sustainability

Article

An Energy-Efficient Virtualization-Based Secure Platform for Protecting Sensitive User Data

Kyung-Soo Lim [1], **Jinho Park** [2] **and Jong Hyuk Park** [3,*]

[1] Intelligent Security Research Group, Electronics and Telecommunications Research Institute, Daejeon 34129, Korea; lukelim@etri.re.kr

[2] School of Software, Soongsil University, Seoul 07040, Korea; j.park@ssu.ac.kr

[3] Department of Computer Science and Engineering, Seoul National University of Science and Technology, Seoul 01811, Korea

* Correspondence: jhpark1@seoultech.ac.kr; Tel.: +82-2-970-6702

Received: 23 April 2017; Accepted: 30 June 2017; Published: 17 July 2017

Abstract: Currently, the exchange cycles of various computers, smartphones, tablets, and others have become shorter, because new high-performance devices continue to roll out rapidly. However, existing legacy devices are not old-fashioned or obsolete to use. From the perspective of sustainable information technology (IT), energy-efficient virtualization can apply a way to increase reusability for special customized devices and enhance the security of existing legacy devices. It means that the virtualization can customize a specially designed purpose using the guest domain from obsolete devices. Thus, this could be a computing scheme that keeps energy supplies and demands in balance for future sustainable IT. Moreover, energy-efficient virtualization can be the long-term and self-sustainable solution such as cloud computing, big data and so forth. By separating the domain of the host device based on virtualization, the guest OS on the segmented domain can be used as a Trusted Execution Environment to perform security features. In this paper, we introduce a secure platform to protect sensitive user data by domain isolation utilizing virtualization. The sensitive user data on our secure platform can protect against the infringement of personal information by malicious attacks. This study is an effective solution in terms of sustainability by recycling them for special purposes or enhancing the security of existing devices.

Keywords: secure platform; virtualization; hypervisor; inter-domain communication

1. Introduction

Since the advent of smartphones, daily life cannot be imagined without a mobile service. Smartphone users can manage their daily schedules, reply to e-mails, watch movies via a streaming service, and so forth. Most users no longer have to find and visit a bank to transfer their money and transact their financial business. They can also use e-map and e-navigation applications to find the fastest route to their destination. Likewise, the popularity of mobile devices has brought a paradigm shift from PC-based information technology (IT) services to future mobile services including business enterprises, mobile cloud computing, smart working, e-finance, smart vehicles, mobile health care, military service and others.

The exchange cycles of various IT devices such as smartphones and tablets have become shorter, as new high-performance devices continue to roll out rapidly. However, the existing legacy devices are not yet outdated or obsolete. Used smartphones are exported worldwide for sustainable IT. A forecast from International Data Corporation (IDC) estimates that the market for used smartphones will grow from 81.3 million devices in 2015 to 222.6 million units in 2020, representing a compound annual growth rate of 22.3% [1]. Therefore, if there were a way to retrieve these discarded devices and customize these for another purpose, this would be an efficient solution for the future sustainability of IT. For example, energy-efficient virtualization can offer a way to increase the reusability of these specially customized devices and enhance the security of existing legacy devices. Virtualization can provide customization for a specially designed purpose, using the guest domain from obsolete devices. By separating the domain of the host device based on this virtualization, the guest operating system (OS) on the segmented domain can be used as a Trusted Execution Environment to enable security features. It may, therefore, be a computing scheme that can keep energy supply and demand in balance for future sustainable IT. Moreover, energy-efficient virtualization may involve a long-term and self-sustainable solution such as cloud computing, big data and so forth. These will be key components for the realization of such a complex network and energy-aware devices. In recent years, numerous studies have expanded the use of virtualization technology for efficiency and sustainability purposes [2,3].

On the other hand, security and privacy issues are essential components of and necessary tasks within this reuse for special purposes. In particular, mobile platforms, as end-terminal and entry points to various services, are not only generated/stored user's sensitive and private information on the terminal, but they may also be sent and spread through the network. As we have shown in Figure 1, malicious attacks on mobile devices, and especially the Android platform, are rapidly increasing in the form of malware or exploits, such as smishing, app repacking attacks, app update attacks and so on [4]. Android's high market share and the openness of this platform have meant it has become the main target of mobile attacks by hackers [5]. Security countermeasures are therefore essential in reusing and recycling Android devices. A secure platform for mobile devices plays an important role in maintaining reliability and sustainability.

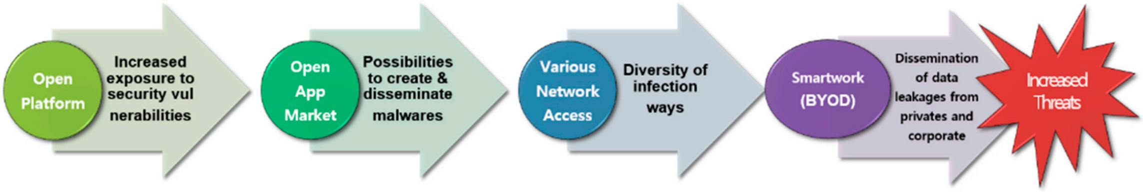

Figure 1. The increased threat to Android devices [6,7].

In view of the above security issues, we examine the current approaches to domain separation techniques, particularly for mobile security platforms. Currently, secure platforms based on domain separation have been widely studied in order to overcome these vulnerabilities, and especially the openness of the Android platform and app market [4–11]. The domain separation technique allows for the separation of different domains, that is, the normal domain for the host operating system and a secure domain for operating special-purpose services such as security features. The normal domain is mostly identical to the existing smart device environment, such as Android OS. The isolated and secure domain, however, protects a user's sensitive information and performs a trusted execution such as financial transactions, business applications, military service and so on [12,13]. Relevant and similar proposals will be described in related works.

Currently, the secure platforms using domain isolation techniques are monopolized by the smartphone or chipset makers such as Samsung KNOX and ARM TrustZone. On the other hand, academia continues to carry out mobile hypervisors such as KVM [14], Xen project [15] and others. There are still difficulties and obstacles, meanwhile, for developing the hypervisor according to specific hardware chipset, because it depends on the technical references of the chipset maker such as the ARM processor. However, the most advantage of mobile virtualization can configure the security platform for diverse purposes that operates independently of the smart device manufacturers. Thus, it will apply various legacy Androids devices without relying on a specific manufacturer and customize the guest domains for a special purpose such as a secure platform.

