International Journal of Information Management 34 (2014) 329–335

Contents lists available at ScienceDirect

International Journal of Information Management

journal homepage: www.elsevier.com/locate/ijinfomgt

A framework of cloud-based virtual phones for secure intelligent information management

Jiun-Hung Ding[a], Roger Chien[b], Shih-Hao Hung[b,*], Yi-Lan Lin[a], Che-Yang Kuo[a], Ching-Hsien Hsu[c], Yeh-Ching Chung[a]

[a] Department of Computer Science, National Tsing Hua University, Taiwan
[b] Department of Computer Science and Engineering, National Taiwan University, Taiwan
[c] Department of Computer Science and Information Engineering, Chung Hua University, Taiwan

ARTICLE INFO

Article history:
Available online 30 December 2013

Keywords:
BYOD
Mobile cloud computing
Mobile security
Information management

ABSTRACT

As mobile networks and devices being rapidly innovated, many new Internet services and applications have been deployed. However, the current implementation faces security, management, and performance issues, which are critical to the use in business environments. Migrating sensitive information, management facilities, and intensive computation to security hardened virtualized environment in the cloud provides effective solutions. This paper proposes an innovative Internet service and business model to provide a secure and consolidated environment for enterprise mobile information management based on the infrastructure of cloud-based virtual phones (CVP). Our proposed solution enables the users to execute Android and web applications in the cloud and connect to other users of CVP with enhanced performance and protected privacy. The organization of CVP can be mixed with centralized control and distributed protocols, which emulates the behavior of human societies. This minimizes the need to handle sensitive data in mobile devices, eases the management of data, and reduces the overhead of mobile application deployment.

1. Introduction

Nowadays, mobile devices, such as *smartphones* and *tablets*, have become increasingly popular, and its shipping volume already exceeds the volume of personal computers (PC's). Mobile devices are being integrated into our personal lives, business activities, government services, and even military operations. *Enterprises* must carefully use such rapid-evolving mobile technologies in daily operations to meet high security and management requirements.

For information management in an enterprise environment, it is critical to evaluate the potential risks and issues as mobile technologies being integrated into its infrastructure. Valuable, sensitive and private contents can be leaked and cause great damages when a mobile device is compromised (Li & Clark, 2013). For instance, as unofficial mobile applications are downloaded, many of them may be *malwares* created by repackaging existing applications and injecting malicious code (Zhou & Jiang, 2012). A malware can steal the credentials of a mobile user and gain access to data and recourses of an enterprise via the user's mobile device. After analyzing more than 1.85 million mobile apps, Juniper Networks recently reported that the total amount of mobile malware increased by 614% between March 2012 and March 2013 to a total of 276,259 malicious mobile applications (Protalinski, 2013).

Facing the increased number of security threats for mobile devices, it is important to find proper solutions to strengthen today's mobile environments. In this paper, we hope to address the following issues for the information security and management in an enterprise IT infrastructure:

- *Operating environments*: There are diversified operating environments for mobile devices. While Android and iOS dominate mobile market today, new technology evolves so fast that there are multiple versions of operating environments being used by vendors, which has created a fragmentation problem which makes deployment of applications and management of mobile devices difficult for enterprises (Han et al., 2012).
- *Security and isolation*: While modern mobile operating environments use sandbox to isolate the execution of applications and provide a seemingly more secure execution environment, mobile devices are still subjective to many types of malware attacks (Zhou & Jiang, 2012). Unfortunately, due to resource limitations, such as CPU speed and battery time, mobile devices are not protected as well as PC's are, in terms of antivirus/anti-malware schemes, application management facilities, network traffic monitoring mechanisms, and virtualization technologies. Furthermore, a *rooted* Android device or a *jailbroken* iOS device

[*] Corresponding author. Tel.: +886 919695141.
E-mail address: hungsh@ntu.edu.tw (S.-H. Hung).

0268-4012/$ – see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.ijinfomgt.2013.11.006

allows applications to execute in the superuser mode and even gain the highest privilege to break the sandbox isolation protection (Li & Clark, 2013).

- *Sensory applications*: Different from PC's, each mobile device usually contains a rich set of sensors. There are sensory applications which may take advantage of the sensors to identify the user's location and position with a GPS and a gyroscope, record audio with a microphone, and connect to a payment system via the near-field communication (NFC) protocol. These features may not be aware by traditional enterprise management software.
- *Consumerization of IT:* It has become a trend that enterprise employees prefer to carry their own devices, use their own applications and connect to the corporate network with their own device, with or without the approval from the organizations. The term BYOD (Bring your own device) refers to such mobile workers who bring their own mobile devices into their worksplace. Embracing the consumeration of IT will not only save money but also improve employee productivity (Webopedia, 2013). However, this poses security threats to the organization as it introduces untrusted devices and unsecure network connections to the work environment.

