	1.33
Bird 256	(53 x 53)	31.68 dB	44.50	46.99 dB	1.30
Boat 512	(125 x 125)	31.68 dB	44.50	46.10 dB	1.60
Pepper 1024	(210 x 210)	31.68 dB	44.50	47.17 dB	1.25

TABLE 2: Results Comparison between [16] 3rd Bit and Modified MSLDIP.

Cover Image	Watermark Image	[14] MSLDIP		Modified MSLDIP	
		PSNR	MSE	PSNR	MSE
(600 x 600)	(200 x 200)	43.10 dB	3.18	43.06 dB	3.22
(768 x 768)	(200 x 200)	45.66 dB	1.77	45.63 dB	1.78
(1024 x 1024)	(200 x 200)	48.23 dB	0.98	48.22 dB	0.98
(1280 x 1280)	(200 x 200)	49.69 dB	0.67	49.46 dB	0.74
(1500 x 1500)	(200 x 200)	51.29 dB	0.48	51.19 dB	0.49

TABLE 3: Results Comparison between [14] MSLDIP and Modified MSLDIP.

Cover Image	Watermark Image	[17]	Modified MSLDIP
		PSNR	PSNR
baboon.bmp (512 x 512)	lena.bmp (64 x 64)	58.64 dB	52.00 dB
baboon.bmp (512 x 512)	Barbara.bmp (64 x 64)	58.99 dB	51.81 dB
Watermark (FC) applying equations (6,7) With Cover Image size (512 x 512)		(181 x 181)	(295 x 295)

TABLE 4: Results Comparison between [17] and Modified MSLDIP.

5. COMPARATIVE EVALUATION

From the comparison in table (2), the reason of why Modified MSLDIP has been compared with [16] 3rd Bit? Has been clarified as, in Modified MSLDIP substitutions can change the value of pixel which the difference ranges from 0 to 5 and change in the 3rd Bit in pixel can change the value of pixel which the difference ranges from 0 to 7 which include the Modified MSLDIP difference range. However results in Modified MSLDIP are better.

From the comparison in table (3), the results of the modified MSLDIP are compared with the results of the MSLDIP [14] (the previous version), and It can be concluded that the results were very close, as the difference didn't not exceed the one after the decimal point, but in the modified MSLDIP the data which has been watermarked is more secured with a key, it can be proved that the modified MSLDIP technique is better.

From the comparison in table (4), the results of modified MSLDIP are compared with results of [17], and from the comparison it can be concluded that the two algorithm have very good PSNR results that mean no one can discover the watermark when looking at the image, also it can be conducted that the watermark full capacity of modified MSLDIP is greater than [17], suppose cover image (512 x 512) and watermark (256 x 256) algorithm of [17] cannot embed the watermark in cover image but the modified MSLDIP can embed this watermark successfully.

After Implementing and analyzing the results, conclude that, the visual quality of the image doesn't change significantly, on the other hand this algorithm is more robust than LSB technique [17], because in LSB technique some attackers can possibly zero out several least significant bit of pixels of the image and hence clear the watermark. This technique has increased the capacity of watermark in embedding process.

6. CRITICAL DISCUSSION

Watermarking algorithm proposed in [16], the full capacity of watermark which can be embedded in cover image of size 128x128 pixels is 682.67 px^2 that approximately equal to watermark of size 26x26 pixels however in our proposed method the full capacity of watermark is 1280.44 px^2 that approximately equal watermark of size 42x42 pixels, that mean our proposed method can embed watermark with capacity larger than [16], also in [16] there isn't any way to prevent unauthorized users from accessing the watermark, however in our proposed method the watermark is encrypted using RC4 with a key which only users who have this key can access the watermark, that mean in our proposed method the watermark is more secured.

Watermarking algorithm proposed in [14], regardless of it can embed watermark with capacity equal to our proposed method, in [14] there isn't any way to prevent unauthorized users from accessing the watermark, however in our proposed method the watermark is encrypted using RC4 with a key which only users who have this key can access the watermark that mean in our proposed method the watermark is more secured.