In this paper, we propose a secure platform with sustainability which protects sensitive user data utilizing mobile hypervisor as one of the domain separation technologies. The sensitive user data, such as phone history, which is related to personal privacy including contacts, call logs, and messages, can be stored in the secure domain, which is a virtual guest domain created by the mobile hypervisor. The secure domain is protected and secured by encrypting the user's sensitive information using various authentication mechanisms. Although security-enhanced Android platforms have recently been introduced, domain separation using a hypervisor is more secure than Android OS itself, since this approach can operate several OS environments and consolidate the security features within the isolated domain. Our proposed mobile security platform, (referred to here as the Trusted Mobile Zone or TMZ), provides a trusted execution environment, secure storage, a secure service and other features. In an Android environment with TMZ, the user's primary activities and sensitive information can be stored and encrypted in the TMZ security domain. The TMZ can use in various fields such as smart work, e-government, and e-financial transactions. It provides a secure domain via a mobile hypervisor, file system encryption, secure middleware, access control, a cryptography library, certification management and a security services API.

2. Related Work

First, we describe relevant and similar techniques involved with domain separation technology. The secure domain is used to store a user's phone and address book, and to schedule data for protecting the user's sensitive information and providing a trusted execution environment for secure services, mobile banking, smart work services, e-government, and so forth. The techniques used in this domain separation can be divided into a hardware chipset-based separation technique, hypervisor-based mobile virtualizations [16], and a logical domain separation technique [17]. The mobile hypervisor-based domain separation can separate a single physical mobile device into several domains. The isolated and secure domain provides a trusted execution environment by running on a different operating system. Each virtual machine communicates only through the authenticated channel [18]. The Android OS plays the role of the host (general) domain, while a guest (security) domain runs on a different installed operating system, such as a real-time OS for an embedded system.

Mobile Trust Mobile (MTM) is a platform within the mobile environment for a trusted execution operating environment, which is based on the computer-based security technology Trusted Platform Module (TPM) presented by the Trusted Computing Group. It is a cryptoprocessor, which is configured to support cryptography features in hardware-related library module internally. ARM TrustZone, supporting the ARM processor since the version 6 architecture, provides a mobile security environment at the level of the hardware chipset. It supports two operation modes with the normal and secure mode. It is also referred to as the normal and secure world. According to their feature within applications, the ARM processor allows that which the operation mode needs. It also provides a secure booting and isolated memory for blocking against data leakage into another operation mode. The secure world operates on secure processors or secure executions (SE) supported by TrustZone. It enables to communicate through the authorized channel created by the virtual machine or operation mode. Recently, Samsung Electronics announced the KNOX platform. It is an opportunity for people to get to know the domain separation technique in the market and research area. The KNOX smartphone technology separates the security zone against the common user area in Android smartphones. The domain separation in KNOX technically is based on TrustZone, not mobile virtualization, and this platform is only available for Samsung smart devices.

The most widely acknowledged secure platform based on a smart device is the US Army's Nett Warrior system. The US Army has integrated a Samsung Galaxy Note II smartphone into the Nett Warrior (NW) system to enhance situational awareness in the battlefield. The NW system introduced as part of the Army's tactical network modernization. In the NW system, the smartphone mounts on the chest of a field soldier as an end-user device. It provides the soldier with enhanced mission planning, field monitoring, communication and situational awareness during combat operations. It utilizes only the smartphone hardware, but the US Army develops and provides a defense information system, which includes mobile platform, network communication and certain military applications by themselves. Moreover, the US Army's Joint Battle Command-Platform (JBC-P) is a networked battle command-and-control information system which is connected to various military devices such as the NW system. This is the first attempt to develop a framework based on Android devices under US Army supervision for creating suites of military applications for tactical operations.

The mobile healthcare sector is one of an emerging set of mobile applications and services. It allows individuals to access web-enabled mobile devices for managing their healthcare conveniently. Even though this technology makes m-health possible, many open issues still exist within the mobile healthcare environment, such as the security of electronic data transactions, mobile user authentication and secure data storage on a mobile device with privacy protection [19,20].

Connected car services have become popular due to the expansion of smart devices. These provide Internet services and vehicle control and monitoring applications for comfortable and convenient driving. Connected car services are based on smart devices which operate built-in car console systems by connecting to a user's smartphone, such as Android Auto, Apple CarPlay and so on. Cars in the near future are likely to evolve the connected car utilizing the smart device features such as a network connection, social network service (SNS), navigation, driving control, and entertainment applications. In this way, the infringement of smart devices can jeopardize the security of the connected car [21].

For above reasons, mobile devices have become an attractive target for malicious attacks. Security countermeasures including malicious activity detection, secure platforms secure file system, and other technologies have been receiving high levels of attention and are widely studied nowadays in order to overcome these threats. In particular, secure platforms based on domain separation have been widely studied with the aim of overcoming these vulnerabilities, and especially the openness of the Android platform and app market.

3. The Mobile Hypervisor-Based Secure Platform

The authors have developed a secure platform based on the mobile hypervisor, known as TMZ, and several papers related to this secure platform have previously been published in international conferences and journals [22,23]. The TMZ platform provides a trusted runtime environment, secure storage and various security services that operate within a separately designed security domain. Using the domain separation provided by the mobile hypervisor, the specially customized Android device is divided into two separate domains, the normal domain (ND) and the security domain (SD).

Android OS runs within the ND, while the TMZ platform is runs within the SD as a guest domain supported by virtualization. The main security activities of the TMZ platform are performed in the secure domain, which is run by MicroC/OS-II (also presented by µC/OS-II), a popular embedded real-time OS, as shown in Figure 2. The mobile hypervisor of TMZ is based on ViMo (Virtualization for Mobile), developed by ETRI. ViMo is a micro virtual machine monitor for ARM mobile systems, which enables multiple operating systems to be run simultaneously on a single mobile system, and is based on full or para-virtualization for mobile systems. The recently developed version of ViMo can be installed on any commercial 64-bit Android device without support from the manufacturer, and achieves portability by using pure software virtualization while preserving high performance [24]. The primary contribution of the ViMo design is to put the guest OS and the hypervisor together into a single address space, which results in avoidance of the address space compression problem and reductions in the major virtualization costs, using a 32-bit compatible mode [25]. It is currently evaluated on the Google Nexus 6P. The following section describes the architecture of the TMZ platform and the way in which the two domains communicate.