To help enterprises solve these problems in terms of manageability and security, we propose a framework called *cloud-based virtual phone* (CVP) technology for mobile devices based on our previous works on virtualized execution environment for smartphones (Hung, Shih, Shieh, Lee, & Huang, 2012). The concept of CVP is inspired by the behavior of the *human societies* and the idea of *federalism*, which describes the progress of federation that divides sovereign into federal government and states (Bednar, Eskridge, & Ferejohn, 1999). The proposed framework enables critical business applications to be executed in a controlled virtualized environment on enterprise server farm, while the client-side software can be quickly deployed to almost any mobile devices to interact with the business applications. Unlike traditional *Virtual Desktop Infrastructure* (VDI) technology (Baratto, Potter, Su, & Nieh, 2004), our framework is designed to support local execution of non-critical mobile applications with data synchronization protocols. Overall, our framework contains an *HTML5 Web-based front end* (Kanaka, 2013) to provide different modes of operations, a *KVM-based virtual phone system* (KVM, 2013) to execute mobile applications efficiently, and a set of *security/management modules* to ensure the confidentiality of data and to mitigate the complexity of policy enforcement. The framework also provides a set of APIs is also provided for enterprise to develop their own applications that can be deeply integrated into this framework.

The rest of this paper is organized as the following. Section 2 further describes the weaknesses in the current solutions and the related works. Section 3 describes the models and the proposed framework. Section 4 presents the case studies and discusses experimental results. Finally, Section 5 concludes this paper and discusses future research directions.

2. Background and related works

Mobile technologies brought anytime, anywhere access to information resources and caused significant impacts to the IT organization (IBM, 2012). There are a variety of wireless mobile networks available today, such as WiFi, 3G/4G, Bluetooth, NFC, etc. for connecting a mobile device to the Internet Service Providers (ISP's) or surrounding devices. Unlike a PC which is connected to a fixed network router, it is possible for the user to send out messages via one of those wireless communication channels without being monitored by the enterprise.

BYOD has become a new trend for the enterprise. According to the reports from Unisys and IDC, there are about 40% of information workers who use their own mobile devices to access business applications. This trend has increased 10% compared with 2010 (Burt, 2011; IDC, 2011). While BYOD can bring several benefits to the enterprise, such as making the enterprise to be agile and more competitive (Tomson, 2012), BYOD also represent the security and management challenges to the IT management. It is obvious that security is not easy to control (Miller, Voas, & Hurlburt, 2012; Tomson, 2012). When employee lost their devices, it may cause enterprise's internal information to be stolen. When employee quit their job, the enterprise data stored in their own devices would be a threat if they sell enterprise data to enterprise's competitor (Assing & Calé, 2013; Miller et al., 2012).

IDC has reported that IT managers need to realize that even if they do not allow employees to use their own devices, they will find workarounds and BYOD will still seep into the enterprise (Sacchi, 2012). Therefore, IT departments must find the right balance between flexibility and potential security risks. The existing solutions, such as MDM and VDI (discussed below), to incorporate mobile devices into an enterprise environment still have some weaknesses.

MDM (mobile device management) is inherited from device management or endpoint management in PC era. A client software needs to be installed on client mobile device, and it regularly reads the status of the device, checks if any policies is violated and reports to central management system. It may also create the secure storage for specified applications. However, to deploy MDM to various mobile OSes and device configurations has a portable issue. In addition, MDM client software will detect threats through granting more privileges, but application sandboxing in mobile platform may limit the functions of MDM. Furthermore, if a MDM is exploited, mobile devices and the data they contain will be compromised (Rhee, Won, Jang, Chae, & Park, 2013). Compared with MDM, our CVP (Cloud-based virtual phone) will use the virtual machine technology to isolate business domain from mobile devices. Security threats are detected in a centralized virtualization environment with a better portability and fewer privilege issues.

VDI (Virtualization Desktop Infrastructure) allows the user to access to the user's desktop environment hosted by a remote enterprise server via a remote display protocol. This enables the IT department to control and manage the user's environment. Since VDI is designed to deliver server-hosted virtual desktops to a range of devices, it can be easier for IT to perform cross-platform management and security checks (Oracle, 2013). However, there are two problems for using VDI on mobile device: (1) the mobile device must be connected to the network all the time, and (2) VDI does not support sensory mobile applications. In comparison, our CVP offers both on-line and offline execution models as well as sensor-aware interfaces.

3. The proposed framework

In this section, we describe our proposed framework in details. Section 3.1 gives an overview on the framework. Section 3.2 describes the information management facilities in this framework. This section focuses on the concept and organization of the CVP framework.