Watermarking algorithm proposed in [17], the full capacity of watermark which can be embedded in cover image of size 512x512 pixels is 32768 px^2 that approximately equal to watermark of size 181x181 pixels however in our proposed method the full capacity of watermark is 87381.33 px^2 that approximately equal watermark of size 295x295 pixels, that mean our proposed method can embed watermark with capacity larger than [17], also in [17] there isn't any way to prevent unauthorized users from accessing the watermark, however in our proposed method the watermark is encrypted using RC4 with a key which only users who have this key can access the watermark, that mean in our proposed method the watermark is more secured.

7. CONCLUSION

Digital watermarking with cryptography is the current area of research where lot of scope exists. Currently digital watermarking with cryptographic technique is being used by several countries for secretly transfer of hand written documents, text images, financial documents, internet voting etc.

This paper starts from some basic knowledge of information hiding categories includes digital image watermarking, and from results conclude that the visual quality of the image doesn't change significantly, this algorithm is more robust than LSB technique because in LSB technique some attackers can possibly zero out several least significant bit of pixels of the image and hence clear the watermark, this algorithm is more secure because of using cryptographic technique which has been merged with the watermarking technique, and this technique has increased the capacity of watermark which will be embedded. In the future, the aim of this paper is to extend the cryptography to higher dimensions and apply it in frequency domain in order to consider more security and robustness.

8. REFERENCES

[1] F. A. P. Petitcolas, R. J. Anderson and M. G. Kuhn, "Information Hiding – A Survey", proceedings of the IEEE, Special Issue on Protection of Multimedia Content, vol. 87(7), pp. 1062 - 1078, July 1999.

[2] I. J. Cox, m. L. Miller, J. A. Bloom, J. Fridrich and T. Kalker, "Digital Watermarking and Steganography", ISBN 978-0-12-372585-1, 2nd edition, Elsevier inc, 2008.

[3] Preeti Gupta, “Cryptography based digital image watermarking algorithm to increase security of watermark data”, International Journal of Scientific & Engineering Research, Volume 3, Issue 9 (September 2012) ISSN 2229-5518.

[4] J. A. Mathew, "Steganographic Techniques for Subliminal Communication in Open Systems Environment", Sam Higginbottom Institute of Agriculture, Technology and Sciences, PHD. Thesis, 2010.

[5] Jan C A, Van Der Lubbe, "Basic Methods of Cryptography", English translation Cambridge University Press 1998.

[6] E. Cole, "Hiding In Plain Sight: Steganography and The Art of Covert Communication", ISBN 0-471-44449-9, Wiley publishing, inc, 2003.

[7] S. A. Baker and Dr. A. S. Nori, "Steganography in Mobile Phone over Bluetooth", International Journal of Information Technology and Business Management (JITBM), Volume 16, Number 1, Pages 111- 117, 29 August 2013.

[8] T. Morkel, "Image Steganography Applications for Secure Communication", Master of Science (Computer Science) Thesis, Faculty of Engineering, Built Environment and Information Technology University of Pretoria, Pretoria, May 2012.

[9] F.C.Gonzalez, "Counter Terrorist Steganography Search Engine", Master of Science Thesis, Department of Aerospace, Power and Sensors, Royal Military College of Science, Shrivenham, Cranfield University, 2002.

[10] S. A. Sohag, Dr. M. K. Islam and M. B. Islam, "American Journal of Engineering Research (AJER)", Volume 2, Issue 9, Pages 118 - 126, 2013.

[11] S.Deepa and R.Umarani, "A Study on Digital Image Steganography", International Journal of Advanced Research in Computer Science and Software Engineering, International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 1, January 2013.

[12] B Surekha, Dr GN Swamy, “A Spatial Domain Public Image Watermarking”, International Journal of Security and Its Applications Vol. 5 No. 1, January, 2011M. Abdullatif, A. M. Zeki, J. Chebil, and T. S. Gunawan, "Properties of Digital Image Watermarking".