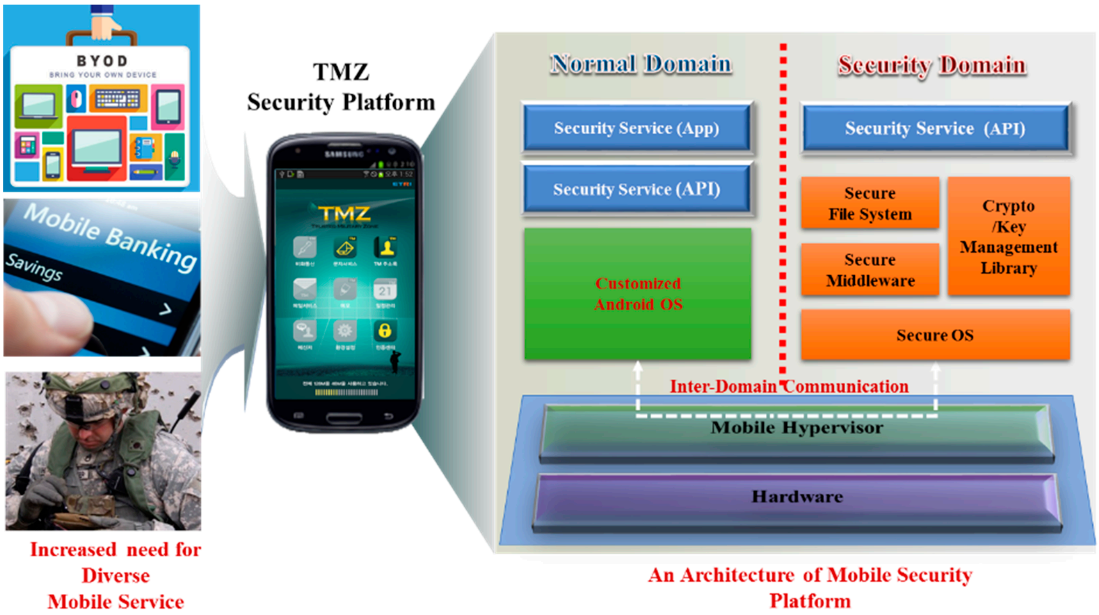

Figure 2. Mobile hypervisor-based secure platform (TMZ: Trusted Mobile Security Zone).

3.1. System Architecture

The TMZ platform provides more enhanced secure functions, compared with smart devices in general. The following are the main subsystems of the TMZ platform:

- The Mobile Hypervisor Subsystem (MHVS)
- The Inter-Domain Communication Subsystem (IDCS)
- The Mobile Secure Platform Subsystem (MSPS)
- The Secure Service API Subsystem (SSAS).

The mobile hypervisor is an essential part of the secure platform for separating and isolating the secure domain against unauthorized or illegal access. The MHVS within the mobile hypervisor engine can provide a trusted execution environment using domain separation. MHVS emulates a CPU, memory and hardware resources for the guest domain, and also provides CPU virtualization, memory virtualization and interrupt virtualization for execution of the virtual machine. It also includes a feature allowing inter-domain message exchange via communication drivers, which exists in both domains. Using these features, MHVS provides domain isolation and an inter-domain channel. The Inter-Domain Communication Subsystem (IDCS) performs the roles of various management mechanisms for message handling, communication and multi-channel handling between the two domains. IDCS locates both the normal domain and secure domain, and is in charge of managing the inter-domain channels, multi-client concurrent connections, secure sessions, message handling and so forth. The Trusted Channel Manager (TCM) provides handling multi-client concurrent connections.

The Mobile Secure Platform Subsystem (MSPS) consists of several modules: the Secure Filesystem Module (SFS); the Authentication/Access Control Module (AAC), and; the Encryption/Key Management Module (EKM), as shown in Figure 3. Descriptions of some of these modules have previously been published in international journals or conferences. For example, Park et al. have presented SecureDom, which is focused on data-centric security and aims to protect stored data and process access control within the secure domain [26]. Based on TMZ architecture, the SecureDom involves subsystems such as the SFS, AAC, and EKM subsystems. In this chapter, however, the focus is on domain separation and inter-domain communication subsystems, in order to show how to exchange a message between two domains.

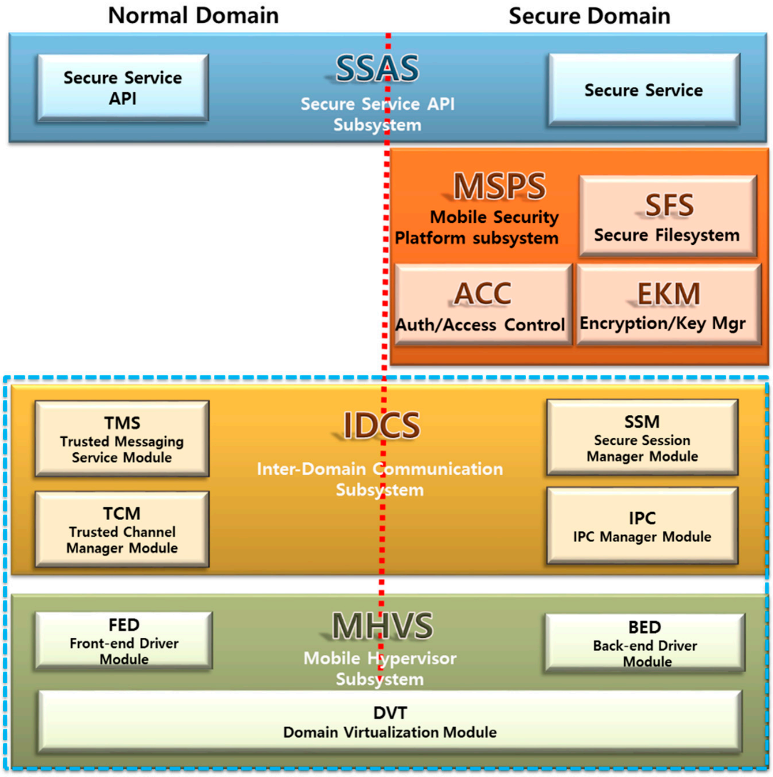
Normal Domain
Secure Domain
Secure Service API
SSAS
Secure Service API Subsystem
Secure Service
MSPS
Mobile Security Platform subsystem
SFS
Secure Filesystem
ACC
Auth/Access Control
EKM
Encryption/Key Mgr
TMS
Trusted Messaging Service Module
IDCS
Inter-Domain Communication Subsystem
SSM
Secure Session Manager Module
TCM
Trusted Channel Manager Module
IPC
IPC Manager Module
FED
Front-end Driver Module
MHVS
Mobile Hypervisor Subsystem
BED
Back-end Driver Module
DVT
Domain Virtualization Module

Figure 3. Architecture of the TMZ platform.

The Secure File System module (SFS) provides a secure file system and management feature. Secure file management is important in order to protect confidential or sensitive information in case of a lost or stolen device. The secure storage mechanism for the secure platform is based on domain separation and is implemented by the SFS. It was carefully designed to handle small pieces of text data as sensitive information, since the mobile device has limited memory and CPU resources and constraints from middleware regarding domain separation [24]. The secure file management system consists of a secure file system with file encryption, secure deletion for protecting file recovery, continuous file defragmentation within a particular maintenance period, and so forth. Moreover, the secure file system is located and operates on the volatile memory of the smart device, with the characteristics of a Type 2 hypervisor. This means that when the device is powered off, the file system is wiped and is then unable to recover user data. TMZ also provides the expected dumping/loading of a file system image when it prepares for rebooting. The file system image can be saved on the storage device of the Android platform for backup (the capacity of file system is restricted to 64 megabytes). owever, this dump image file is encrypted by a cryptography algorithm such as AES 128 bit.