3.1. Overview of the framework

Bednar et al. (1999) describes the progress of federation that divides sovereign into federal government and states as *federalism*. Three features in federalism are referred as follows. The first

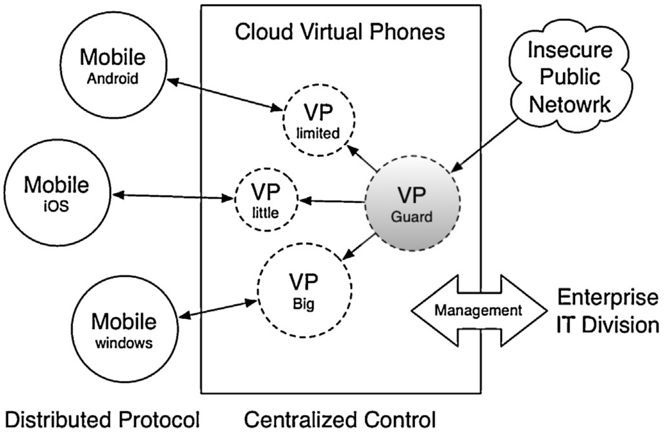

Fig. 1. The organization of CVP.

feature is *decentralized authority*, which means that the federation are divided into several independent states with sovereign. Decentralized authority not only makes federation more flexible and agile when the country have vast territories and population, but makes the regime of federation to be more stable (Rodden, 2004). The second feature mentions that *independence of sovereign* may give every state to have their own authority and different policies. It will helps each state develop independently and fit its requirements (Kelemen, 2007). The last feature is *supremacy of the constitution*. According to the federal constitution, every state has its own state constitution, law, and regulation. The state constitution, law of state, and regulation of state cannot disobey the federal constitution (Bednar et al., 1999).

Inspired by the idea of federalism, we proposed the CVP framework by associating the concept of *mobile cloud computing* (Dihn, Lee, Niyato, & Wang, 2013) with our previous work (Hung et al., 2012) to satisfy the need of computing and data storage for mobile devices via the cloud-based virtual phone service. Mobile cloud computing has similar features as decentralized authority, independence of sovereign and supremacy of the constitution.

As shown in Fig. 1, the virtual phones (VP's) can be mixed with centralized control and distributed protocols. This minimizes the need to handle sensitive data in mobile devices, eases the management of data, and reduces the overhead of mobile application deployment. The CVP framework also makes use of virtualization technology to create different kinds of virtual phone based on resource usage on cloud, such as *VP-big*, *VP-little*, *VP-limited* and *VP-guard* in Fig. 1. VP-guard is responsible for threat detections from insecure public network. VP-big and VP-little are represented as virtual phones with more computing resources and virtual phones with less computing resources respectively. VP-limited is considered as a high-secure virtual phone with more constraints to limit unexpected behaviors from client mobile devices. The CVP framework has the following advantages:

1. *Security and isolation*: Running mobile applications on a virtual phone is more secure than executing them on a physical phone because most of malicious activities will be detected immediately in a centralized controlled environment. The protection of private enterprise data is enhanced because critical data need not to be resided in the physical phone and exposed to the world during data transmission. If one of the virtual phones is compromised, only the files on the virtual phone are at risk, and the others are isolated from the risks. VP-guard will be called to clean the security threats and restore the compromised virtual phone to a known good state.
2. *Intelligent control and management*: The CVP framework will give a way of control and management for the IT department. There is an interface to configure and manipulate the virtual phones in the framework. Illegal accesses can be blocked out automatically by configuring rules and policies.
3. *Flexibility*: Virtualization is a popular topic of computer science. It supplies a method to deploy devices dynamically. The virtualization technology can be easily utilized in enterprise environment. With these technology, it becomes efficiently to manage and control the mobile devices of employee. When some employees promote, resign, or work in another department, the data in their virtual phone will be deleted or updated. It benefits the management of human resource in the enterprise.
4. *High performance*: Some applications for business may consume much computing resource. Despite some of these applications are developed on mobile platform, the performance of these applications cannot satisfy the requirement for business because there is limited computing resource inside the mobile device. Via the CVP framework, the user may utilize a VP-big to acquire higher computational power and/or bigger storage capacity in order to execute these heavy weight business applications.
5. *Cost effective*: The CVP system can be easily constructed on existing equipment of enterprise. It only requires the server supports virtualization technology, and then many virtual phones can be created on the existing server. Compared to providing feature phones or laptop computers to employees, the benefit of mobile cloud computing and the relative low cost are given by the CVP system.
6. *Ease of use*: The CVP system provides a cross-platform interface to everyone, so that the user may access the virtual phone through the different mobile platforms such as Android, iOS, or Firefox OS. The operation of a virtual phone is nearly identical to a physical Android phone with possibly some customization, which should be easy to use since many people are already familiar with the user interface of smartphones.