[13] Ahmed A. Radwan, Ahmed Swilem, Al-Hussien Seddik, "A High Capacity SLDIP method", ICICIS, July 2011.

[14] Abdelmgeid A. Ali, Ahmed A. Radwan, and Ahmed H. Ismail, "Digital Image Watermarking using MSLDIP (Modified Substitute Last Digit in Pixel)", IJCA, Volume 108 – No 7, Pages 30-34, December 2014.

[15] Amit Kumar Singh, Nomit Sharma, Mayank Dave, Anand Mohan, "A Novel Technique for Digital Image Watermarking in Spatial Domain", 2012 2nd IEEE International Conference on Parallel, Distributed and Grid Computing.

[16] Deepshikha Chopra, Preeti Gupta, Gaur Sanjay, Anil Gupta, "LSB based digital image watermarking for gray scale image", IOSRJCE, October 2012.

[17] Krishna Kumar, and Shashank Dwivedi, "Digital Watermarking using Asymmetric Key Cryptography and Spatial Domain Technique", IJARCSMS, Volume 2, Issue 8, August 2014.

Efficient Security Alert Management System

Minoo Deljavan Anvary *Minoo.deljavan@yahoo.com*
IT Department School of e-Learning
Shiraz University
Shiraz, Fars, Iran

Majid Ghonji Feshki *Ghonji.majid@yahoo.com*
Department of Computer Science
Qzvin Branch, Islamic Azad University
Qazvin, Qazvin, Iran

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

Abstract

Nowadays there are several security tools that used to protect computer systems, computer networks, smart devices and etc. against attackers. Intrusion detection system is one of tools used to detect attacks. Intrusion Detection Systems produces large amount of alerts, security experts could not investigate important alerts, also many of that alerts are incorrect or false positives. Alert management systems are set of approaches that used to solve this problem. In this paper a new alert management system is presented. It uses K-nearest neighbor as a core component of the system that classify generated alerts. The suggested system serves precise results against huge amount of generated alerts. Because of low classification time per each alert, the system also could be used in online systems.

Keywords: Intrusion Detection, Security Alert Management, K-nearest Neighbor, Real-time Security Alert Classification, Reduction of False Positive Alerts, Precise Classifying True Positive Alerts.

1. INTRODUCTION

An intrusion detection system (IDS) inspects all inbound and outbound network activity and identifies suspicious patterns that may indicate a network or system attack from someone attempting to break into or compromise a system [1]. There are several ways to categorize an IDS such as misuse detection vs. anomaly detection, network-based vs. host-based systems and passive system vs. reactive system. In misuse detection, the IDS analyzes the information it gathers and compares it to large databases of attack signatures. Essentially, the IDS looks for a specific attack that has already been documented. Like a virus detection system, misuse detection software is only as good as the database of attack signatures that it uses to compare packets against. In anomaly detection, the system administrator defines the baseline, or normal, state of the networks traffic load, breakdown, protocol, and typical packet size. The anomaly detector monitors network segments to compare their state to the normal baseline and look for anomalies. In a network-based system, or NIDS, the individual packets flowing through a network are analyzed. The NIDS can detect malicious packets that are designed to be overlooked by a firewalls simplistic filtering rules. In a host-based system, the IDS examines at the activity on each individual computer or host. In a passive system, the IDS detects a potential security breach, logs the information and signals an alert. In a reactive system, the IDS responds to the suspicious activity by logging off a user or by reprogramming the firewall to block network traffic from the suspected malicious source. These systems are known to generate many alerts. Analyzing these

alerts manually by security expert are time consuming, tedious and error prone. From another point of view false positive alerts have huge share of generated alerts. Identifying attack types and generating correct alerts related to attacks is another problem with IDS. In order to overcome mentioned problems alert management systems was introduced. Alert management systems help security experts to manage alerts and produce a high level view of alerts.

