KIPS Tr. Comp. and Comm. Sys.
Vol.2, No.12 pp.541~552 pISSN: 2287-5891

http://dx.doi.org/10.3745/KTCCS.2013.2.12.541

An Integrative Method of FTA and FMEA for Software Security Analysis of a Smart Phone

Myong-Hee Kim[†] · Wildan Toyib[††] · Man-Gon Park[†††]

ABSTRACT

Recently software security of the smart phone is an important issue in the field of information science and technology due to fast propagation of smart technology in our life. The smart phone as the security critical systems which are utilizing in terminal systems of the banking, ubiquitous home management, airline passengers screening, and so on are related to the risk of costs, risk of loss, risk of availability, and risk by usage. For the security issues, software hazard analysis of smart phone is the key approaching method by use of observed failures. In this paper, we propose an efficient integrative framework for software security analysis of the smart phone using Fault Tree Analysis (FTA) and Failure Mode Effect Analysis (FMEA) to gain a convergence security and reliability analysis technique on hand handle devices. And we discuss about that if a failure mode effect analysis performs simpler, not only for improving security but also reducing failure effects on this smart device, the proposed integrative framework is a key solution.

Keywords : Software Security, Smart Phone, Reliability, Fault Tree Analysis(FTA), Failure Mode Effect Analysis(FMEA)

스마트 폰의 소프트웨어 보안성 분석을 위한 FTA와 FMEA의 통합적 방법

김명희[†] · Wildan Toyib[††] · 박만곤[†††]

요 약

최근 우리 생활에 스마트 기술의 빠른 전파 때문에 정보 과학 및 기술 분야에 있어서는 스마트 폰의 소프트웨어 보안성이 중요한 이슈가 되고 있다. 보안성 중요 시스템인 스마트 폰은 은행 서비스, 유비쿼터스 홈 관리, 항공 고객의 검색 등의 서비스 시스템에 이용되기 때문에 비용의 리스크, 손실의 리스크, 이용가능 리스크, 그리고 사용상의 리스크에 관련 되어 있다. 스마트 폰의 보안성 이슈는 이들의 관찰된 고장들을 사용하여 소프트웨어 장애 분석을 하는 것이 핵심 접근 방법이다. 본 연구에서는 손으로 조작하는 디바이스들의 수렴하는 보안성과 신뢰성 분석 기법을 얻기 위해서 결함 트리 분석 (FTA)와 고장 모드 효과 분석(FMEA)을 사용하여 스마트 폰의 소프트웨어 보안성 분석을 위한 하나의 유효한 통합적 프레임 워크를 제안한다. 그리고 만약 하나의 고장 모드 효과 분석이 더욱 더 간단해지면 스마트 디바이스들의 보안성 개선뿐만 아니라 고장효과 의 감소를 위해서 제안된 통합적인 프레임 워크는 핵심 해법이 됨을 논의한다.

키워드 : 소프트웨어 보안성, 스마트 폰, 신뢰성, 결함 트리 분석(FTA), 고장 모드 효과 분석(FMEA)

1. Introduction

Smart Phone is the new generation of mobile and embedded devices, such as mobile phones and Personal Digital Assistants (PDA) which support a rich set of applications, web browsing, SMS-MMS, i-television, i-radio, multimedia and entertainment applications[1]. The time to market pressure forces manufacturers to deliver products with new features within very short time testing efforts. We witness an increasing susceptibility of hand-held devices to accidental errors and malicious attacks. The example is recently reported first mobile phone virus, namely cabir, affecting symbian. Security becomes even more critical as new critical applications emerge for mobile phones, as examples robot control[2-3], traffic control[4], telemedicine, pervasive and ubiquitous applications[5]. In such scenarios, a phone failure affecting

[†] 준 회 원 : 부경대학교 정보시스템학과 강사
[††] 비 회 원 : 부경대학교 첨단정보 및 정보기술전공
[†††] 종신회원 : 부경대학교 IT융합응용공학과 교수

논문접수 : 2013년 10월 7일
심사완료 : 2013년 11월 18일

* Corresponding Author : Man-Gon Park(mpark@pknu.ac.kr)

the application can result in a significant loss or hazard, the robot performing uncontrolled actions on the mobile device remote monitoring system, despite these concerns, very few studies have looked into the dependability of smart phones related to the security system.

There is a few understanding of how and why smart phones fail. This article presents software security analysis of a smart phone using integration of FTA (Fault Tree Anlysis) and FMEA (Failure Mode Effect Analysis). The analysis starts with a high level failure characterization of smart phones based on everyday user's experiences. Data for this study spans the two years period and obtained from publicly available web forums, society and communities. The collected data enables the characterizations of the failure, occurrence, detection and severity, identification of the high level failure manifestation, categorization of the user initiated recovery from the device failure and, list of fault related to security of the development tools, web browsers, multimedia and entertainment applications.

This initial analysis is then used to guide the development of a failure data logger for smart phones. Initially introduced in the logger employs heartbeat mechanism to detect system and application failures[7]. Upon failure detection, the logger records information about the smart phone activities, presenting the list of fault related to safety and security mechanism because of the human factor, and social networking. The error conditions signal effect by the system or application modules and web browser authentication through https in the OSI layer[8]. Table 1 shows recent global sales figures and market share of smart phone operating systems.

Based on Table 1 which represent global market of OS within smart phone, Google and apple are the obvious winners in the smart phone ecosystem. The combined share of iOS and Android in the smart phone operating system market double to nearly 62% in the second quarter 2011, up from just over 31% in the corresponding period of 2010.

Table 1. Global Sales Figures and Market Share of Smart Phone Operating Systems for Third Quarter of 2010 and 2011

Platform	July-October 2010		July-October 2011	
	Units/1k	Share[%]	Units/1k	Share[%]
Android	10,652.7↑	17.2↑	46,775.9↑↑	43.4↑↑
Symbian	25,386.8↑	40.9↑	23,853.2↑	22.1↑
iOS	8,743.0↑↑	14.1↑↑	19,628.8↑↑	18.2↑↑
RIM	11,628.8↑↑	18.7↑↑	12,652.3↑	11.7↑
Bada	577.0↑	0.9↑	2,055.8↑	1.9↑
Microsoft	3,058.8↑	4.9↑	1,723.8↓	1.6↓
Others	2,010.9↑	3.2↑	1,050.6↓	1.0↓
Total	62,058.1	100.0	107,740.4	100.0

(↑↑: 2 periods increase, ↑: 1 period increase, ↓: 1 period decrease)

In these methods and technologies, FTA, which generates the use cases by the minimal cutsets of fault trees, can't determine the priorities of all the use cases and can't utilize the finished software test result. In order to solve these problems, a software security analysis approach with software FMEA which are transferred from fault trees is proposed in this paper.

Some researchers considered that the forward integrated analysis of hardware is more labor intensive and difficult to apply compared with the backward one. And even the principle of the integrated analysis techniques is still under study. The principle and process of both forward and backward integrated analysis techniques of software FTA and FMEA are discussed [31-34], also the failure of smart phone affected by weak security is the object of analysis in this paper.