3.2. Information management

While multiple layers of *certification mechanisms* are required for the users to access enterprise data, the management of information is made easy for the IT department and the owner of the data. The CVP system supplies a comprehensive API for the IT department to control the CVP, with many options which can be used to implement the policies in the enterprise. It is also quite easy for the IT department to configure the options when the policies change and during some special situations. For information management, we provide the following three modules:

1. The *policy check module* examines if the current status and configuration can meet the defined security policies. In our framework, since majority of the operations residents in server side, it is far easier to have powerful check and enforcement mechanisms implemented in practice than other known MDM solutions, since the security policies are not constrained by the capability of mobile OS or the computational power on mobile devices.
2. The *data management module* is for the gatekeeper to decide the users, the applications, and the modes which may have access to any given data file. It also screens the data being sent out to Internet by integrating with DLP functions.
3. Finally, a set of API composes the last piece of proposed framework which enables the enterprise to develop applications that have deep integration with security check and policy enforcement module. It also enables the framework to determine if a

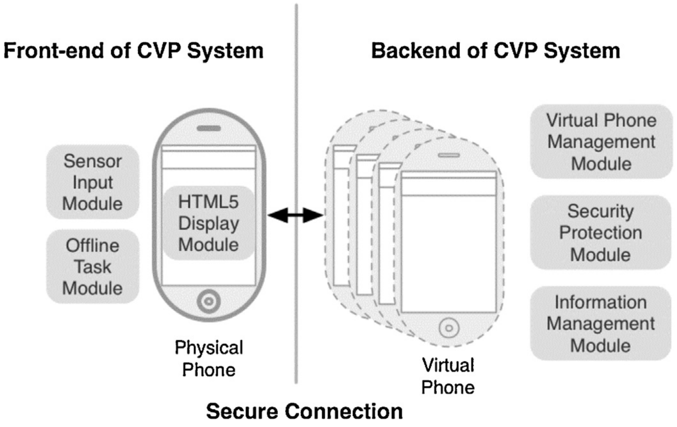

Fig. 2. A CVP system is composed of two parts, front-end and backend.

mode transition is feasible at certain points during the application execution.

4. System architecture

In this section, we discuss the software components in the CVP framework for constructing a CVP system for an enterprise, as shown in Fig. 2. The entire CVP system architecture is divided into two major parts: *front-end* and *back-end*. The front-end includes *HTML5 Display Module, Sensor Input Module and Offline Task Module.* Employees can operate their own CVP via any HTML5-compatible browser of mobile devices. In the back-end, three modules, such as *Virtual Phone Management Module*, *Security Protection Module* and *Information Management Module*, are cooperated to build up a virtualized secure environment for application and information management. The following subsections describe the components in this framework in details.

4.1. Front-end modules

The front end is a web-based application GUI based on HTML5 (W3C, 2010) technology which supports two modes: *remote mode* or *native web application mode*. The adoption of HTML5 makes this framework highly portable as it is capable of running within any HTML5-capable browser which exists on most smart mobile devices, which also eliminates the portability issue of operating environments, since HTML5 is becoming standard in the near future.

4.1.1. HTML5 display module

The CVP system does not require employees to setup any "*agent*" programs on their physical phones. Using HTML5, employees can operate their virtual phones via browser on their own devices, so it would not matter which operating environment is installed on the physical phones. The user simply needs to open the company's website and login. We use *noVNC* (Kanaka, 2013) to relay visual frames and control messages, for collecting user input events like key stokes and touch screen actions and sending the events to the server, between the physical phone and virtual phone. In the *remote mode*, visual frame is rendered and transmitted to noVNC by virtual phone on server side. Quality of rendered images can be adapted to configuration of mobile devices and network speed. All connections are protected by secure transmission protocol such as SSL/TLS.

4.1.2. Sensor Input Module

The *Sensor Input Module* is designed for collecting sensor data from a physical phone and then forwarding the data to a virtual phone. For example, if any application running on a virtual phone needs to acquire the current GPS location of a physical phone, the *Sensor Input Module* will seamlessly deliver the required information to virtual phone. Our implementation takes advantage of HTML5's capability in accessing device hardware functions through the hardware abstraction layer. All of sensor data are kept in virtual phones.

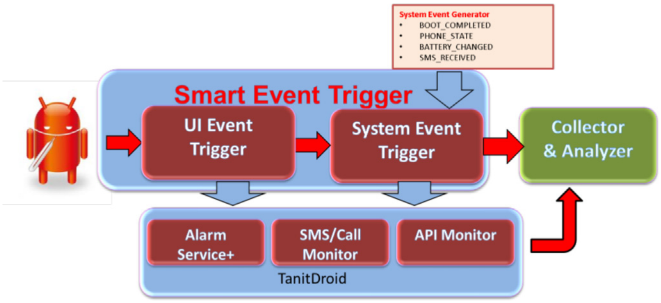

Fig. 3. The APE testing flow.

4.1.3. Offline Task Module

In certain situations, the user may need to work offline. For example, network connection is lost, and therefore the remote mode is not applicable. The CVP system provides *native web application mode* to tackle this issue. Initially, the user may start a virtual phone to run a web application in remote mode and later switch to the native web application mode if the backend security policy is allowed to grant such a request. By utilizing features of HTML5 offline application and persistent local storage, the user can continue to work offline with limited data permitted by backend. Later, when the network connection is resumed, the *Offline Task Module* can update local changes to server and switch to remote mode again. In general, only low-sensitive data or data collected on the physical phone are allowed to run in the native web application mode.