The Authentication/Access Control module (ACC) is an important feature in TMZ which protects against unauthorized access. It supports two-factor authentication by the user and application requesting security services within the secure domain. Moreover, TMZ checks and verifies the integrity of TMZ applications periodically as an essential part of its access control features. Authentication management in TMZ complies with the FIPS-196 standard to provide mutual authentication with remote servers for user or device confirmation. This feature supports electronic authentications based on Korean Certificate-based Digital Signature Algorithm (KCDSA) and RSA digital signatures.

The Encryption/Key Management module (EKM) provides a data encryption/decryption function to protect inter-domain messages and the secure file system. EKM also supports cryptography algorithms when it requests other functions involved in security. The cryptography library consists of a symmetric key, public key, hash algorithm and authentication library. It is a core part of the operation of each subsystem such as file encryption and digital signatures for certificates. Finally, the Secure Service API Subsystem (SSA) provides APIs for developing secure functions for implementation of the TMZ app.

3.2. The Mobile Hypervisor Subsystem

The Mobile Hypervisor Subsystem (MHVS) provides a secure domain for security features through mobile virtualization, which are separated from the normal domain on an Android-based smart device. It allows the exchange of messages using inter-domain communication between these two domains. MHVS is composed of three modules and interfaces with IDCS. This paper does not focus on the mobile hypervisor, but rather on the domain separation-based secure platform. Thus, we do not intend to describe advances in the mobile hypervisor, including the virtual machine monitor, since ViMo has been described above. Instead, a description is given here of the way in which the two domains communicate within the TMZ platform.

The Domain VirTualization module (DVT) emulates a CPU, memory and hardware resources for the guest domain. It also provides CPU virtualization, memory virtualization and interrupt virtualization for execution of the virtual machine. The shared memory function shares memory between several domains for inter-domain communication. The event channel function triggers notification of certain events to other domains. The hypercalls function invokes the functionality of the DVT from the secure domain.

Domain Virtualization Module (DVT)

- Event type and event handler entry management
- Call event handler
- Shared memory management
- Processing shared memory connect/disconnect
- Shared memory management
- Hypercall type and hypercall handler entry management
- Call hypercall handling

The Front-End Driver module exists to support the features of the DVT in the normal domain, to enable inter-domain communication, and provides functions for handling the sending and receiving of messages. These functions process received messages from the TCM and send response messages back to the TCM when the secure service operation is complete. It sends these received messages to the secure domain and transfers response messages to the normal domain via the shared memory functions of the DVT. It also manages the events occurring in the security domain, including event registration and handling. Most of events occurs when the message is received from the secure domain.

Front-End Driver Module (FED)

- Event handler registration management
- Event handling
- Shared memory connection management
- Message transmission and reception via shared memory

The Back-End Driver module exists to support the features of the DVT in the secure domain enabling inter-domain communication, and provides functions for handling the sending and receiving of messages. These functions process messages received from the DVT and send response messages back to the DVT. This serves to receive the message from the secure domain and transfer the resulting messages to the secure domain via the shared memory functions in the DVT. It also manages the events that occur in the security domain, including event registration and handling. An event can occur when a response message is sent to the normal domain via the DVT, when the operation of the requested secure service is complete.

Back-End Driver Module (BED)

- Event sending
- Event handler registration management
- Shared memory allocation management
- Message transmission and reception using shared memory

Figure 4 illustrates the message flows in the Mobile Hypervisor (MHV) and Inter-Domain Communication (IDC) subsystems. It shows the inter-domain message process via the interface module between the IDC and MHV subsystems. When a TMZ app requests a security service from the secure domain, the TMS module in the Secure Service API (SSA) subsystem generates a message, which is sent to the TCM using UNIX Domain Socket (UDS) socket communication. TCM receives this and writes it to the message queue driver of the FED module (mq-front). The mq-front module connects with the shared memory in the DVT and writes message data to an allocated memory space. The shared memory management in the DVT module performs data transmission in a similar way for inter-domain communication.

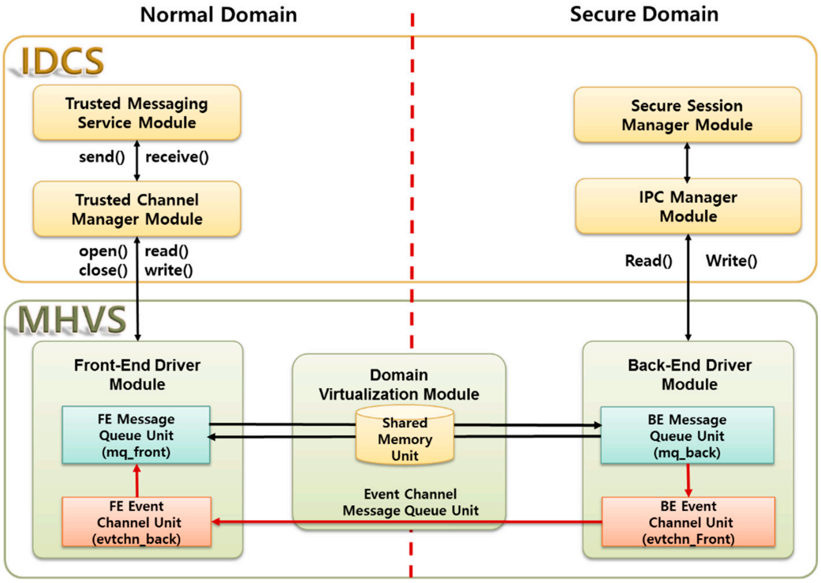
Normal Domain
Secure Domain
IDCS
Trusted Messaging Service Module
send()
receive()
Trusted Channel Manager Module
open()
close()
read()
write()
Secure Session Manager Module
IPC Manager Module
Read()
Write()
MHVS
Front-End Driver Module
FE Message Queue Unit (mq_front)
FE Event Channel Unit (evtchn_back)
Domain Virtualization Module
Shared Memory Unit
Event Channel Message Queue Unit
Back-End Driver Module
BE Message Queue Unit (mq_back)
BE Event Channel Unit (evtchn_Front)

Figure 4. Structure of the Mobile Hypervisor and Inter-Domain Communication subsystems.

The Back-End Driver Module of the secure domain receives the data from the shared memory module in the DVT and transfers the data to the message queue driver in the BED (mq-back). Then, the IPC manager of the IDC reads the received data from the mq-back module and transforms the message format. Next, it transfers the received message to the SSA for execution of the security services. Once the requested services are complete, a response message is created by the IDC and written to the mq-back driver in the BED module. This data is passed to the shared memory space in the DVT module.

When the data transfer to the allocated shared memory is complete, the event handler of the BED module (event-front) requests an event notification from the DVT module using hypercall. The DVT module sends the event to the event handler of the normal domain (event-back) to let it know that the transfer is complete. The FED module reads the message from the message queue and sends it to the TCM. The TCM receives the incoming message and transmits it to the TMZ app. Finally, the TMZ app reads the response message and checks the result of the requested operation.