In this paper authors design new algorithm based on framework introduced in [2] that uses K-nearest neighbor algorithm (KNN) [3] as a core component. It classifies the generated alerts based on attack type of alerts, detects false positive alerts, high speed classification to use with alert generation in IDSs. The proposed system uses some techniques same as previous work techniques [3] such as alert filtering, alert preprocessing, and alert filtering to improve accuracy of the results.

This paper is categorized in to 5 sections. In Section 1 the alert management techniques are mentioned. Related alert management techniques are investigated in section 2, section 3 explains the suggested alert management system and describes all component of the proposed system, the experimental results are shown in section 4 and finally section 5 is a conclusion and future works.

2. RELATED WORKS

There are several techniques that used to manage security alerts one of them is clustering and classification of alerts. In [4, 5] clustering algorithms based on genetic algorithm, named Genetic Algorithm (GA) and Immune based Genetic Algorithm used to manage IDS alerts. In [6] author was introduce clustering algorithm based on root causes which finds main cause of alerts and join them together to construct clusters. He shows that by deleting these root causes consequent alerts reduced to 82%. This method is very good but it depends on undelaying network structure and the approach should be change when the structure of network is changed.

In [7] DARPA 2000 dataset [8] is evaluated by three algorithms. The proposed algorithms used alerts without any preprocessing techniques. In another work expert system used to make decision [9, 10]. Debar et al. [11] designed a system by placing them in situations aggregates alerts together. Situations are set of special alerts. Some attributes of alert are used to construct a situation.

A new alert management system is introduced in [2]. For evaluation of the system Azimi et. al. used generated alerts from DARPA 98 dataset [12]. The main unit of the system is cluster/classify unit that uses Self-Organizing Maps (SOM) [13] to cluster and classify IDS alerts. In another work, an alert management system is introduced [14] that similar to [2]. In that work usage of seven genetic clustering algorithms is evaluated. In [15] Learning Vector Quantization (LVQ) [16] is used as a classification engine. The accuracy of the suggested approach is acceptable [15].

In this paper an alert management system based on system proposed by authors in [2] is proposed that uses KNN as a tool to classify input alert vectors. The system will be able to improve accuracy of results and also to reduce the number of false positive alerts.

3. EFFICIENT SECURITY ALERT MANAGEMENT SYSTEM

New alert management system is described in this section. In this paper Snort [17] IDS is used to generate alerts from DARPA 98 dataset [12]. Figure 1 shows the proposed system. Snort is a free and open source network intrusion prevention system and network intrusion detection system created by Martin Roesch in 1998. Snort can reads binary dumped network traffic files named tcpdump of DARPA 98 dataset and analyze them. After producing alerts with snort; they are labeled, normalized, filtered, preprocessed and classified respectively. Each steps of the system is described below.

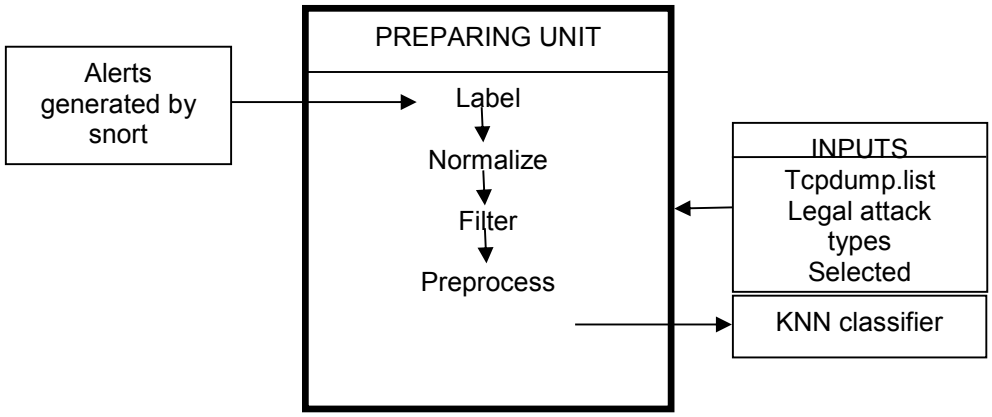


FIGURE 1: New Alert Management System.