Hence, in this paper, we propose integrative method of software FTA and FMEA for security analysis within smart phone, this integrative approach is aimed to reduce and classify faults, failures and vulnerable networks. We create a mathematical logic which is applied in the fault tree cutset symbols, hence, find the accurate technique to reduce failure most of them in the smart phone with software FMEA. Also we assert these methods are very strong related to the security system technique.

2. Related Works

The top most application layer provides applications such as a phone call, web browser, email client and more applications. Each application in android is packaged in an .apk archive for installation. This archive is similar to a Java standard *.jar file in the way that it holds all code and noncode resources such as images or manifest for the application. Android applications are written in Java based on the APIs Software Development Kit (SDK) provides. William Enck et al [2011] discussed the main components of an android application and how to use an android specific mechanism to protect applications[6]. As depicted in Table 2, several security mechanisms are incorporated in smart phone operating systems. We can cluster them into three general groups such as mechanism, description, and security issue in detail.

Smart phones carrier security features, telephony systems have a basic set of attributes and functionalities

Table 2. Software Security Mechanism Incorporated in Smart Phone Operating Systems

OS	Mechanism	Description	Security Issue
Linux	POSIX users	Each application is associated with a different user ID	Prevents application from disturbing
	File access	Application's directory available to the application	Prevents application from accessing
	Memory ma'ment unit (MMU) Type safety	Each process is running in its own address space	Prevents privilege escalation, disclosure and denial
	Type safety	Type safety enforces variable content to adhere to a specific format both in compiling time and runtime.	Prevents buffer overflows and stack smashing
	Mobile carrier security features.	Smart phones use SIM cards to authenticate and authorize user identity.	Prevents phone call theft
Android	Application permissions	Each application declares which permission it requires at install time.	Limits application abilities to perform malicious behavior
	Component encapsulation	Each component in an application (activity or service) has a visibility level that regulates access to it from other applications (i.e., binding to a service).	Prevents one application from disturbing another or accessing private components or APIs
	Signing applications	The developer signs application .apk files, and the package manager verifies them	Verifies that two applications are from the same source
	Dalvik virtual machine	Each application runs in its own virtual machine	Prevents buffer overflows, remote code execution and stack smashing.
Symbian	Beyond Client/Server Sessions	Publish & Subscribe also known as properties that provide a means to define and publish system. Message Queues	Bug on the both user and kernel side program via similar APIs. Message queues offer a peer-to-peer or many-to-many communication paradigm.
	Usage scenarios new IPC	Shared Buffer I/O drivers no need to have a buffer of their own	I/O device drivers not need to have a buffer but can share a buffer with a user space process.
iOS	Root Exploits	Root exploits libtif and SMS fuzzing	Multiple buffer overflows (Spamms)
	Personal Data Harvesting	Aurora Feint, MogoRoad, Storm8 complaint, Pinch Media	The blackmailer, the jealous husband
	Worms on jailbroken devices	This privacy contains of Ikee, dutch 5€ransom, iPhone/Privacy.A, Ikee.B/Duh	Attacks targeting jailbroken iPhones, exploit the fact that very few user bother to change the default root password(alpine) after jailbreaking their iPhone and installing a SSH server
	iPhone forensics	Physical access to any device means that pretty much everything can be compromised with the notable exception of passwords, which encrypted in the phones KeyChain	The Jealous Husband, apple got a lot of bad press, insecure configuration
RIM	Sandboxing	a virtual container that consists of the memory and the part of the file system that the application process has access to at a specific time	Brute-force attack, online dictionary attack, eavesdropping, impersonating a smartphone, man-in-the middle attack and small subgroup attack.
Windows mobile	Safeguards	Concerned individuals and organizations aware of the potential risks involved can often mitigate many of the associated threats with add-on security mechanisms.	Loss Theft or disposal, Because of their small size, handheld devices have a propensity to become lost or misplaced.
	Maintain Physical Control	Verifying an individual's claimed identity through user authentication is the first line of defense against unauthorized use of a mobile handheld device.	Unauthorized access, guessing authentication credentials (e.g., a PIN or password
	Enable User Authentication	handheld device as the sole repository for important is an invitation for disaster.	Denial of Service, key logger, open https.
	Backup data	Authentication mechanisms can be bypassed or broken and even deleted information can often be recovered from memory.	Trojan horse, duplication data, failure in encryption process
	Reduce Data Exposure	Malicious programs to mobile phones mainly through communications channels such as multimedia messages or Bluetooth connections, Any messages or contacts received on a mobile phone from an unknown number or device should be treated with suspicion.	Malware is typically targeted more toward handheld devices for which a SDK is available than those without one, since code development is easier to perform
	Shun Questionable Actions	A simple defense against many forms of malware is to turn off Bluetooth, Wi-Fi, infrared, and other wireless interfaces.	Hijack, tcdump, mirror connection, attack on SSID by hacking tools
	Curb Wireless Interfaces	The most direct way of electronic eavesdropping is for spy software to be installed onto a device to collect and forward information within another phone or server	Keylogger, and broken firewall
	Deactivate Compromised Devices	Device lost or stolen, disabling service, locking it, or completely erasing its contents are useful actions to take remotely.	Electronic Tracking
	Minimalize Functionality	user authentication alternatives including biometric and token-based mechanism	Cloning into a second cell phone
	Add Prevention and Detection Software	Memory card encryption, firewall, antivirus, instruction detection, antispam, device content and memory card erasure and virtual private networking	Server-resident data is the server was able to be accessed by unauthorized

stemming from a need to identify users, monitor usage and charge the client accordingly, A more general term for these features is AAA (Authentication, Authorization and Accounting), as a smart phone platform, an android borrows these classical security features from cellular phone design. Authentication usually occurs via SIM card and associated protocols. Thus, below we conduct to interpret a software fault for android, symbian, iOS, RIM, and Windows.

The first, Android specific security mechanisms. Android provides the following dedicated security mechanisms introduced by google. Android has roughly 100 built-in permissions that control operations ranging from dialing the phone *(CALL_PHONE)*, taking pictures *(CAMERA)*, using the Internet *(INTERNET)*, listening to key strokes *(READ_ INPUT_STATE)*, and even disabling the phone permanently *(BRICK)*[10-12].

At installation phase, the system grants permissions that the installed application requests based on checks of that application's signature against those of the applications declaring the permissions. After the user has installed the application and it receives permissions, it can not longer request any more permissions. There are certain risks of device which can be called risk of costs, risk of loss, risk of availability, and risk by usage[10],[13-14]. The Second, Symbian security features like public key signatures on applications and others root CAs in ROM. It also others different spaces for the kernel and the user space, and of course symbian supports access controls such as SIM-PIN, a device security code and bluetooth pairing with a key. This is hardly handled by symbian. Symbian has no functionality to remotely shut down a lost or stolen device. This risk is hardly covered as any application can render the phone unusable which has been proven by the skull trojan[10].

The another risk of Symbian is viruses as well known skull trojan. When installed, every link on the mobile phone will be replaced by a scull making the device unstable. Another virus is the proof of concept virus EPOC, well known cabir, which came out in 2004, and spreaded over Bluetooth. Since then, hundreds of viruses have come into being[11-12].

The third OS is Apple iOS, within this OS every application runs in its own sandbox and terminates when the user presses the 'Home' button. iOS runs a daemon called the 'security server' which implements several security protocols. Apple provides the only possibility of loading programs from "App Store" to device. These programs are checked before they appear in 'App Store'.

