# Security Architecture for Federated Mobile Cloud Computing

Shouhuai Xu and E. Paul Ratazzi and Wenliang Du

**Abstract** Mobile cloud computing systems are getting increasingly popular because they can facilitate many new applications, such as opportunistic social computing by smartphone users who happen to be at a scene of importance (e.g., disaster rescue), while possibly uploading compute-heavy tasks to the resource-rich clouds. Federated mobile cloud computing further allows to coordinate and optimize the services to mobile users of different clouds. Accompanying the great deal of opportunities it brings up, federated mobile cloud computing imposes a diverse set of new challenges, especially from a security perspective because the defender needs to cope with a large spectrum of attacks. Example security questions are: How should we better deal with the new dimension of threats that are caused by that smartphones run a huge population of *untrusted* third-party applications (apps)? How should we monitor the mobile clouds for security purposes? How should we deal with the targeted attackers that attempt to launch attacks against the various credentials used for authentication purposes (e.g., banking)? How should we enhance the privacy of users when a malware breaks into their smartphone (e.g., records of location information?) How should the federated mobile clouds share security information and possibly coordinate their defense activities? In this chapter, we explore the threat model against, and security requirements of, federated mobile clouds computing. We then propose and investigate a comprehensive security architecture, which can seamlessly integrate a set of novel security mechanisms that are tailored to satisfy the security needs of federated mobile cloud computing.

Shouhuai Xu
Department of Computer Science, University of Texas at San Antonio, One UTSA Circle, San Antonio, TX 78249, e-mail: shxu@cs.utsa.edu

E. Paul Ratazzi
Information Directorate, Air Force Research Laboratory, Rome, NY 13441, e-mail: edward.ratazzi@us.af.mil

Wenliang Du
Department of Electrical Engineering and Computer Science, Syracuse University, Syracuse, NY 13244, e-mail: wedu@syr.edu

**Keywords**: Cloud computing, mobile cloud, mobile computing, federated cloud, cloud security, smartphones, security architecture

## 1 Introduction

Cloud computing is getting widely deployed and is changing the landscape how Information Technology (IT) will serve the needs of government, enterprises, society and home users. The most recent development in cloud computing is characterized by the incorporation of mobile devices, especially smartphones that are embedded with a diverse set of sensors.[1] This naturally leads to the concept of *mobile cloud computing*, which is getting increasingly popular because smartphones are becoming the most popular user-end "computers" — both home users and enterprise employees would use smartphones for their information processing needs. For example, home users outsource their data to the cloud and they can have access to their data from anywhere in the world at any point in time using any smartphones, which would not be possible without the support of mobile communication and cloud computing technologies. As another example, enterprises outsource their IT systems as well as their data to the cloud so that their employees can do their job from any desktop/laptop/smartphones, while substantially reducing the (maintenance) cost of their IT systems and enjoying elastic compute as well as storage resources at low price. Improvement in performance, and possibly security, could be gained by *federated mobile cloud computing*, where multiple clouds may coordinate their activities for better serving their users and for enhancing security.

On the other hand, it is also natural to treat mobile cloud computing as evolved from the concept of *opportunistic computing* [1], where mobile devices or smartphones with various kinds of sensors can collect and share different kinds of data that can be further processed for various applications. The concept of opportunistic computing can be seen as evolved from the concept of *opportunistic* networks, for which a good example is the so-called mobile ad hoc networks (MANETs) where nodes cooperatively conduct some tasks [1]. A concrete example of this type of mobile cloud computing is *mobile phone sensing*, which exploits the various kinds of sensors embedded into the smartphones (e.g., camera, digital compass, GPS, and accelerometer) for various real-life applications, such as healthcare, transportation and environmental monitoring. We refer to [2] for a nice survey on mobile phone sensing algorithms, applications and systems. As promising as these applications, smartphones impose a significant security challenges exactly because of its strength — the rich set of third-party applications. Unlike traditional MANET-like scenarios where the number of applications would be reasonably small and may actually be developed by trusted parties/vendors, the huge number of third-party applications for smartphones are often developed by *untrusted* (or even malicious) developers. Indeed, our understanding of smartphones security is at a very early stage, and

[1] In the rest of the paper, the term "mobile devices" and the term "smartphones" are used interchangeably. Cloud and cloud data centers are often used interchangeably as well.

smartphones pose a new vector of threats against (federated) mobile cloud computing when compared with its predecessors.