3.3. The Inter-Domain Communication Subsystem

The Inter-Domain Communication Subsystem (IDCS) manages message communication between the two domains, so that TMZ apps in the normal domain can request the security services provided by the security domain. IDC makes it possible to utilize the domain communication channel with the Front-End Driver in the normal domain and Back-End Driver in the security domain, and manages secure sessions for the TMZ service apps to support multiple accesses. IDC is locates in both the normal and the secure domains, and is in charge of managing inter-domain channels, client concurrent connection management, session management, message handling and so forth.

IDCS is composed of the Trusted Messaging Service (TMS) and Trusted Channel Manager (TCM) module in the ND, and the Secure Session Manager (SSM) and Inter-Process Call Manager (IPC) modules in the SD. TMS is in charge of message marshaling/unmarshaling for the inter-domain message format, and the sending and receiving this message to and from the TCM. The TMS is implemented as a shared library with Android NDK to enable its use within TMZ apps. As described above and shown in Figure 4, the requested service from a TMZ app transforms the message in the TMS. TCM operates like a UNIX Domain Socket (UDS) socket server to manage multi-client concurrent access, which transfers it to the security domain. The IPC in the SD is the main entry point for the operation of security tasks in the microC/OS-II environment, including the message process and security procedure calls. The SSM in the SD manages the status of security sessions for TMZ app clients.

Trusted Messaging Service Module (TMS)

- Provides secure service API interfaces for TMZ apps
- Generates messages
- Transforms parameters of service API to message format
- Processes message transfer and reception

The TMS interfaces with the TMZ app to request security services for sending and receiving messages. It generates a message for use in the security domain by transforming the parameters of the security service API. The generated message is transmitted to the TCM via a socket connection. When the message is received, the TMS converts its parameters for the secure service API using the reverse order of the transmission process.

The message processing mechanism in the TMS uses an abstraction of the required parameters for the security service API calls. In the case of the transmission process, it transforms a variable of each parameter of API to parameter format in the message structure, such as the data type, the direction of the request, the call type and the values of the variable by abstraction, using data serialization and marshaling, as shown as Figure 5. The generated message is transferred to the SD through the MHVS using the TCM. The message received in the SD translates these parameters using the same process as TMS; it then calls the requested security service.

Trusted Channel Manager Module (TCM)

- Manages multiple app connections
- Manages multiple channels using logical virtualization
- Writes (transmits) messages to the FED and reads (receive) messages from FED

Currently, the FED provides a single channel for inter-domain communication. Even though the FED physically provides a single channel, the TCM enables multiple apps to connect concurrently using a logical multi-channel function. The TCM is therefore located between the client app and the TMS and FED, as shown in Figure 3. The TCM consists of the channel manager and ID manager, as shown in Figure 6. The channel ID issued for each client app is used to distinguish which app is requesting the secure service. The virtualized single channel performs the sending and receiving of messages from the requesting apps by associating them with the corresponding ID. It also has functions for writing (transmitting) and reading (receiving) these messages through an inter-domain message pipe which interfaces with the FED.

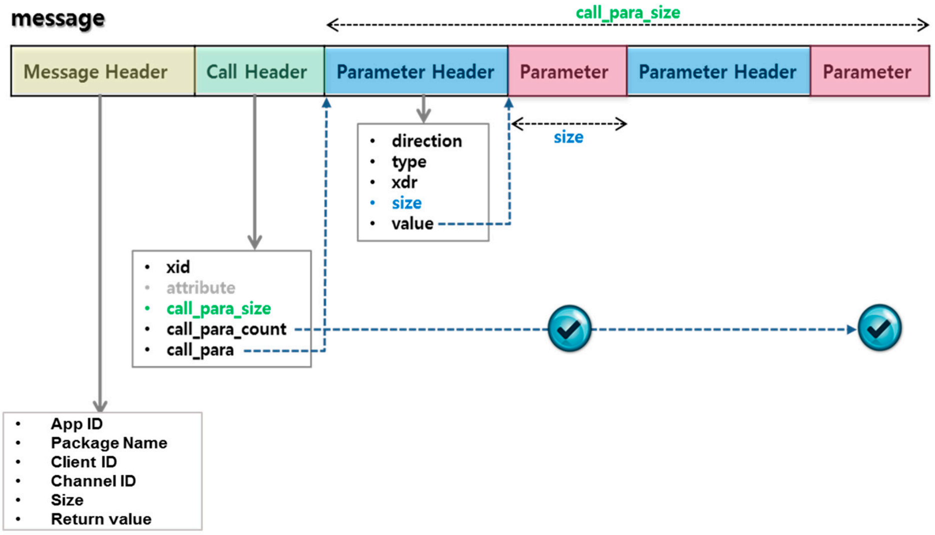

Figure 5. Structure of the inter-domain message format.

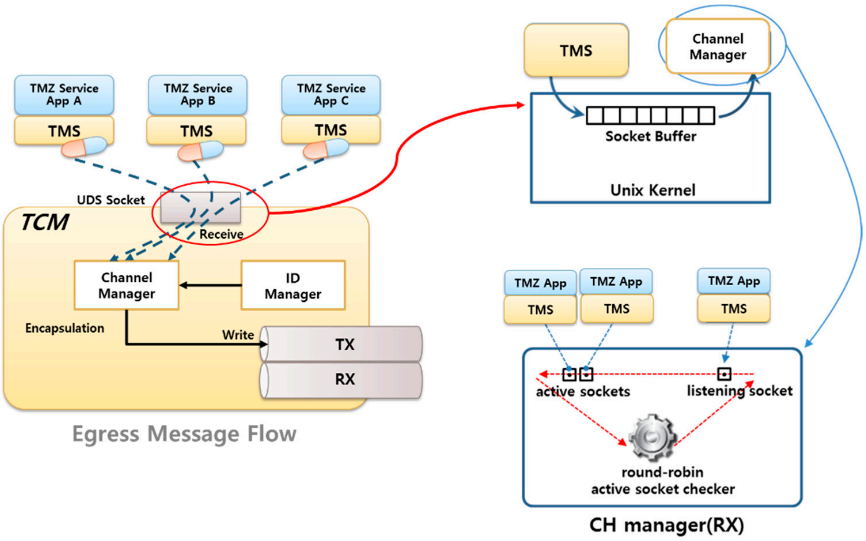

Figure 6. The multi-channel mechanism in the Trusted Channel Manager (TCM).

As described above, the channel manager of the TCM handles requests from multiple apps by allocating the channel ID to each app, as issued by the ID manager. TCM also provides read/write functions for message processing using the inter-domain message pipes, which are established by the FED.

In the outward message path on the left-hand side of Figure 7, when the app requests security services in the normal domain, the TMS sends a message to the TCM. The TCM receives it, and the ID manager inserts a unique ID for the requesting app. The channel manager transfers (writes) the message to the inter-domain message pipe. Conversely, the inward message flow is shown on the right-hand side of Figure 7. When the requested security procedures in the SD are complete, an acknowledgment message will be sent via the inter-domain message pipe. The channel managers verify the ID to find out which app has requested this, and it is finally sent to the appropriate channel.