But a lot of people have jailbroken their iOS is allowing to load software on to their device without going through the 'App Store', this indeed is risky when we install instable software. Since Apple controls which app lands in the app store we can say that we are quite secure not to load a dialer or something like that on to our iPhone.

iOS handles this risk if we are connected to a microsoft exchange server. Then we can remotely wipe our phone from the server side, iOS showed this feature as a highlight on their SDK. First, the applications run inside their own sandbox without contact to other parts of the iOS than the permitted space. Second, iOS checks every application goes to the 'App Store' so harmful software will not find the way on to the iOS[13], [15-17].

Finally, Windows mobile edition. There are studies that show the Windows mobile edition seems to be just as vulnerable as Symbian, if not even more[12]. Microsoft began as well to implement security policies with signed applications. But unfortunately these policies do not work as advertised because of bugs, consequently vulnerability still exists. In addition to that, the application unlock for Windows mobile edition needs no knowledge and takes at least ten minutes. A lot of Windows mobile devices may be unlocked as open source software is mostly unsigned[12],[16-19]. How does Windows mobile cover our risks? Users either install the malicious software itself or they may spread through WiFi or Bluetooth. A dialer could use this for doing extensive harm[12]. Windows mobile has no possibility of being shut down remotely. Windows mobile is vulnerable here as a virus may access the data on the device. Windows mobile users can take a deep breath now as in the beginning of 2008 only 5 viruses existed[13],[19].

3. Software Security Analysis Methods

3.1 Identification of Software Faults in Smart Phone Environment

The identification of software faults for security analysis is based on the FTA[5]. Our process is comprised of four main steps. In the first step, the Software Requirements Fault Tree (SRFT) is generated which identifies the security faults in the software requirements. This fault tree is then verified, validated and corrected if necessary in the second steps. In the third step, security requirements are generated and the original software requirements specification is modified to comply with the security requirements. In the fourth step, the security of the resulting software requirements is verified and validated. These steps are iterative and they can be performed any number of times during the process[14]. We provide entities for smart phone failure related to security system. The explanation of the list of fault is an analysis in hardware both of software system as a basic system for an application, the entities are divided into two categories, hardware and software. Thus, we can collect many fault caused in panic neither failure activities that make easy to be vulnerable a security within software system on the smart phone activities. Table 3 describes all entity variables related to software failures of the smart phone security. The core function below is an introduction which can run on hardware and software, these failures are considered for security effect, thus, we will integrate by use of Software FTA and FMEA.

3.2 Functional Block Diagram for Software Security Analysis of a Smart Phone

The functional block diagram (FBD) describes a function between input and output variables. A function is described as a set of elementary blocks, Input and output variables are connected to blocks by connection lines. In this method we classify the main category within control panel, type of operating system, hardware and software system, data logger, five key aspects of security, threat model of security, three class of target, threat attack model for smart phone device security[19].

The performance illustrates on the Fig. 1 where FDB describes the security method for handle in security analysis which is extracted with four block diagram. On the other hand, the first block will perform hardware system detail which is caused the smart phone failure.

Table 3. The list of entity variables related to software failures of the smart phone security

Entity Variables	Failure (Panic)
KERN -EXEC	This Panic is raised when the kernel execute cannot find an object in the object index.
	This panic is raised when an unhandled exception occurs, and causes
	This Panic is raised when a timer event is requested from an asynchronous timer service, an RTimer, and a timer event is already outstanding. It is caused by calling either the At( ), After( ) or Lock( ) members.
E32USER -CBase	This panic is raised by the destructor of a *CObject.*
	This panic is raised by an active scheduler, a *CActiveScheduler.*
	This panic is raised by the *Error( ), CActiveScheduler,RunL( ), Error( ) and CActiveScheduler.*
	This panic is raised if no trap handler has been installed as *CTrapCleanup::New()*
USER	This panic may be raised by the Left( ), Right( ), Mid( ), Insert( ), Delete( ), and Replace( ).
	It may be caused by any of the copying, appending or formatting member functions and, specifically, by the Insert(), Replace(), Fill(), Fillz(), ZeroTerminate(), and SetLength() function
KERN-SRV	This panic is raised by the kernel server when it attempts to close a kernel object in response to a RHandleBase::Close () request.
ViewSrv	Occurs when one active object event handler monopolizes the thread active scheduler loop and the applications ViewSrv active cannot respond in time.
EIKON -LISTBOX	Occur when using a list box object from the eikon framework and no view is defined to display the object
	Occur when using a list box object from the eikon framework and an invalid current item index specified
Virus Attacker	Such as an intensive IP address scan/sweep attack on MS can evoke a paging storm, and consequently a connection setup storm, which would overload the mobile network equipment such the Radio Network Controller (RNC) and Serving GPRS Support Node (SGSN).
Fast Dormancy	Such as RRC state such as IDLE, CELL_PCH, URA_PCH, and CELL_FACH.
Phone.app	Bugs
EIKOCTL	Corrupt Edwin state for in lining editing
MSGS Client	Failed to write data into asynchronous call descriptor to be passed back to client
Always on line PDP context	Always on line application requires a permanent IP connection, such as Deactivation Accept, Deactivation Ignore and Re-Activate PDP After Deactivation.
Always on line application	Smart Phone allows people to access the internet anytime anywhere for any kind of service, for real-time web services, pull/polling, long polling and Push.