4.2. Back-end modules

4.2.1. Virtual Phone Management Module

The CVP system is facilitated by the *Virtual Phone Management Module* which creates and manages virtual machines that serve as virtual phones, which can be centrally controlled by private cloud infrastructure. The specifications of virtual phones can be configured by different usage scenario of business applications, such as computing intensive applications and high-secure applications Hence, in a CVP system, three kinds of virtual phones are implemented: VP-big, VP-little and VP-limited.

4.2.2. Security Protection Module

The *Security Protection Module* performs security threat detections to prevent employees' CVP's from compromising. The mechanism of security threat detections is based on a dynamic analysis approach which observes informal behaviors of unofficial mobile applications in virtual phones for security guard (VP-guard). To support automate dynamic analysis, a smart mechanism to generate proper stimulus to an application is the key issue. Android *Monkey* (Google, 2013) tool is capable of generating screen input and system event in a random fashion for Android's development and testing. However, the Monkey tool ignores the information and underlying structure embedded in GUI, which makes it an inefficient tool for malware testing because most of generated events are invalid. In addition, Monkey tool only supports a limited and simple set of trigger-able system events.

To improve the automation dynamic analysis of malwares, VP-guard uses APE+, as shown in Fig. 3, to stimulate more meaningful inputs to expose the malicious behavior in a malware in short time (Chang, 2013). APE+ can understand GUI view structure of one mobile application and can provide a wider support for system event generation, such as *battery status*, *connectivity status change*, *SMS message receiving*, which are often used by malwares to wake up themselves by registering themselves as listeners for such events.

For better coverage, VP-guard enable many security probes in virtual phones to collect detailed traces for analysis, such as *TaintDroid*, which is capable of a real-time privacy monitoring on smartphones (Enck et al., 2010).

4.2.3. Information Management Module

Another special group of virtual phones on the server are called *shadow phones*. When a user tries to access the company's data, the user's information on their virtual phone will copy into shadow phone, as if it is an *agent* of enterprise's data management department. It is the only gate via which employees can acquire data. Since the shadow phone monitors the commands to access company's data from the user, illegal data access requests will be immediately blocked and data will not be leaked.

The information copied into the shadow phone includes employee's identity (i.e. position, department, contact, message log and so on in their virtual phone). The shadow phone serves as a gatekeeper of enterprise and monitors the user's data access pattern to see if the user is well-behaved. First, normally, a user is not supposed to access the data belong to another group. Second, the user should not suddenly acquire a large amount of sensitive information.

The implementation of shadow phones is similar to other virtual phones in the CVP system but provides an interface for enterprise's management department. The manager can choose to block out employees' virtual phone when a shadow phone detects unauthorized operations or data access patterns. Or, the manager can let the shadow phone automatically send a warning message to the employee. With the shadow phone, it is easy to track the data access records and spot malicious employee.

5. Evaluation

In this section, the CVP framework is evaluated to reveal its performance, memory consumption and security strength. At first, *Vellamo Benchmark* (Qualcomm Innovation Center, 2013) is used to measure the performance of a virtual phone. Then, we measure the memory resources consumed by a single virtual phone. Finally, we evaluate the capability of the APE security testing component to detect malwares.

The specifications of the server and the virtual phone in the experiments are listed in Table 1. The server is equipped with 2 Intel Xeon L5640, 24 GB memory and 2 TB storage. Each CPU has 6 cores and 12 hyper threads, therefore the server has totally up to 12 cores and 24 hyper threads. Running on top of the Ubuntu 13.04 Linux operating system, the virtual phone is configured to run the Android 4.3 mobile operating system using a single core CPU, 2 GB RAM, and 2 GB data storage.

Table 1
Specification of server and virtual phone.

Item	Server	Virtual phone
OS	Ubuntu 13.04	Android 4.3
CPU	2 × Intel Xeon L5640	Simulated × 86 CPU
Cores	12 cores	1 core
Thread	24 Hyper Threads	1 thread
Memory	24 GB	2 GB
Disk	2 TB	2 GB

Table 2
The Vellamo mobile benchmark.

HTML5 chapter bencharks	Metal chapter benchmarks
See the Sun Canvas	Dhrystone
Pixel Blender	LINPACK
Canvas Crossfader	Branch-K
Aquarium Canvas	Stream 5.9
Sun Spider, v0.9.1	RamJam
V8 Benchmark Suite	Storage
Surf Wax Binder	
DOM Node Surfer	
Reflo	
Image Scroller	
Ocean Scroller	
WebGL Jellyfish	
Inline Video	
Load And Reload	

Table 3
The performance comparison of the virtual phone and the physical phones with the Vellamo mobile benchmark.

	HTML5 chapter	METAL chapter
Virtual Phone	3456	2358
Samsung Galaxy Note3	2739	1282
LG G2	2844	1202
Sony Xperia Z Ultra	2919	1119
Samsung Galaxy S4	1973	1032
HTC One	2356	763
Xiaomi MI-2s	1897	711

5.1. Virtual phone performance

The Vellamo mobile benchmark is widely used for mobile performance test, which consists of two sets of benchmark programs. One is *HTML5 Chapter*, which is use to evaluate the performance of mobile web browsers in terms of graphic rendering, JavaScript execution, pixel blending, and network operations. The other is *Metal Chapter* which measures the performance of mobile processors, such as floating point operations, memory read/write, and memory branching. The benchmark programs are listed in Table 2.