As you see in the figure 1 preparing unit has some inputs that provide basic information about traffic and legal operations. Labeling operation according to tcpdum.list files label generated security alerts. It means that this unit appends attack type of each alert as an attribute of processed alert to proper alert. These labels are used to train and to test results of the system. Figure 2 shows the labeling algorithm.

In: TCPDUMP list files and alert log files.
Out: Labeled alerts
1. Create an empty *AttackList* and *AlertList* set.
2. For each row in TCPDUMP list files:
   2.1. If the row is a labeled attack then add the row to the *AttackList* set.
3. For each row in alert log files:
   3.1. Create key with the five attributes: *source ip, destination ip, source port, destination port, ICMP code/type.*
   3.2. 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.
   3.3. Add the selected row to the *AlertList* set.
4. Return the *AlertList* set.

FIGURE 2: Alert Labeling Algorithm [2, 14].

After labeling snort alerts they should be normalized. It means that alerts with unexpected attack types should be removed. According to [18] snort is unable to detect all attacks in DARPA 98 dataset then we use “alert types” file to specify accepted attack types in this investigation [2, 14]. In this paper accepted alerts attack types are: BACK, LAND, POD, PHF, ROOTKIT, NMAP, IMAP, and DICT. After normalizing, redundant alerts are removed and only one instance of them are remained in the final dataset.

Preprocessing operation convert string data types such as protocols and IP values to numerical one, and also transform their values to unit range. The formula to convert IP values and protocol values are in equation (1) and (2) respectively. There are two methods to reduce value range attributes named Unit Range (UR) and Improved Unit Range (IUR). As described in [2] the accuracy of IUR is better than UR method, so in this paper IUR method is used.

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

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

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

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

In this unit KNN algorithm is used as a classifier. KNN is one of famous and traditional classification algorithms [3]. It widely used in pattern recognition. It determines class type of input vectors according to its neighbors. So some data should be entered as a training data set and then test data set is entered to classification purpose. KNN has two steps. In first step when an input vector entered to system for classification, k nearest data vectors to input vector from training data set is selected. In the second step class of input vector according to selected data vectors from previous step is calculated. In this paper we use Euclidean Distance as a distance metric [3].

4. EXPERIMENTAL RESULTS

To simulate the new system C#.net programming language, MATLAB software and SPRTool toolbox is used [19, 20]. The parameters of simulation are shown below.

In the investigation the number of nearest neighbor (K) is in range of [1 100]. The attack types used in this simulation are: BACK, POD, IMAP, NMAP, DICT, ROOTKIT, PHF and LAND. Total number of alert vectors is 14279 that splits into two sets named training and testing datasets that contains 70%, 30% of totals alert data vectors respectively. About 30% of alerts in each dataset is false positives.

The result of the proposed system is evaluated by four measurement named Classification Error (ClaE), Classification Accuracy percent (ClaAR), Average Alert Classification Time (AACT) and False Positive Reduction Rate (FPRR).

As you can see in table 1 the classification error rate reduces by reducing the value of K. When K is 1 the best result is achieved. It means that one neighbor can describe one point well.

In table 1 value of these metrics are shown. The best values of ClaE and ClaAR are 8 and 99.82% respectively. The value of AACT measurement is 0.006623 that shows the proposed system can be used in active IDS alert management systems that evaluate alerts beside alert production by IDS concurrently.

Table 2 shows the results of accuracy of proposed system in identifying attack type of each alert vector in test phase. As it can be seen in table 2, the proposed system can identify all of attack types of alerts with high rate of accuracy when k is 1. Table 2 shows that when value of K is decreased the accuracy of the metrics is increased.