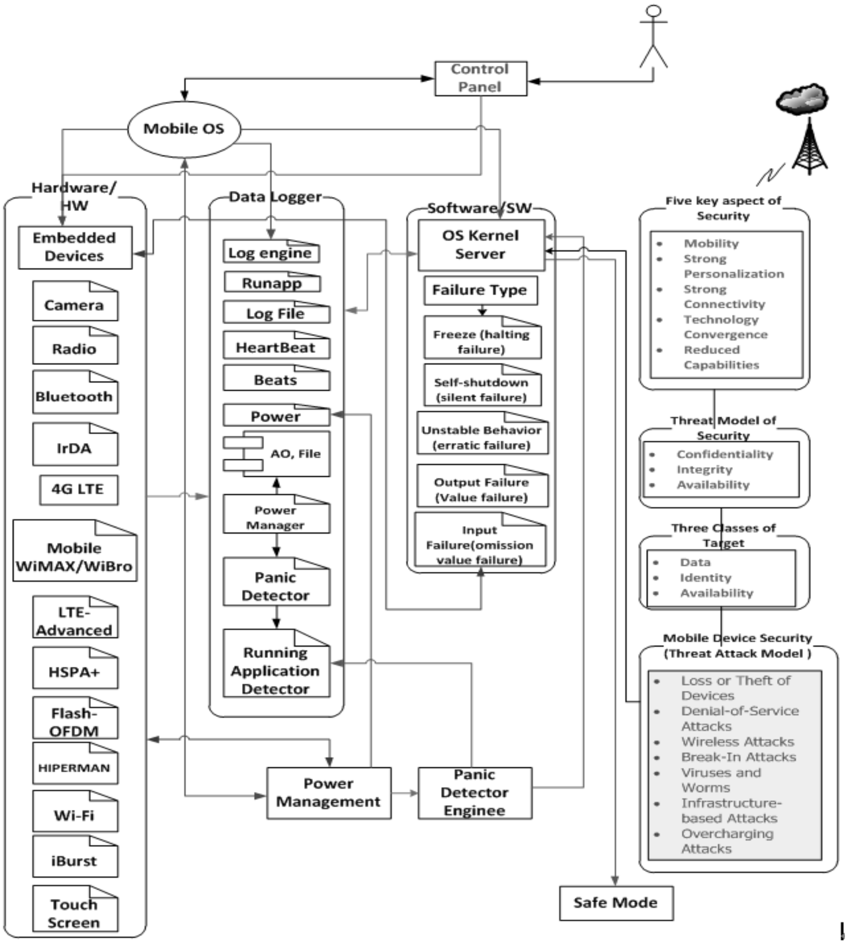


Fig. 1. Functional Block Diagram of Smart Phone for Security Analysis

Second block is containing of data logger which have function as a parameter for detection methods. The third is software system which is the critical security for this issue in Kernel management server. And the fourth is a block for network infrastructure with the correlation of information security area, aspect security, threat, target and model of attack, for detail performance, it will illustrate as is below completely with the user access, cloud internet infrastructure, implementation of power management, applying panic detector engine, since then the operating system will getting safe mode, this behavior appear as a mark that system in recovery mode on, for further detail information.

We perform a key aspect of security such as mobility hand handle device, strong personalization within network, strong connectivity with another tools such as bluetooth, irDA, RFID and another, then technology convergence which means a central information provided for anywhere, anytime with smart device accessing information, finally the key concept of security is a capabilities to reduce an attack both of virus neither worm.

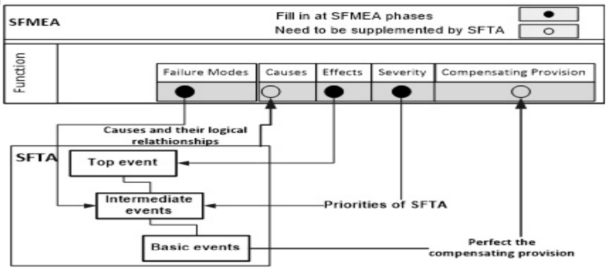

SFMEA
Fill in at SFMEA phases ●
Need to be supplemented by SFTA ○
Function
Failure Modes
Causes
Effects
Severity
Compensating Provision
Causes and their logical relathionships
SFTA
Top event
Intermediate events
Basic events
Priorities of SFTA
Perfect the compensating provision

Fig. 2. Forward Integration Security Analysis Technique of Smart Phone Failures for FMEA

3.3 Forward Integrated Security Analysis of Software FMEA and FTA

Software FMEA was taken as the main concepts for the forward integrated for this security analysis for smart phone, decompose by Software FTA as supplementation. The mechanism of forward integrated software security analysis is performed in Fig. 2. We will classify, the difference of forward and backward analysis. Forward analysis principle implies SFTA can be performed in sequence according to the severity degree of failure effects from the results of SFMEA. Also, the weakness of this method is the failure modes can be used as intermediate events of SFTA to identify the causes of the failure modes, which might be difficult to identify or express with SFMEA. Security analysis can be showed more comprehensively through forward integrated analysis. The merits within this method, improvement actions might be suggested.

Software FTA is able to evaluate in many sequence related to the severity degree of the failure effects on smart phone from the results of software FMEA generator. The cause and effect with higher severity ought to be implemented as top events of software FTA to classify the causes of these effect detail. Also, the failure mode can use as intermediate events if software FTA to identify the causes of the failure modes, which should be hard to identify or explain with Software FMEA, security analysis can be showed more integrative through forward integrated analysis as shown in Fig. 2. Here the step of the forward integrated analysis methods. *First step:* Web can choose analysis level for security software for smart phone, both of functional neither structural level within the device. *Second step:* Evaluation of this failure will be performed to the matrix of each failure effects.

If the condition permitted, the critical analysis may be performed by multiplying the value of severity, occurrence and detection, which is defined as the risk priority number (RPN). Severity shows the seriousness of the effect of the failure. Occurrence represents the frequency of the failure. And detection is the probability of the failure being detected of the impact effect. *Third step:* the higher severity degree as top event is function of security analysis SFTA for smart phone. *Forth Step:* Intermediate event for software security analysis on the smart phone is like erratic, freeze failure. *Fifth Step:* the failure mode for this device such as alive event, low battery event, reboot and manual off fault, which will be figured on the basic FTA.

3.4 Backward Integrated Security Analysis of Software FMEA and FTA

Software FTA is taken as the main concept for the backward integrated security analysis, followed by software FMEA as supplementation, the principle of backward integrated analysis is simulated in Fig. 3.

If criticality analysis has been performed in the process of software security analysis FMEA, the occurrence probability of the top event can be calculated with the assistant of CA report. The backward integrated security analysis on the smart phone fault technique are performed. *First step:* Collect the top events base on software security requirement, and create the software fault trees. *Second step:* Classified the minimal cutsets and necessary level of bottom events are; the lower the order of minimal cutset, the more necessary this minimal cutset, the bottom events in minimal cutset with lower order is more important than that with higher order, the most often a bottom event appears in different minimum cutset, the most important that bottom event, if the orders of the minimal cutset is the same. *Third step:* Show software security analysis FMEA with the more important bottom events taken as failure modes. *Forth step:* Redesign the software security analysis fault trees and development actions. *Fifth step:* Classify new failure effects and causes as top events to build new fault trees and deploy security analysis furthers. Sixth step: Create the failure of effect, cause and severity to create new symbols of failure mode.

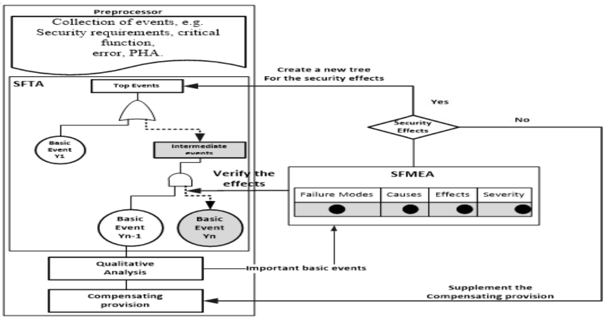


Fig. 3. Backward Integration Security Analysis Technique of Smart Phone Failures for FMEA

3.5 Integrative Method of FTA and Software FMEA for Software Security Analysis

We have some guidelines in the selection of integrated analysis techniques. The first is forward integrative security analysis. Namely, an analysis which works in this phase should be comprehensive and meticulous enough to discover software defects within smart phone as completely as possible at early stage of software development. Some times this method might be a better choice to avoid omission due to human factors, error correcting code, and so on. Second is backward integrated security analysis. Commonly backward integrated analysis is a technique which advocates efficiency. It might be feasible to select the undesired events with higher severity degree or of greater concern to carry out analysis in the design phase within smart phone fault and failure. Furthermore, we create cut sets by approaching FTA method for system failure within smart phone

Table 4. Worksheet of the Software FMEA Module of Security Analysis on the Smart Phone Failure