As the experimental results shown in Table 3, the performance of a virtual phone is substantially higher than today's flag-ship smartphones in the market, which indicate the advantage of the virtual phone in heavy weight business applications. With modern virtualization technologies, virtual phones can perform very well for both HTML5 applications and native mobile applications. Although the performance of a virtual phone may be dynamically changed due to the computing power of server and workloads running in the server, it can be solved by adopting a load balancing algorithm with dynamic cost-performance decision to deploy a proper server to effectively accelerate the mobile application while saving power consumption of physical phone (Hung, Liang, Tu, & Chang, 2013).

5.2. Memory consumption of virtual phone

In this subsection, we measure the memory resources utilized by adding a single virtual phone. In the beginning of this experiment, our server is initialized without any virtual phone. The initial memory consumption is about 600 MB. As each of the ten virtual phones is started one by one, the memory consumption grows linearly with the number of virtual phones, as shown in Fig. 4. The average memory consumption of a virtual phone is about 900 MB, while the memory configuration of a virtual phone is configured to 2 GB. Note that when the number of virtual phones exceeds 28, the server runs out of empty memory and starts to swap old memory data into the disk, since the server is equipped with 24 GB. The swapping operations cause dramatic performance degradation.

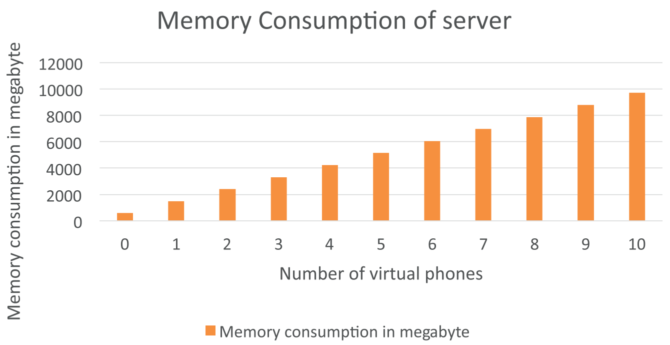

Fig. 4. Memory consumption of server.

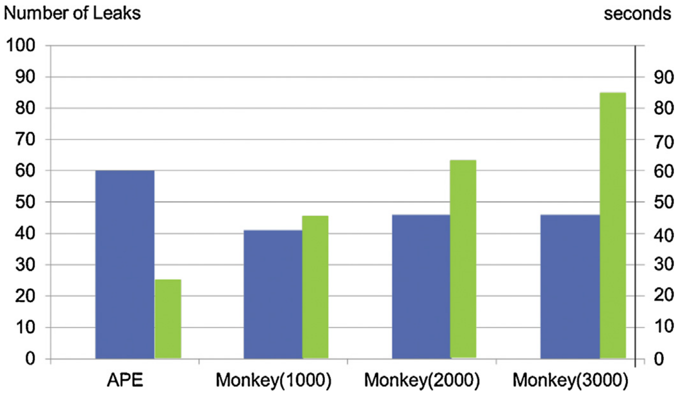

Fig. 5. Performance comparison between APE and Monkey for 3rd-party Android applications.

5.3. Security strength test

First, let us compare the performance of APE and Monkey with 100 popular mobile apps downloaded from third-party market. As shown in Fig. 5, APE was able to trigger more number of leaks than Monkey even when Monkey was configured to generate 3000 events. APE is also more efficient than Monkey as it far less time to trigger more leaks. Note that this test catches any type of information leaks even if the application has asked for user's permission. The results show that 60% of the downloaded applications collect user's information.

Similar results are shown in Fig. 6, where 100 *malicious* Android applications were used to conduct the test. The difference was even more significant since APE quickly detected 86 malwares within roughly 20 s. We also conduct another test for comparing APE & APE+ with 300 malicious apps. The results are shown in Table 4, which suggest that APE+ delivered more coverage in triggering the leaks but also spent more time to handle the broadcast events. In Android, a broadcast event is sent to every receivers registered to receive the event, which takes time to complete.

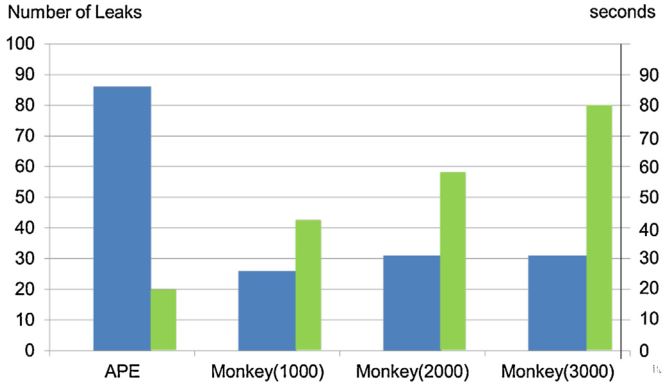

Fig. 6. Performance comparison between APE and Monkey for 100 Android malware applications.