After years of study by academic researchers and practice by the industry, there has been a relatively established understanding about the model, system, security architecture, and security mechanisms of the cloud-end systems. However, our understanding of secure mobile cloud computing, let alone secure federated mobile cloud computing, is at its infant stage. This motivates our study to systematically examine the threats against, and security needs of, federated mobile cloud computing.

**Our Contributions.**

In this Chapter, we systematically explore the threats against federated mobile cloud computing systems, with an emphasis on the threats that are caused by the unique features of federated mobile cloud computing. We also systematically examine the challenges encountered in designing security architecture for federated mobile cloud computing systems. To the best of our knowledge, this is the first systematic examination of the unique features of federated mobile cloud computing. In particular, we are the first to put forth that mobile cloud computing can aim to achieve the "$1 + 1 > 2$" effect in terms of functions (i.e., mobile cloud computing can achieve functions that are infeasible or even impossible to achieve by mobile computing or cloud computing alone). Despite that mobile cloud computing has an enlarged vulnerability surface than the underlying mobile computing and cloud computing, respectively, it is important to achieve the "$1 + 1 > 2$" effect in terms of defense, meaning the exploitation of cloud computing to help secure mobile computing and *vice versa*. For example, from the point of view of physical security, mobile devices and clouds are quite different. Nevertheless, it is possible to mitigate, if not minimize, the negative impact of mobile devices' physical insecurity, while leveraging the much stronger physical security of a cloud data center. Another point that is highlighted by our thorough examination is: It is often assumed that public clouds are not trusted, but private clouds are trusted. This perception can be misleading because one may think that private clouds are secure as well. In contrast, private clouds can be compromised and a compromised cloud may return misleading results back to the users. This is important especially for mission-critical applications, where the decision-making process needs to have a high trustworthiness in its input data. The false sense of security as caused by the perception "private clouds are more secure than public clouds" can be dangerous.

Based on a set of design principles that we believe to be the most relevant, we present a systematic security architecture for federated mobile cloud computing, where the cloud-end can be public clouds, private clouds, hybrid clouds, and social clouds [3]. To the best of our knowledge, this is the first security architecture for federated mobile cloud computing. The proposed security architecture supports a three-tiered system structure of federated mobile cloud computing: the mobile devices, the cloudlets, and the clouds (i.e., the data and/or compute centers). The

proposed security architecture can be implemented as a middleware that can be deployed at the mobile devices, the cloudlets, and the clouds. The proposed security architecture can accommodate a diverse set of security services. We not only discuss the security services offered by the security architecture components, but also explore approaches to fulfilling the security services. In particular, we propose the novel concept of *situational authentication*, which is different from the existing authentication methods that are based on "what you know" (e.g., a password or cryptographic private key), "what you have" (e.g., a hardware token), "who you are" (e.g., fingerprints and biometrics). Instead, situational authentication is based on "whom you are with" (e.g., the people surrounding you), "where you are" (e.g., your location or elements of your environment), and/or "what time is it" (e.g., dependent on particular events or other temporal factors), and can provide another layer of assurance when a mobile device (including the associated password or the authentication method it uses) is under the control of the attacker.

Having a good and competent security architecture is important for both commercial and military uses. The report "Cyber Security and Reliability in a Digital Cloud" published by the U.S. DoD's Defense Science Board in January 2013 [4] discussed security for data centers, but did not investigate security architecture and did not consider mobile cloud computing. Specifically, this reports focused on identifying applications of cloud computing for DoD's mission areas; enhancing DoD's cloud computing implementation; improving cloud resiliency (especially for deployed forces). Since, for example, cloud computing offers DoD agile compute resources to support complex missions and could allow packet-level or large-scale log data analysis for detecting malicious malware, it was recommended that DoD should establish security mandates for its cloud computing systems [4]. For the emerging federated mobile cloud computing, which could be the dominating computing paradigm for DoD, there is no documented security architecture in the public domain. The present chapter fills the void.

The rest of the chapter is organized as follows. Section 2 explores the model of, and threats against, federated mobile cloud computing. Section 3 presents and elaborate our security architecture for federated mobile cloud computing. Section 4 briefly reviews related prior work. Section 5 concludes the present chapter.

## 2 Federated Mobile Cloud Computing

The National Institute of Standards and Technology (NIST) defines *cloud computing* as "*a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.*" [3] However, the concept of mobile cloud computing is relatively new, and there is no well-accepted definition of it (see, for example, [5]). Nevertheless, in this chapter we explore some representative application scenarios of mobile cloud computing, and further investigate how

federation of mobile cloud computing can enhance the function, performance, and security of mobile cloud computing.