ID	Item	Failure Mode	Causes	Effects	Severity	Compensating provision
18	SW	Device output constant, not respond to user's input	Freeze	Lock-up, halting failure	High	Repeat the action; MTBF_Freeze
		Device self shutdown, no service delivered to user interface	Self-halt	Silent Failure	Minor	Wait an amount of time; MTBF_Shutdown
		Device exhibits erratic without any input inserted by the user	Unstable behaviour	Erratic Failure	Moderate	Reboot; Running Application Detector;
		Error monitoring and failure data analysis, fault injection & design methodology	Weak security	Failure data analysis	Moderate	Shifting error sources, explosive complexity, and global volume inaccurate
		Failure data for data logger applications	Software Failure	Out of date	High	Software Event Failure detector; RTimer Updater

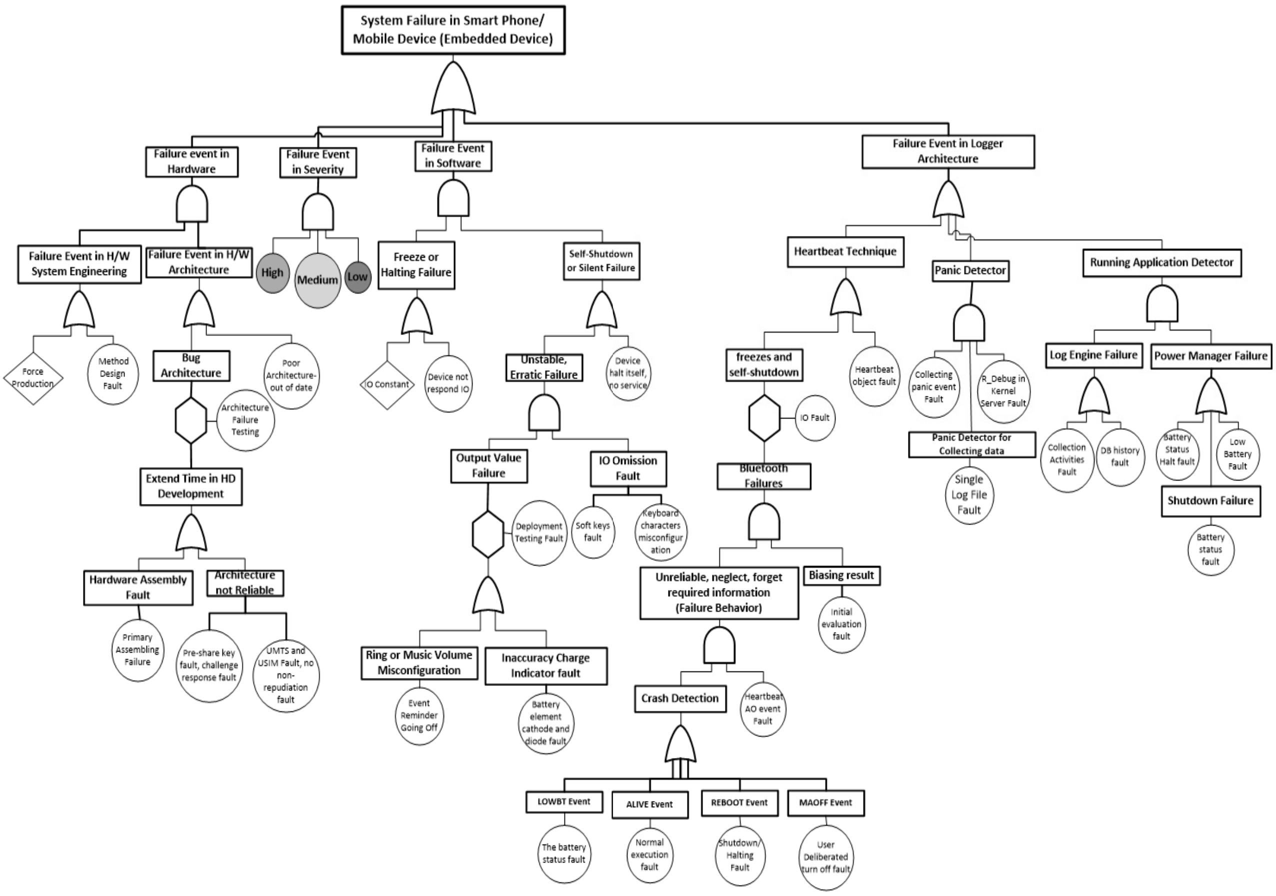

System Failure in Smart Phone/ Mobile Device (Embedded Device)
Failure event in Hardware
Failure Event in Severity
Failure Event in Software
Failure Event in Logger Architecture
Failure Event in H/W System Engineering
Failure Event in H/W Architecture
High
Medium
Low
Freeze or Halting Failure
Self-Shutdown or Silent Failure
Heartbeat Technique
Panic Detector
Running Application Detector
Force Production
Method Design Fault
Bug Architecture
Poor Architecture-out of date
IO Constant
Device not respond IO
Unstable, Erratic Failure
Device halt itself, no service
freezes and self-shutdown
Heartbeat object fault
Collecting panic event Fault
R_Debug in Kernel Server Fault
Log Engine Failure
Power Manager Failure
Architecture Failure Testing
IO Fault
Panic Detector for Collecting data
Collection Activities Fault
DB history fault
Battery Status Halt fault
Low Battery Fault
Extend Time in HD Development
Output Value Failure
IO Omission Fault
Bluetooth Failures
Single Log File Fault
Shutdown Failure
Deployment Testing Fault
Soft keys fault
Keyboard characters misconfigur ation
Battery status fault
Hardware Assembly Fault
Architecture not Reliable
Unreliable, neglect, forget required information (Failure Behavior)
Biasing result
Primary Assembling Failure
Pre-share key fault, challenge response fault
UMTS and USIM Fault, no non-repudiation fault
Initial evaluation fault
Ring or Music Volume Misconfiguration
Inaccuracy Charge Indicator fault
Crash Detection
Heartbeat AO event Fault
Event Reminder Going Off
Battery element cathode and diode fault
LOWBT Event
ALIVE Event
REBOOT Event
MAOFF Event
The battery status fault
Normal execution fault
Shutdown/ Halting Fault
User Deliberated turn off fault

Fig. 4. The Main Structure of Security Events on the Software FTA for Smart Phone Security Analysis