Table 4
Comparison with APE and APE+ with 300 malware applications.

	APE	APE+
Number of malware detected (%)	198 (66%)	226 (75.33%)
Average time per malware (s)	25.7	58.3

6. Conclusion

Embracing mobile technologies and consumerization of IT can make enterprises more agile, save costs and improve employee's productivities. However, the current mobile environments come with serious secure threats and management issues which require innovative solutions. Inspired by the behavior of human societies and federalism, the proposed CVP system eliminates the fragment problem of mobile systems, simplifies the development and deployment of business applications, and constitutes a secure mobile environment with the virtualization technique, malware detection and informal behavior monitoring. In the near future, we will research further to refine the software components and conduct more experiments based on different forms of federalism.

References

Assing, D., & Calé, S. (2013). *Mobile access safety: Beyond BYOD*. Wiley-ISTE. 61 Available at: http://as.wiley.com/WileyCDA/WileyTitle/productCd-1848214359.html.

Baratto, R. A. B., Potter, S., Su, G., & Nieh, J. (2004). MobiDesk: Mobile virtual desktop computing. In *Proceedings of the 10th annual international conference on mobile computing and networking* (pp. 1–15).

Bednar, J., Eskridge, W. N., & Ferejohn, F. (1999). *A Political Theory of Federalism*. Retrieved from http://www.personal.umich.edu/~jbednar/Pubs/befwbib.pdf

Burt, J. (2011). *BYOD Trend Pressures Corporate Networks*. eWeek. Available at: http://winfwiki.wi-fom.de/images/2/2e/65469365.pdf

Chang, S. J. (2013). *APE: A Smart Automatic Testing Environment for Android Malware* Master Thesis. National Taiwan University. Available at: http://www.airitilibrary.com/Publication/alDetailedMesh1?DocID=U0001-1908201316171100.

Dihn, H. D., Lee, C., Niyato, D., & Wang, W. (2013). A survey of mobile cloud computing: Architecture, applications, and approaches. *Wireless Communications and Mobile Computing*, *13*(18), 1587–1611.

Enck, W., Gilbert, P., Chun, B.-G., Cox, L. P., Jung, J., McDaniel, P., et al. (2010). Taintdroid: An information-flow tracking system for realtime privacy monitoring on smartphone. In *Proceedings of the 9th USENIX conference on operating systems design and implementation, pp. 1–6*. Available at: http://dl.acm.org/citation.cfm?id=1924943.1924971

Google. (2013). *UI/Application Exerciser Monkey*. Available at: http://developer.android.com/tools/help/monkey.html

Han, D., Zhang, C., Fan, X., Hindle, A., Wong, K., & Stroulia, E. (2012). Understanding android fragmentation with topic analysis of vendor-specific bugs. In *Proceedings of the 19th working conference on reverse engineering* (pp. 83–92).

Hung, S.-H., Shih, C.-S., Shieh, J.-P., Lee, C.-P., & Huang, Y.-H. (2012, January). Executing mobile applications on the cloud: Framework and issues. *In Computers & Mathematics with Applications*, *63*(2), 573–587.

Hung, S.-H., Liang, F.-T., Tu, C.-H., & Chang, N. (2013). Performance and power estimation for mobile-cloud applications on virtualized platforms. In *Proceedings of the seventh international conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS-2013)* Taichung, Taiwan, July 3–5.

IBM. (2012). The flexible workplace: Unlocking value in the "bring your own device" era. In *IBM global technology services thought leadership white paper*. Available at: ftp://ftp.software.ibm.com/common/ssi/ecm/en/enw03010usen/ENW03010USEN.PDF

IDC. (2011). *2011 Consumerization of IT Study: Closing the Consumerization Gap*. Available at: http://idc.cycloneinteractive.net/unisys-iview-2011/en/

Kanaka. (2013). *VNC client using HTML5 with encryption support*. Available at: http://kanaka.github.io/noVNC/

KVM. (2013). *Kernel Based Virtual Machine*. Available at: http://www.linux-kvm.org/page/Main_Page

Li, Q., & Clark, G. (2013). Mobile security: A look ahead. *IEEE Security & Privacy*, *11*(1), 78–81.

Miller, K. W., Voas, J., & Hurlburt, G. F. (2012). BYOD: Security and privacy considerations. In *IT Professional*. Available at: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=06320585

Oracle. (2013). *Oracle Virtual Desktop Infrastructure*. Available at: http://www.oracle.com/us/technologies/virtualization/virtual-desktop-infrastructure/overview/index.html

Protalinski, E. (2013). *Juniper: Mobile malware is 'an increasingly profit-driven business' as 92% of all known threats target Android*. Available at: http://thenextweb.com/insider/2013/06/26/juniper-mobile-malware-is-an-increasingly-profit-driven-business-as-92-of-all-known-threats-target-android

Qualcomm Innovation Center, Inc. (2013). *Vellamo Mobile Benchmark*. Available at: http://www.quicinc.com/vellamo/

Rhee, K., Won, D., Jang, S.-W., Chae, S., & Park, S. (2013). Threat modeling of a mobile device management system for secure smart work. *Electronic Commerce Research*, *13*(3), 243–256.