## *2.1 The Concept of Federated Mobile Cloud Computing*

**Three scenarios of mobile cloud computing.**

We start with three possible scenarios of mobile cloud computing.

- Scenario I: Using mobile devices in cloud computing. In this scenario, users mainly use mobile devices to access cloud data and resources. This is perhaps the most popular scenario in business computing. In this case, it is important to offload computing tasks to clouds so as to alleviate the resource constraints at the mobile devices. ThinkAir [6] and MAUI [7] are examples of such systems that allow fine-grained offloading of mobile programs (or program components) to the cloud while achieving the best of the following two approaches: human-supported program partition for offloading and coarse-grained process/VM migration. This allows, for example, MAUI to achieve fine-grained program offloading to maximize battery lifetime while minimizing the burden on programmers. CloneCloud [8] can achieve even finer-grained offloading, by migrating threads as well as its execution state at any time and allowing to return from the migrated state to the original process. Finally, the idea of cloudlet was proposed in [9], which allows a mobile device to instantiate customized VM on a nearby cloudlet via wireless LAN (rather than 3G/4G/LTE communications). A cloudlet is a trusted powerful computer that has good Internet connections. However, security aspect of offloading has not been duly investigated, except some specific designs that are tailored to some specific offloading schemes [10, 11].
- Scenario II: Mobile cloud computing based on mobile devices. As mobile devices or smartphones are getting increasingly powerful, it is possible to formulate a cloud computing system/infrastructure on demand and based on the mobile devices themselves [12]. This is relevant for facilitating compute tasks in the lack of cloud data centers (e.g., emergency rescue). This can be seen as a natural evolution of MANET-based computing as well.
- Scenario III: Hybrid. This is a hybrid of the above two scenarios. Specifically, this is the case where mobile devices are used for accessing cloud data and resources, and mobile devices are also used to formulate cloud computing substrate for the needs of some special applications. This is perhaps especially relevant for military applications and rescue applications, where the network connection to the cloud data centers can be disrupted by the adversary or by nature. Under such circumstances, data may be downloaded from the cloud data center to the smartphones that are physically located close to each other. When the communication to the cloud data center is disrupted, the smartphones can use distributed computing to accomplish some tasks. Under other circumstances, the smartphones may choose to conduct some compute tasks, rather than uploading the tasks to

the cloud data centers, because the smartphone-to-cloud communication based on 3G/4G/LTE can actually consume much more battery power than what is consumed when the smartphones conduct the tasks by themselves.

**Mobile cloud computing.**

By *mobile computing*, we mean that a set of users who conduct some joint computational and communication tasks based on their mobile devices. As mentioned above, we will primarily focus on smartphones because they are becoming the dominating mobile devices. However, as we will elaborate later, there is a big difference between traditional mobile computing and smartphone-based mobile computing. The difference is caused by that smartphones run a large number of applications (apps) that are often developed by *untrusted* (or even malicious) third parties. In contrast, traditional mobile computing, especially military-based MANET applications, is often based on special devices that may only run a small number of applications that are developed by *trusted* parties or vendors.

While it is intuitive that

$$mobile\ cloud\ computing = mobile\ computing + cloud\ computing,$$

it is important to recognize that mobile cloud computing is not simply the integration of mobile computing and cloud computing. Instead, mobile cloud computing aims to achieve the best of the two worlds, while overcoming their own disadvantages (i.e., the "$1+1>2$" effect). In what follows, we highlight the advantages (indicated by "**+**" below) and disadvantages (indicated by "**-**" below) of the two computing paradigms.

The advantages and disadvantages of mobile computing can be characterized as follows.

**+** Real-time situational information: Smartphones are equipped with numerous sensors which can be used to collect real-time information for the relevant applications. Example application scenarios include: discovering a disaster rescue scene, where the rescue team members can take pictures that may be transferred to the cloud that uses image-processing techniques to recover the entire scene.

**+** Mobility-enabled opportunistic security: It is often said that mobility makes security harder to achieve. However, there are scenarios where mobility can actually help achieve better security. One example is the establishment of cryptographic key infrastructure that would not be possible without exploiting mobility [13, 14]. Another example is that smartphones can share security intelligence (e.g., attack alerts, blacklist of malicious websites, security patches) in a peer-to-peer fashion.