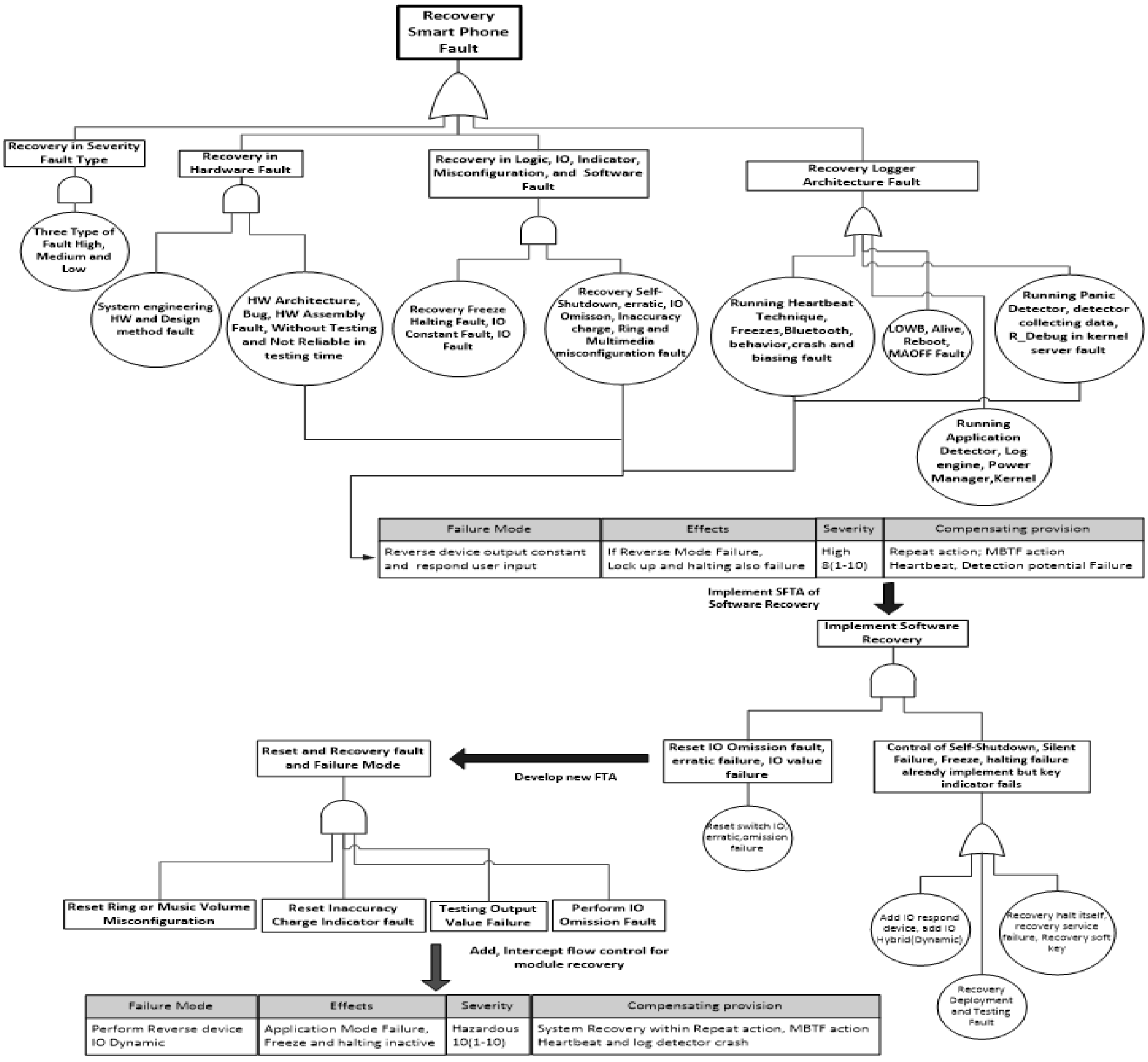
structured, who is the higher level severity is fault in the system software. In the Fig. 4, the process to minimize the high level failure will be performed.

Based on the Fig. 4, we mention the security analysis within smart phone, integrated method analysis in requirement, and integrated method analysis in framework. So, each of the 32 functional modules is analyzed thoroughly and 30 failures modes are identified. Failure modes with higher severity degree of failure effects are also identified, e.g. crash detection, shutdown failure, ring and music volume mis-configuration, inaccuracy charge indicator fault, failure on behavior, bluetooth failure, power management failure, freeze. etc. we can use these failures effects as top events to perform FTA. Thus, integrated method for security analysis within this framework, use this failure effect as a top event to build a new fault tree for further analysis, automatically a new improvement action, adding a new software watchdog module, is recommended to activate software reset function when hardware reset signal is

Table 5. Risk Priority Number Assessment According to Failure Modes

ID	Item	Failure Mode	S	O	D	RPN
18	SW	Device output constant, not respond to user's input	8	7	6	336
		Device self shutdown, no service delivered to user interface	9	6	8	432
		Device exhibits erratic without any input inserted by the use	6	7	5	210
		Error monitoring and failure data analysis, fault injection and design methodology	6	5	8	240
		Failure data for data logger app.	9	8	6	432
		Sub-total	38	33	33	1650

S: Severity, O: Occurance, D: Detection


Recovery Smart Phone Fault
Recovery in Severity Fault Type
Recovery in Hardware Fault
Recovery in Logic, IO, Indicator, Misconfiguration, and Software Fault
Recovery Logger Architecture Fault
Three Type of Fault High, Medium and Low
System engineering HW and Design method fault
HW Architecture, Bug, HW Assembly Fault, Without Testing and Not Reliable in testing time
Recovery Freeze Halting Fault, IO Constant Fault, IO Fault
Recovery Self-Shutdown, erratic, IO Omisson, Inaccuracy charge, Ring and Multimedia misconfiguration fault
Running Heartbeat Technique, Freezes,Bluetooth, behavior,crash and biasing fault
LOWB, Alive, Reboot, MAOFF Fault
Running Panic Detector, detector collecting data, R_Debug in kernel server fault
Running Application Detector, Log engine, Power Manager,Kernel
Implement SFTA of Software Recovery
Implement Software Recovery
Reset IO Omission fault, erratic failure, IO value failure
Control of Self-Shutdown, Silent Failure, Freeze, halting failure already implement but key indicator fails
Reset switch IO, erratic,omission failure
Add IO respond device, add IO Hybrid(Dynamic)
Recovery halt itself, recovery service failure, Recovery soft key
Recovery Deployment and Testing Fault
Develop new FTA
Reset and Recovery fault and Failure Mode
Reset Ring or Music Volume Misconfiguration
Reset Inaccuracy Charge Indicator fault
Testing Output Value Failure
Perform IO Omission Fault
Add, Intercept flow control for module recovery

Failure Mode	Effects	Severity	Compensating provision
Reverse device output constant and respond user input	If Reverse Mode Failure, Lock up and halting also failure	High 8(1-10)	Repeat action; MBTF action Heartbeat, Detection potential Failure

Failure Mode	Effects	Severity	Compensating provision
Perform Reverse device IO Dynamic	Application Mode Failure, Freeze and halting inactive	Hazardous 10(1-10)	System Recovery within Repeat action, MBTF action Heartbeat and log detector crash

Fig. 5. Backward Integrated Faults Analysis Diagram of Security Events for Smart Phone Security Analysis

shielded, after framework is modifying to the improvement tools, software FTA continues based on supplementation of new deployed information.

A new bottom event, namely, watchdog failure is detected, which is then used as a failure mode to perform software FMEA. The potential failure effect is that software abort abnormally event out of control. Thus, an improvement action which is performed through Table 4 will be classified within failure mode effect analysis framework. Furthermore, we identify some items for these failure effect both of software application and hardware system.