An important point is accuracy percent of false positive identification. That is the proposed system can reduce false positive alerts with 99.94 percent. Which shows to be a solution of an important problem of IDSs. Proposed alert management system reaches 100 percent for BACK, LAND, POD, DICT and NMAP attack types. For attack types PHF, IMAP and ROOTKIT accuracy percent values are 66.67, 66.67 and 28.57 respectively.

Table 3 shows the results of approach [14]. It is obvious that the proposed framework gets better accuracy than genetic clustering based approach. Also the result of the system is better than its

base framework based on SOM in [2]. Table 4 shows the result of SOM based framework. One of the benefits of proposed system is capability of working in real-time mode.

K	ClaE	ClaAR	AACT
100	119	97.27	0.014054
50	83	98.09	0.007908
25	65	98.51	0.007699
20	61	98.60	0.007911
15	49	98.87	0.007100
10	46	98.94	0.006864
5	22	99.49	0.006135
1	8	99.82	0.006623

TABLE 1: Extracted performance metric values from simulation.

Back	Land	Pod	Phf	Rootkit	Imap	Dict	Nmap	False Positive(FPRR)	K
99.69	0	89.80	0	0	0	100	85.68	98.47	100
99.69	0	93.88	0	0	0	100	89.15	99.49	50
99.84	100	97.96	0	0	0	100	90.89	99.60	25
99.84	100	97.96	0	0	0	100	91.76	99.60	20
99.84	100	97.96	0	0	0	100	94.14	99.69	15
99.84	100	97.96	0	0	0	100	94.36	99.77	10
99.92	100	97.96	66.67	12.29	33.34	100	98.26	99.83	5
100	100	100	66.67	28.57	66.67	100	100	99.94	1

TABLE 2: Proposed system accuracy percent for each attack type of alerts.

Bahrbegi et. al. in [14] proposed a framework that uses genetic algorithm families to clustering and classification propose. As two works are similar we have to compare our results with their work. These results are shown in table 3. For all metrics the proposed system has high value in contrast of all GA based techniques. As shown in table 3, these algorithms could not be able to work actively because of the execution times are high. Although the proposed method reaches high accuracy results per alert attack types.

Algorithm	ClaE	ClaAR	FPRR	AACT
GA	1218	72.03	52.15	Offline
FGKA	314	92.79	97.51	Offline
GKA	1011	75.2	62.11	Offline
IGA	306	92.97	95.24	Offline
GFCMA	148	96.60	97.51	Offline
GPCMA	91	97.91	96.03	Offline
GFPCMA	148	96.60	97.51	Offline

TABLE 3: Results of GA-Based Algorithms [14].

ClaE	ClaAR	FPRR	AACT
33	99.36	99.71	0.003

TABLE 4: Results SOM based Algorithms [2].

5. CONCLUSION AND FUTURE WORKS

In this paper a fast and accurate algorithm is proposed that is used KNN algorithm as its classification engine manage IDS alerts. The results show that accuracy and false positive reduction rate of the system is high. Also the system is able to identify the attack types of the

alerts more accurate in real-time manner. Using other artificial intelligence techniques such as evolutionary algorithms and decision trees to improve the accuracy of the system are future works of this paper.

6. REFERENCES

[1] Debar, H., M. Dacier, and A. Wespi, *Towards a taxonomy of intrusion-detection systems.* Computer Networks, 1999. **31**(8): p. 805-822.

[2] Ahrabi, A.A.A., et al., *A New System for Clustering and Classification of Intrusion Detection System Alerts Using Self-Organizing Maps.* International Journal of Computer Science and Security (IJCSS), 2011. **4**(6): p. 589.

[3] Cover, T. and P. Hart, *Nearest neighbor pattern classification.* Information Theory, IEEE Transactions on, 1967. **13**(1): p. 21-27.

[4] Wang, J., H. Wang, and G. Zhao. *A GA-based Solution to an NP-hard Problem of Clustering Security Events*. 2006. IEEE.

[5] Wang, J. and B. Cui. *Clustering IDS Alarms with an IGA-based Approach*. 2009. IEEE.