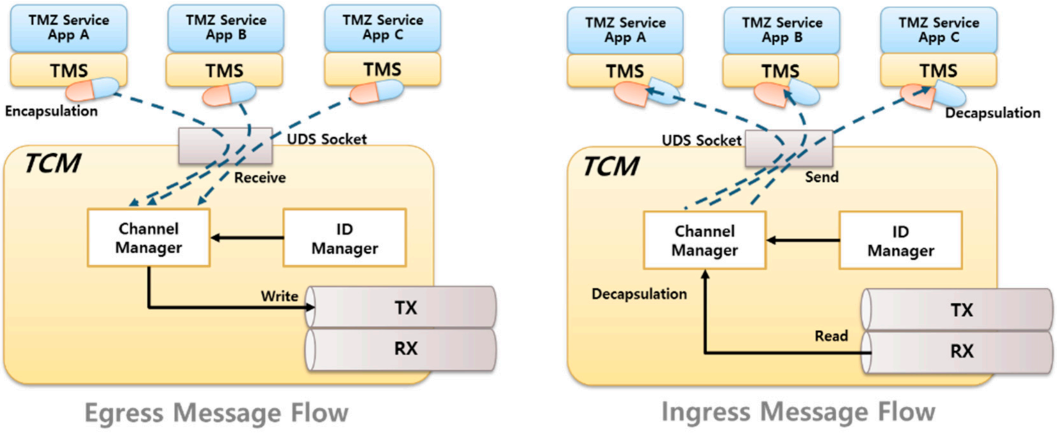

Figure 7. The multi-channel mechanism in the TCM.

In the case of the SD, the Inter-Process Call manager module (IPC) is the main service for carrying out security tasks in the uC/OS environment, including message handling and security procedure calls. It operates as the main function of the security system and acts as a bridge to connect with the other subsystems of the SD. Thus, as the first entry point of the SD, the IPC passes data to each subsystem such as access control, security services, session manager and others.

Inter-Process Call Manager Module (IPC)

- Reads (receives) messages from the BED and writes (transmits) messages to the BED
- Transforms the message format to parameters of security procedures in the SD
- Provides an interface with security procedures
- Calls the requested security procedures

The IPC provides functions for reading and writing the message from the BED. In this regard, the features of the IPC which are related to message processing are similar to those of the TMS and TCM in the ND. The IPC decomposes the received message and checks the reliability of the requesting app. Through the interface with the ACC, the IPC checks the reliability of the requesting app, using app authentication and resolution of access control to determine whether or not the requested app is trusted. The result of this authentication is recorded in the Secure Session Manager (SSM) module. The functions of the SSM are described in the next paragraph. After all processes involved in reliability verification are complete, the IPC checks the ID of the requested service from the security procedure list in SSAS and calls the security procedures. The transmission of the result message to the ND is described above.

Secure Session Manager Module (SSM)

- Provides Secure Service API interfaces with TMZ apps
- Generates messages
- Transforms parameters of the service API to message format
- Processes message transfer and reception

The Secure Session Manager Module (SSM) in the SD manages the status of security sessions with respect to the client apps, according to the access control policies of the app authentication process. The status of the secure session is described in Table 1 below. When a TMZ app requests a security service, the TCM creates a channel between the two domains. When the channel creation and the verification of the trusted app are complete, the session status of this client app is marked “Created” in the session table in the SD. To use the security services, the app needs an access permission such as PIN authentication. When the ACC grants each access permission, the SSM changes the session status from “Created” to “Authorized”. When the secure session has been established (“Authorized”), the secure session is assigned and held only for a limited period of time. When this period has elapsed, the client app is required to perform a re-authentication process. If it does not perform this re-authentication, the corresponding session will terminate (“Timeout”).

Table 1. Session status for a client app.

Session Status	Description
Created	Channel is created
Authorized	Secure session is established
Session Closed	Session is closed
Closed	Channel is closed
Timeout	Session has expired

4. Implementation and Experiments

This chapter describes the implementation of and experiments on the TMZ platform. A prototype of the TMZ platform and the TMZ security service apps were implemented. The prototype device for the TMZ platform was a Samsung Galaxy S3 with a Samsung Exynos4412 Cortex-A9 chipset and a 2GB DDR2 SDRAM.

As described above, the mobile hypervisor of the TMZ platform is ViMo. The current 32-bit version of ViMo was installed on the Galaxy S3 is. We implemented and evaluated our platform using one of the latest commercial mobile phones, the Google Nexus 6P. Since the TMZ operates on the guest domain provided by the mobile hypervisor, the performance of the entire system will inevitably be reduced. Moreover, the round-trip time (RTT) of the message TX/RX through the inter-domain channel takes more time; this is an expected result in comparison with message operation within a single domain. Two experiments were therefore carried out, that is, the determination of the virtualization overhead and the RTT of an inter-domain message.

The prototype service apps of the TMZ provide a military security service, including secure SMS and MMS, contacts, camera, gallery, secure voice communication and emergency situation notices, as shown in Figure 8. These app services operate in Android OS. However, all of the data in these apps are transferred from the secure domain when it needs to view the user actions. The TMZ platform and service apps have been installed and tested on the Samsung Galaxy S3 LTE, and those are being tested on the Google Nexus 6P currently as commercial smart devices.

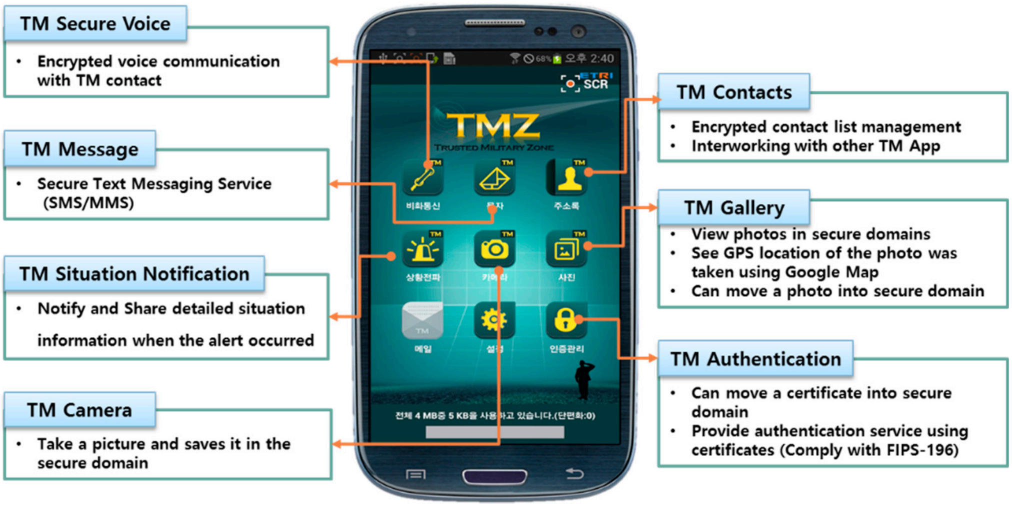
TM Secure Voice
• Encrypted voice communication with TM contact
TM Message
• Secure Text Messaging Service (SMS/MMS)
TM Situation Notification
• Notify and Share detailed situation information when the alert occurred
TM Camera
• Take a picture and saves it in the secure domain
TM Contacts
• Encrypted contact list management
• Interworking with other TM App
TM Gallery
• View photos in secure domains
• See GPS location of the photo was taken using Google Map
• Can move a photo into secure domain
TM Authentication
• Can move a certificate into secure domain
• Provide authentication service using certificates (Comply with FIPS-196)
오후 2:40
ETRI SCR
TMZ
TRUSTED MILITARY ZONE
비화통신
문자
주소록
상황전파
카메라
사진
메일
설정
인증관리
전체 4 MB중 5 KB을 사용하고 있습니다.(단편화:0)