Table 6. Software Security Variables on Recovery Actions

Failure Type	Recovery/Security Action					
	Service Phone	Reboot	Battery Removal	Wait an amount of times	Repeat	Unrepeated
Failure Severity		√				
Heartbeat		√			√	
Running App. Detector	√					√
Log Engine		√		√	√	
Power Manager	√					
Panic Detector	√			√	√	√

Table 7. Risk Matrix Table for Security Analysis of the Smart Phone Faults (T: *True*, F: *False*)

High Level Event	Panic Category	Applications														
		Log Browser	Browser	Messages	Message Log	Camera Log Telephone	Clock	Clock Log	Log	Log Contacts	BT_Browser Log	Contacts	Telephone Contacts	Battery	Messages	Telephone
Freeze	KERN-EXEC	T					T	T	T	T	T					
Self-Shut-down	KERN-EXEC		T			T							F	F		
	MSGS Client			T					T							
No High Level Event	E32USER-CBase										T				T	
	EIKCOCTL		T			T										
	EIKON-LISTBOX							T					T	T		T
	KERN-EXEC		T		T		T		T	T	T	T	T	T	T	T
	USER	T		T			T					T				T
	ViewSrv	T				T	T		T	T	T					

Software FTA and FMEA can be applied as supplementary techniques. More comprehensive and effective analysis results can be obtained by integrating both FTA and FMEA. These two analysis methods can be integrated become single view as shown in Fig. 5. RPN (Risk Priority Number) for identifying major risks is equal multiplication from severity, occurrence and detection as shown in Table 5.

For further recovery, we take action within software security variables as shown in Table 6. And by collecting recovery data from smart phone failure analysis table, we build the risk matrix table to perform the recovery failure analysis as represented in Table 7.

4. Conclusion

In this paper, we propose an integrative methods of software FTA and FMEA for security analysis of a smart phone through integrative approaches of FTA and software FMEA. This integrative methods for security analysis has generated a simple approach procedure for software security analysis of a smart phone. The analysis reports are used by smart phone global sales figures and market share with 3rd quarter, software security issue incorporated through operating system, functional flow diagram for software security requirement analysis. On the other hand, the variables of smart phone faults are related to software security system, threat attack model especially for mobile device security, a logic functional block diagram with three aspects for functional diagram such as software, hardware and data logger which are supported by five key aspects of security, threat model of security, three cause of target security, forward and backward integrated security analysis techniques for software FTA and FMEA. The proposed business processes of integrative methods of software FTA and FMEA for security analysis are followed several steps such as failure mode identification, worksheet preparation of the software FMEA modules comprehensive and effective analysis through forward and backward integrated faults analysis, risk priority number calculation, definition of software security variables on recovery actions for failure, and risk matrix table building. We believe that this procedure is very reliable approach for software security analysis through integration of software FTA and FMEA of a smart phone.

Reference

[1] M. Cinque, D. Cotroneo, Z. Kalbarczyk, R. K. Iyer, “How Do

Mobile Phones Fail? A Failure Data Analysis of Symbian OS Smart Phones," in *Proc. of Dependable Systems and Networks (DSN '07)*, on 37th Annual IEEE/IFIP International Conference, pp.585-594, Jun., 2007.

[2] A. Shabtai, Y. Fledel, Y. Elovici, "Automated Static Code Analysis for Classifying Android Applications Using Machine Learning," in *Proc. of 2010 International Conference on Computational Intelligence and Security (CIS)*, pp.329-333, Dec., 2010.

[3] H. Reza, S. Buettner, V. Krishna, "A Method to Test Component Off-the-Shelf (COTS) Used in Safety Critical Systems," in *Proc. of Information Technology: New Generations 2008 (ITNG08) on Fifth International Conference*, pp.189-194, Apr., 2008.

[4] J. White, C. Thompson, H .Turner, B. Dougherty, and D. C. Schmidt, "Primary Title: Wreck Watch: Automatic Traffic Accident Detection and Notification with Smartphones," in *Proc. of Mobile Networks and Applications*, Springer, Vol.16, pp.285-303, 2011.

[5] Hsiu-Sen Chiang, Woei-Jiunn Tsaur. "Identifying Smartphone Malware Using Data Mining Technology," in *Proc. of 20th International Conference on Computer Communications and Networks (ICCCN2011)*, pp.1-6, Jul. 31-Aug. 4, 2011.

[6] W. Enck, D. Octeau, P. McDaniel, and S. Chaudhuri, "A Study of Android Application Security," in *Proceedings of the 20th USENIX Security Symposium*, San Francisco, August, 2011.

[7] Yuan Wei, Jin Qin, "Safety-Driven Software Reliability Allocation in Medical Device Application," in *Proc. of Quality, Reliability, Risk, Maintenance, and Safety Engineering (ICQR2MSE)*, pp.105-109, Jun., 2011.

[8] A. Kumar Maji, Kangli Hao, S. Sultana, S. Bagchi, "Characterizing Failures in Mobile OSes: A Case Study with Android and Symbian," in *Proc. of International Symposium on Software Reliability Engineering (ISSRE)*, pp.249-258, Nov., 2010.

[9] M. Becher, F. C. Freiling, J. Hoffmann, T. Holz, S. Uellenbeck, C. Wolf, "Mobile Security Catching Up? Revealing the Nuts and Bolts of the Security of Mobile Devices," in *Proc. of Security and Privacy (SP)*, on 2011 IEEE Symposium, pp.96-111, May, 2011.

[10] Ashwin Chaugule, Zhi Xu and Sencun Zhu, "A Specification Based Intrusion Detection Framework for Mobile Phones," Applied Cryptography and Network Security, *Lecture Notes in Computer Science*, Vol.6715, pp.19-37, 2011.

[11] R. Mojdehrakhsh, Wei-Tek Tsai. S. Kirani, L. Elliott, "Retrofitting Software Safety in an Implantable Medical Device," in *Proc. of IEEE Software*, Vol.11, No.1, pp.41-50, Jan., 1994.

[12] Zhang Hong, Liu Binbin, "Integrated Analysis of Software FMEA and FTA," in *Proc. of International Conference on Information Technology and Computer Science 2009 (ITCS09)*, Vol.2, pp.184-187, Jul., 2009.

[13] M. Becher, "Security of smartphones at the dawn of their ubiquitousness," Ph.D. dissertation, University of Mannheim, Oct., 2009.

[14] Shuaifu Dai, Yaxin Liu, Tielei Wang, Tao Wei, Wei Zou, "Behavior-Based Malware Detection on Mobile Phone," in *Proc. of 6th International Conference on Wireless Communications Networking and Mobile Computing (WiCOM10)*, pp.1-4, Sept., 2010.

[15] N. G. Leveson and P. R. Harvey, "Analyzing Software Safety," in *Proc. of IEEE Transactions on Software Engineering*, Vol.SE-9, No.5, pp.569-579, Sept., 1983.

[16] D. Dagon, T. Martin, T. Starner, "Mobile Phones as Computing Devices: The Viruses are Coming!" in *Proc. of IEEE on Pervasive Computing*, Vol.3, No.4, pp.11-15, 2004.

[17] J. Wayne, S. Karen, "Guidelines on Cell Phone and PDA Security," in *Proc. of National Institute of Standard and Technology*, US Department of Commerce, pp.800-124, 2008.

[18] P. Zheng, L. M. Ni, "Spotlight: The Rise of the Smart Phone," in *Proc. of IEEE Distributed Systems Online*, Vol.7, No.3, Mar., 2006.

[19] A. Tansu, B. Christian and A. D. Schmidt. "A Probabilistic Diffusion Scheme for Anomaly Detection on Smartphones," *Lecture Notes on Information Security Theory and Practices. Security and Privacy of Pervasive Systems and Smart Devices*, pp.31-46, 2010.