[6] Julisch, K., *Clustering intrusion detection alarms to support root cause analysis.* ACM Transactions on Information and System Security (TISSEC), 2003. **6**(4): p. 443-471.

[7] Maheyzah, S.Z., *Intelligent alert clustering model for network intrusion analysis.* Journal in Advances Soft Computing and Its Applications (IJSCA), 2009. **1**(1): p. 33-48.

[8] *DARPA 2000 Intrusion Detection Evaluation Datasets*, M.L. Lab., Editor. 2000.

[9] Cuppens, F. *Managing alerts in a multi-intrusion detection environment*. 2001.

[10] MIRADOR, E. *Mirador: a cooperative approach of IDS*. in *European Symposium on Research in Computer Security (ESORICS)*. 2000. Toulouse, France.

[11] Debar, H. and A. Wespi. *Aggregation and Correlation of Intrusion-Detection Alerts*. in *Proceedings of the 4th International Symposium on Recent Advances in Intrusion Detection*. 2001.

[12] *DARPA 1998 Intrusion Detection Evaluation Datasets*, M.L. Lab., Editor. 1998.

[13] Kohonen, T., *Self-Organized Maps*. 1997, Science Berlin Heidelberg: Springer series in information.

[14] Bahrbegi, H., et al. *A new system to evaluate GA-based clustering algorithms in Intrusion Detection alert management system*. 2010. IEEE.

[15] Ahrabi, A.A.A., et al., *Using Learning Vector Quantization in IDS Alert Management System.* International Journal of Computer Science and Security (IJCSS), 2012. **6**(2): p. 1-7.

[16] Kohonen, T., *Learning vector quantization, in M.A. Arbib (ed.), The Handbook of Brain Theory and Beural Networks*. 1995: MIT Press.

[17] Snort, *The open source network intrusion detection system*. 2012.

[18] Brugger, S.T. and J. Chow, *An Assessment of the DARPA IDS Evaluation Dataset Using Snort*, D. UC Davis Technical Report CSE-2007-1, CA, Editor. 2007.

[19] Franc, V. and V. Hlavác. *Statistical pattern recognition toolbox for Matlab*. Center for Machine Perception, Czech Technical University 2004; Available.

[20] Mathworks, *MATLAB*. 2014, http://www.mathworks.com.

INSTRUCTIONS TO CONTRIBUTORS

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.

The initial efforts helped to shape the editorial policy and to sharpen the focus of the journal. Started with Volume 9, 2015, IJCSS is appearing with more focused issues. Besides normal publications, IJCSS intend to organized special issues on more focused topics. Each special issue will have a designated editor (editors) – either member of the editorial board or another recognized specialist in the respective field.

We are open to contributions, proposals for any topic as well as for editors and reviewers. We understand that it is through the effort of volunteers that CSC Journals continues to grow and flourish.

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
- Computer Engineering
- Computer Networks
- Cryptography
- Databases
- Image processing
- Operating systems
- Programming languages
- Signal processing
- Theory
- Communications and data security
- Bioinformatics
- Computer graphics
- Computer security
- Data mining
- Electronic commerce
- Object Orientation
- Parallel and distributed processing
- Robotics
- Software engineering

CALL FOR PAPERS

Volume: 9 - **Issue: 5**

i. Submission Deadline : September 30, 2015 **ii. Author Notification:** October 31, 2015

iii. Issue Publication: November 2015

CONTACT INFORMATION

Computer Science Journals Sdn BhD

B-5-8 Plaza Mont Kiara, Mont Kiara

50480, Kuala Lumpur, MALAYSIA

Phone: 006 03 6204 5627

Fax: 006 03 6204 5628

Email: cscpress@cscjournals.org

COMPUTER SCIENCE JOURNALS SDN BHD
B-5-8 PLAZA MONT KIARA
MONT KIARA
50480, KUALA LUMPUR
MALAYSIA

PHONE: 006 03 6204 5627
FAX: 006 03 6204 5628
EMAIL: cscpress@cscjournals.org