Figure 8. Implementation of the TMZ security service applications.

The mobile services based on the TMZ are similar to the essential features of the general smartphone. The TMZ provides a text messaging service including SMS and MMS, a contacts list, a camera, a photo gallery, secure voice communication, and an emergency situation notice app. These services were developed based on the recommendations of an advisory committee consisting of Korean military officials. For example, the TM contacts form an exclusive address book app for the TMZ environment, which encrypts contact lists on the secure file system and plays a part in the main role of interfacing with other services such as text messaging, secure voice communication and so forth. TM messages are similar to the general messaging app in regard to sending and receiving messages. This module operates and connects with the special-purpose messaging server for the TMZ environment.

The experiments on an overhead test of mobile hypervisor were performed using Dhrystone 2.1, a popular benchmark for CPU performance measurement. These demonstrate how the virtualization overhead reduces CPU performance compared with no virtualization. Most of the prior research papers related to enhancing virtualization techniques use Dhrystone for this purpose. The number of runs in Dhrystone was 1,000,000,000, and the result is given in Dhrystone points (MIPS, million instructions per second). We performed this experiment 10 times, and the average value is presented in Table 2.

Table 2. Test for Virtualization Overhead.

	Non-Virtualized Android Device	Virtualized Android Device	Virtualization Engine Overhead
Dhrystones per second	1,378,074	1,135,439	–
Rate (%)	100	82.39	**17.61**

As shown in Table 2, the non-virtualized and virtualized Android devices require 1,377,853 and 1,110,699 MIPS, respectively. The point difference value between the two devices is therefore 267,154. This means that the virtualized Android device has a falloff in system performance of 18.39% as compared with the non-virtualized device; this an unavoidable side effect which occurs in every system using virtualization. However, in order to enhance security, an appropriate level of performance reduction may in fact be as much as the average user will tolerate. If the latest version of ViMo is installed on a brand new mobile device, this reduction will be improved.

The following table shows the file acquisition time using inter-domain communication between two domains according to file types such as text messages, contact lists, and photo images. It shows the acquisition time according to file type and size. As shown in Table 3, the large file (an image) takes longer than the smaller file types. The reason for the differences in these experimental results is the limitation imposed by the shared memory size (4 KB) provided by the hypervisor. The current version of ViMo, installed on a Samsung Galaxy S3, formed a prototype for testing the feasibility of the development of secure mobile platforms. Details of the experiments on the performance of the IDCS and MSPS are described by Park et al. [23,26]. To transmit and receive a large file, the system needs to exchange large numbers of messages repeatedly. Future versions of ViMo are expected to improve performance in order to solve this problem.

Table 3. Message exchange time for various file types.

Type	File Size	Acquisition Time (Averaged over 100 Runs)
Text messages	3504 bytes	0.031 s
Contact lists	520 bytes	0.017 s
Image files	654,532 bytes	3.602 s

As can be seen from the implementation and experiments, virtualization technology can be used as a way to increase reusability for a specially customized device and to enhance the security of existing legacy devices from the perspective of sustainability. Our suggested study can reuse an old-fashioned device as a special-purpose device, despite the use of an obsolete device such as a Samsung Galaxy S3.

5. Conclusions and Future Work

According to IDC, the market for used smart device will grow 222.6 million units in 2020. This means that used smart device still have capabilities to use and are not obsolete and fit to be discarded. However, the security features must be enhanced to overcome known vulnerabilities. If mobile virtualization can be a role for security enhancement, it can reuse an old-fashioned device to emulate a special-purpose device.

In this paper, we introduce a secure platform, called TMZ (Trusted Mobile Zone), to protect a user's sensitive information using domain isolation based on virtualization. By separating the domains of the host device based on virtualization technology, the guest OS on the segmented domain can be used as a trusted execution environment for performing security activities. The TMZ can be a reliable security solution for overcoming this problem since it supports domain separation, a secure file system, secure middleware, multi-factor access control, a cryptography library, authentication management and a secure service API for TMZ app development. Although a security-enhanced Android platform has been introduced recently, domain separation using a hypervisor is more secure than the Android OS itself, as it can operate several different OS environments simultaneously and consolidate security features in the isolated domain. The TMZ can be utilized in various fields such as smart work, e-government and e-financial transactions. The TMZ platform and apps are prototype solutions for ongoing research into the development of military-grade security mobile solutions based on a secure platform using a mobile hypervisor. It is highly likely that the TMZ platform will be a major target for malicious attacks; to overcome these threats, the TMZ has undergone various penetration tests and vulnerability checks, and the issues discovered in this way have been addressed.

For future work, the authors have researched techniques that automatically move a user's sensitive information to the secure domain, using a relationship calculation based on the order of importance of the user's history data. This approach has the advantage that the user does not move private information to secure domain manually. The proposed idea is that the TMZ can automatically move and store the user's sensitive information from the Android OS (the normal domain) to the security domain (the TMZ) based on the relationship calculation, arranged in order of importance of the user's history data. The relationship points are analyzed using the contacts which have high frequency, calculated based on the numbers of sent/received text messages, called/received contacts and the names mentioned by the user as part of social activities, and so forth.

Acknowledgments: This work was supported by Institute for Information & Communications Technology Promotion (IITP) grant funded by the Korea government (MSIP). (2017-0-00207, Development of Cloud-based Intelligent Video Security Incubator Platform).