[20] B. Scorgie, P. Veeraraghavan, S. Ghosh, "Early Virus Detection for Windows Mobile," in *Proc. of 9th Malaysia International Conference Communications (MICC2009)*, pp.295-300, Dec., 2009.

[21] C. F. Kemerer, B. S. Porter, "Improving the Reliability of Function Point Measurement: An Empirical Study," *in Proc. of IEEE Transactions on Software Engineering*, Vol.18, No.11, pp.1011-1024, Nov., 1992.

[22] A. Shabtai, Y. Fledel, Y. Elovici, "Securing Android-Powered Mobile Devices Using SE Linux," in *Proc. of IEEE Security & Privacy*, Vol.8, No.3, pp.36-44, 2010.

[23] Young Mo Kang, Chanwoo Cho, Sungjoo Lee. "Analysis of Factors Affecting the Adoption of Smartphones," in *Proc. of 2011 IEEE International Conference on Technology Management (ITMC)*, pp.919-925, Jun., 2011.

[24] S. Chandra, P.M. Chen, "Whither Generic Recovery from Application Faults? Fault Study Using Open-Source Software," in *Proc. of International Conference on Dependable Systems and Networks (DSN2000)*, pp.97-106, 2000.

[25] Peichang Wang, Junxing Zhang, Zhixue Chang, "Fault Tolerance of Multiprocessor-Structured Control System by Hardware and Software Reconfiguration," in *Proc. of International Conference on Mechatronics and Automation (ICMA07)*, pp.3745-3749, Aug., 2007.

[26] Y. M. Wang, Y. Huang, W. K. Fuchs, "Progressive Retry for Software Error Recovery in Distributed Systems," in *Proc. of The Twenty-Third International Symposium on Fault-Tolerant Computing (FTCS-23)*, pp.138-144, Jun., 1993.

[27] Y. Huang and C. Kintala. “Software Implemented Fault Tolerance: Technologies and Experience,” in *Proc. of the 1993 International Symposium on Fault-Tolerant Computing*, pp.2-9, Jun., 1993.

[28] Y. Huang, C. Kintala, N. Kolettis, N. D. Fulton, “Software Rejuvenation: Analysis, Module and Applications,” in *Proc. of the 23rd International Symposium on Fault-Tolerant Computing (FTCS-23)*, pp.381-390, Jun., 1995.

[29] A. Gopalan, S. Banerjee, A. K. Das, S. Shakkottai, “Random Mobility and the Spread of Infection,” in *Proc. IEEE of INFOCOM 2011*, pp.999-1007, Apr., 2011.

[30] B. S. Medikonda, P. S. Ramaiah, “Integrated Safety Analysis of Software-Controlled Critical Systems,” *ACM SIGSOFT Software Engineering Notes*, Vol.35, No.1, Nov., 2010.

[31] T. Maier, “FMEA and FTA to Support Safe Design of Embedded Software in Safety-Critical Systems,” in *Proc. of CSR 12th Annual Workshop on Safety and Reliability of Software Based Systems*, Bruges, Belgium, 1995.

[32] C. T. Alan and P. M. Steven, “Software Safety Analysis of a Flight Management System Vertical Navigation Function- A Status Report,” in *Proc. of IEEE the 22th Digital Avionics Systems Conference*, Indianapolis, pp.12-16, Oct., 2003.

[33] Tao Jianfeng, Wang Shaoping, and Yao Yiping, “Hybrid Method of Computer Aided FMECA and FTA,” *Journal of Beijing University of Aeronautics and Astronautics*, Beijing, China, Vol.26, No.6, pp.663-665, 2000.

[34] Wildan Toyib and Man-Gon Park, “Process Analysis of Digital Right Management for Web-Based Multicast Content,” *Journal of Korean Multimedia Society*, Vol.14, No.12, pp.1601-1612, Dec., 2011.

Myong-Hee Kim

e-mail : mhgold@pknu.ac.kr

She graduated with the B.E. in Info. and Comm. Engineering at the Dongseo University, and received M.S. degree in Computer Science and Ph. D. in Information Systems from the Pukyong National University, Rep. of Korea. From 2012-2013, She was a lecturer of the Dept. of Computer Science & Engineering, University of Colorado-Denver, USA. Also she was an adjunct professor of Graduate School of Education, Pukyong National University from 2011-2012. In recent, she is a lecturer of Information Systems, Graduate School, Pukyong National University. She served as an Assistant Faculty Consultant and a specialist in Information Communication Technology and Web-Based Multimedia Ttechnology for CPSC which is an Inter-Governmental International Organization for Human Resources Development in Asia and the Pacific Region. She has experience in development of e-Teaching and Learning System for the Ministry of Marine and Fisheries Affairs, Government of Korea. She is very confident in practical methodologies and tools utilization for development of Web-Based Teaching and Learning Systems and computer networking systems. She is also an expert in project management with software tools in development projects. She has embarked on consulting works in ICT and TVET areas for ADB, ILO APSDEP and UNESCO-UNEVOC.

Wildan Toyib

e-mail : wildantoyib@pknu.ac.kr

He received B.E. degree in computer science with Class Honors from Institut Teknologi Sepuluh Nopember in 2007. In 2012, he acquired M.S degree in Advanced Information Science and Technology, Graduate School, Pukyong National University, Republic of Korea. He has experienced in the IT field such as Team Lead of Programmer for Electronic Data Processing from 2007-2008 and as Senior IT Reporting for Chevron Corp. in Energy Component for ideSIDE server report environment IndoAsia Business Unit from 2008-2009. He worked for The Ministry of National Education, Republic of Indonesia with BERMUTU (Better Education Through Reformed Management Universal Teacher Upgrading) which is supported by World Bank and Netherlands Government from 2009-2010. He is member of IAENG and student member of IEEE. His research interests include safety and security of ubiquitous computing systems, ubiquitous sensor networks, software reliability engineering, data engineering, multimedia information processing technology and web and internet technology.

Man-Gon Park

e-mail : mpark@pknu.ac.kr

He is a head professor of the Dept. of IT Convergence and Application Engineering, College of Engineering, Pukyong National University, Republic of Korea since 1981. Also he was the president and vice president of the Korea Multimedia Society (KMMS). He served as the Director General and CEO of the Colombo Plan Staff College for Technician Education (CPSC) from 2002 to 2007, which is an intergovernmental international organization of 29 member governments for Human Resources Development in Asia and the Pacific Region. He served there also as a faculty consultant seconded by the Government of the Rep. of Korea as an expert in information systems development and ICT-based TVET systems from 1997 to 2001. He has been the visiting professor of Dept. of Computer Science, University of Liverpool, UK; exchange professor of Dept. of Electrical and Computer Engineering, University of Kansas, USA; visiting scholar of School of Computers and Information Science, University of South Australia; and visiting professor of Dept. Computer Science and Engineering, University of Colorado-Denver, USA (2012-2013). He was dispatched to Mongolia and People's Rep. of China by KOICA on various projects as information systems consultant. He has also embarked on various consulting works and conducted training programs in ICT on individual capacity for Korean groups of companies, governmental and non-governmental agencies and other institutions in Korea. His main areas of research are software reliability engineering, business process reengineering, Internet and web technology, multimedia information systems, and ICT-based HRD.