Kelemen, R. D. (2007). Built to last? The durability of EU federalism. *Making history: European integration and institutional change at fifty*, 51–66.

Rodden, J. (2004). Comparative federalism and decentralization, on meaning and measurement. *Comparative Politics*, *36*(4), 481–500.

Sacchi, J. (2012). *Addressing Security Issues and Concerns of Bring-Your-Own-Device Initiatives*. Available at: http://www.idc.com/getdoc.jsp?containerId=AE53U

Tomson, G. (2012). BYOD: Enabling the chaos. *Network Security*, *2*, 5–8. Available at: http://www.sciencedirect.com/science/article/pii/S1353485812700132

W3C. (2010). *HTML5 Reference: The Syntax, Vocabulary and APIs of HTML5*. Available at: The Syntax, Vocabulary and APIs of HTML5.

Webopedia. (2013). *Consumerization of IT*. Available at: http://www.webopedia.com/TERM/C/consumerization_of_it.html

Zhou, Y., & Jiang, X. (2012). Dissecting Android malware: Characterization and evolution. In *IEEE symposium on security and privacy (SP)* (pp. 95–109).

Jiun-Hung Ding is currently a PhD student in the Department of Computer Science at National Tsing Hua University. His research interests include mobile virtualization, cloud computing and heterogeneous system architecture. He received his M.S. degree in the Department of Institute of Information System and Application at National Tsing Hua University in 2006 and graduated from National Chiao Tung University with a BS degree in industrial engineering and management in 2004.

Roger Chien is currently a PhD student in the Department of Computer Science and Information Engineering at National Taiwan University. His research interests include high speed networking, mobile system security and network security. He has been worked in the network security industry for years, at Broadweb Corp. (2000–2003) and at Lionic Corp. (2003–2013). He received his M.S. degree in Computer Science and Information Engineering at National Tsing-Hwa University in 2002 and graduated from National Taiwan Normal University with a BS degree in Information and Computer Education in 1997.

Shih-Hao Hung is currently an associate professor in the Department of Computer Science and Information Engineering at National Taiwan University. His research interests include mobile-cloud computing, parallel processing, computer system design, and information security. He worked for Sun Microsystem Inc. (2000–2005) after completing his post doctoral work (1998–2000), Ph.D. training (1994–1998) and M.S. program (1992–1994) at the University of Michigan, Ann Arbor. He graduated from National Taiwan University with a BS degree in electrical engineering in 1989.

Yi-Lan Lin is currently a graduate student in the Department of Computer Science at National Tsing Hua University. His research interests include parallel processing, GPU computing, and heterogeneous architecture. He graduated from National Chung Cheng University with a BS degree in computer science in 2013.

Che-Yang Kuo is currently a graduate student in the Department of Institute of Information System and Application at National Tsing Hua University. His research interests include parallel programming, android security and information security. He graduated from National Chung Cheng University with a BS degree in Information Management in 2013.

Ching-Hsien Hsu is a professor in department of computer science and information engineering at Chung Hua University, Taiwan. His research includes high performance computing, cloud computing, parallel and distributed systems, ubiquitous/pervasive computing and intelligence. Dr. Hsu is the editor-in-chief of international journal of Grid and High Performance Computing, and international journal of Big Data Intelligence; and serving as editorial board for many international journals. He has been acting as an author/co-author or an editor/co-editor of 10 books from Springer, IGI Global, World Scientific and McGraw-Hill. He has also edited a number of special issues at top journals, such as IEEE Transactions on Services Computing, IEEE Transactions on Cloud Computing, Future Generation Computer Systems, Journal of Supercomputing, Concurrency and Computation: Practice and Experience, The Knowledge Engineering Review, Internet Research, Information System Frontiers, etc. He was awarded 5 times annual outstanding research award through 2005 to 2012 and a distinguished award in 2008 for excellence in research from Chung Hua University. He has been serving as executive committee of Taiwan Association of Cloud Computing (TACC) from 2008–2012; executive committee of the IEEE Technical Committee of Scalable Computing (2008–2012). He is IEEE senior member.

Yeh-Ching Chung received a B.S. degree in Information Engineering from Chung Yuan Christian University in 1983, and the M.S. and Ph.D. degrees in Computer and Information Science from Syracuse University in 1988 and 1992, respectively. He joined the Department of Information Engineering at Feng Chia University as an associate professor in 1992 and became a full professor in 1999. From 1998 to 2001, he was the chairman of the department. In 2002, he joined the Department of Computer Science at National Tsing Hua University as a full professor. His research interests include parallel and distributed processing, cloud computing, and embedded systems. He is a senior member of the IEEE computer society.