Author Contributions: Kyung-Soo Lim: research for the related works, analyzing of the proposed model and drafting the article, acquisition of data, analysis; Jinho Park: research for the related works, and readability, grammar, spelling checks of the article; Jong Hyuk Park: total supervision of the paperwork, review, comments, and assessment, and so forth.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Worldwide Market for Used Smartphones Forecast, International Data Corporation (IDC). Available online: http://www.idc.com/getdoc.jsp?containerId=prUS41929916 (accessed on 21 November 2016).
2. Park, J.H.; Kim, H.-W.; Jeong, Y.-S. Efficiency Sustainability Resource Visual Simulator for Clustered Desktop Virtualization Based on Cloud Infrastructure. *Sustainability* **2014**, *6*, 8079–8091. [CrossRef]

3. Kim, H.-W.; Park, J.H.; Majigsuren, D.; Jeong, Y.-S. Efficient Sustainable Operation Mechanism of Distributed Desktop Integration Storage Based on Virtualization with Ubiquitous Computing. *Sustainability* **2015**, *7*, 7568–7580. [CrossRef]
4. Jeeva, S.C.; Rajsingh, E.B. Intelligent phishing url detection using association rule mining. *Hum.-Cent. Comput. Inf. Sci.* **2016**, *6*, 10. [CrossRef]
5. Jaap-Henk, H.; Bart, J. Increased security through open source. *Commun. ACM* **2007**, *50*, 79–83.
6. SANS Institute InfoSec Reading Room, Data Leakage Landscape: Where Data Leaks and How Next Generation Tools Apply. Available online: https://www.sans.org/reading-room/whitepapers/analyst/data-leakage-landscape-data-leaks-generation-tools-apply-34695 (accessed on 10 July 2017).
7. Koh, E.B.; Oh, J.; Im, C. A Study on Security Threats and Dynamic Access Control Technology for BYOD, Smart-work Environment. In Proceedings of the International Multi Conference of Engineers and Computer Scientists (IMECS 2014), Hong Kong, China, 12–14 March 2014; Volume 2.
8. Frenzel, T.; Lackorzynski, A.; Warg, A.; Hartig, H. ARM TrustZone as a Virtualization Technique in Embedded System. In Proceedings of the 12th Real-Time Linux Workshop, Nairobi, Kenya, 25–27 October 2010.
9. Colp, P.; Nanavati, M.; Zhu, J.; Aiello, W.; Coker, G.; Deegan, T.; Loscocco, P.; Warfield, A. Breaking Up is Hard to Do- Security and Functionality in a Commodity Hypervisor. In Proceedings of the 23rd ACM Symposium on Operating Systems Principles, Cascais, Portugal, 23–26 October 2011.
10. Vasudevan, A.; Owusu, E.; Zhou, Z.; Newsome, J.; McCune, J. Trustworthy Execution on Mobile Devices: What security properties can my mobile platform give me? In Proceedings of the International Conference on Trust and Trustworthy Computing, Vienna, Austria, 13–15 June 2012; pp. 159–178.
11. Gaur, M.S.; Pant, B. Trusted and secure clustering in mobile pervasive environment. *Hum.-Cent. Comput. Inf. Sci.* **2015**, *5*, 32. [CrossRef]
12. Kim, Y.; Lee, Y.; Kim, J. TeeMo: A Generic Trusted Execution Framework for Mobile Devices. In Proceedings of the International Conference on Computer, Networks, Systems, and Industrial Applications(CNSI), Jeju Island, Korea, 16–18 July 2012; pp. 579–583.
13. Lim, K.-S.; Park, S.-W.; Kim, J.-N.; Lee, D.-G. Functional Considerations in Military-Grade Security Platform Using a Mobile Hypervisor. *Comput. Sci. Appl.* **2015**. [CrossRef]
14. Oh, S.C.; Kim, K.H.; Koh, K.W.; Ahn, C.-W. ViMo (virtualization for mobile): A virtual machine monitor supporting full virtualization for ARM mobile systems. In Proceeding of the Advanced Cognitive Technologies and Applications (COGNITIVE 2010), Lisbon, Portugal, 21–26 November 2010.
15. Xen Project 4.6 Series. Available online: https://www.xenproject.org/downloads/xen-archives/xen-46-series.html (accessed on 1 April 2017).
16. Barr, K.; Bungale, P.; Deasy, S.; Gyuris, V.; Hung, P.; Newell, C.; Tuch, H.; Zoppis, B. The VMware mobile virtualization platform: is that a hypervisor in your pocket? *ACM SIGOPS Oper. Syst. Rev.* **2010**, *44*, 124–135. [CrossRef]
17. Andrus, J.; Dall, C.; Hof, A.V.; Laadan, O.; Nie, J. Cells: A Virtual Mobile Smartphone Architecture. In Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles, Cascais, Portugal, 23–26 October 2011; pp. 173–187.
18. Kim, K.; Kim, C.; Jung, S.; Shin, H.; Kim, J. Inter-domain socket communications supporting high performance and full binary compatibility on Xen. In Proceedings of the Fourth ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments, Seattle, WA, USA, 5–7 March 2008; pp. 11–20.
19. Mafrur, R.; Nugraha, I.G.; Choi, D. Modeling and discovering human behavior from smartphone sensing life-log data for identification purpose. *Hum.-Cent. Comput. Inf. Sci.* **2015**, *5*, 31. [CrossRef]
20. Weerasinghe, D.; Rajarajan, M.; Rakocevic, V. Device Data Protection in Mobile Healthcare Applications. In Proceedings of the International Conference on Electronic Healthcare, London, UK, 8–9 September 2008; pp. 82–89.
21. Park, J. H.; Joo, J.W.; Lee, J.K. Security Considerations for a Connected Car. *J. Converg.* **2015**, *6*, 1–9.
22. Lim, K.-S.; Jeon, Y.-S.; Kim, J.-N.; Lee, D.-G. A Methodology for Live Forensic Acquisition in Secure Domain Based on Domain Separation Technology. In *Advanced Computer and Communication Engineering Technology: Proceedings of the ICOCOE 2015, Phuket, Thailand, 9–11 June 2015*; Springer: Cham, Switzerland, 2016; pp. 1113–1123.
23. Park, S.W.; Lim, J.D.; Kim, J.N. A secure storage system for sensitive data protection based on mobile virtualization. *Int. J. Distrib. Sens.* **2015**. [CrossRef]

24. Jung, Y.-W.; Sok, S.-W.; Santoso, G.Z.; Shin, J.-S.; Kim, H.-Y. Prototype of Light-weight Hypervisor for ARM Server Virtualization. In Proceedings of the International Conference on Embedded Systems and Applications (ESA), The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), Las Vegas, NA, USA, 27–30 July 2015.
25. Kim, K.H.; Koh, K.; Jeon, S.; Jung, S. Portable hypervisor design for commercial 64-bit Android devices supporting 32-bit compatible mode. In Proceedings of the Advances in Computer Science and Ubiquitous Computing, Phuket, Thailand, 19–21 December 2016; pp. 436–441.
26. Park, S.W.; Kim, J.N.; Lee, D.-G. SecureDom: Secure mobile-sensitive information protection with domain separation. *J. Supercomput.* **2016**. [CrossRef]