**+** Rich applications: Smartphones are getting increasingly popular perhaps because the rich applications (apps) and the ease of installing and configuring them. This application richness is characteristic of smartphone-based mobile computing, and might change the landscape of IT services in the future in a much broader sense.

- Rich applications or large number of untrusted third-party developers: The side-product of the application richness is that most of the app developers cannot be trusted. Indeed, the smartphone apps markets are full of malicious apps that attempt to conduct various malicious activities, including stealing smartphone users' private information.
- Limited battery lifetime: This not only discourages smartphone users from doing "community services" (e.g., unwilling to forward others' packets or participate in crowd compute tasks), but also makes the smartphone users reluctant to conduct some battery-heavy tasks (e.g., heavy-duty authentication protocols for verifying that the remote communication peer is indeed trusted).
- Limited communication capabilities: While the bandwidth of smartphones has been increased substantially, the communication capabilities can be limited by the scarce resource at the wireless base stations. This would render tasks that involve large volume of communications impractical.
- Poor physical security: Smartphones can be easily lost, stolen or tampered with. Despite that there are screen protection mechanisms, many users do not even bother to use it, perhaps in fear of the kind of "benign" denial-of-service attacks (i.e., if a user enters the passcode wrong for a few times, the smartphone will be locked).

On the other hand, the advantages and disadvantages of cloud computing can be characterized as follows:

+ Essentially unlimited computational power: Cloud computing offers elastic compute resources that can be allocated on demand and at a low cost. The agility in allocating essentially unlimited compute and storage resources if often desired in many applications that require to accommodate bursty needs.
+ Essentially unlimited communication capabilities: In contrast to mobile computing, cloud data centers can achieve much higher communication bandwidth, which facilitates high-performance and coordinated computing tasks. This is especially relevant when the computing tasks (e.g., image processing) need to use a large number of computers (or CPU cores, virtual machines) that may need to interact with each other during the process.
+ Good physical security protections: Cloud data centers are well protected from physical intruders. As a result, they are relatively less vulnerable to physical attacks, including attacks that attempt to tamper the hardware systems that may have some embedded secrets such as cryptographic keys.
- Distant from spots of interest: Cloud data centers are dispersed nation-wide or even world-wide to mitigate the damage of potential natural disasters. This also means that they are often far away from the "hot spots" of interest, and that services often require heavy communication duties between the "hot spots" and the data centers (e.g., multimedia data transferred to cloud data centers for automatic analysis). In particular, this imposes a profound constraint on the smartphones when they use 3G/4G/LTE techniques to access the resources in the cloud data centers.

- Insider threats: Cloud data centers are vulnerable to insider threats. Although the insiders may not be able to launch hardware attacks against the cloud computers (e.g., data centers are under surveillance), they could abuse their privileges to compromise, for example, the confidentiality or integrity of the data stored in the cloud data centers (see, for example, [15, 16, 17, 18]) or even the computing results (see, for example, [19, 20]).

Based on the above discussion, we have the following definition of mobile cloud computing.

**Definition 1.** *(mobile cloud computing system)* From the perspective of system components, we consider a mobile cloud computing system as the compute system where the user-end is dominated by mobile devices (especially smartphones), and the cloud-end data centers consist of a large number of computers that can be allocated elastically on-demand. The communication channels between the user-end and the cloud-end are typically a hybrid of wireless channels (often the access network of mobile devices) and wired channels (within the data centers and between the wireless access points and the data centers). From the perspective of system functions, a mobile cloud computing system accommodates the aforementioned three scenarios (i.e., using mobile devices in cloud computing, mobile cloud computing based on mobile devices, hybrid of the two scenarios). Moreover, mobile cloud computing aims to achieve the best of both worlds, namely to inherit the advantages of both mobile computing and cloud computing, while overcoming the disadvantages of mobile computing and cloud computing at the same time (i.e., achieving the "$1 + 1 > 2$" effect).

**Federated mobile cloud computing.**

By "federated mobile cloud computing" we mean that multiple autonomous clouds aim to work with each other to provide better services and security to their users. Unlike the case of *federated database*, where the primary goal is to provide a virtual central database to the users, federated clouds are peers to each other and there is no effort to have a central control over the clouds. Nevertheless, there are decentralized and distributed coordination entities that represent the corresponding clouds and coordinate the activities of the federated clouds when the need arises. In this chapter, we use the following definition of federated mobile cloud computing.

**Definition 2.** *(federated mobile cloud computing system)* From the perspective of system components, a federated mobile cloud computing system consists of multiple autonomous mobile cloud computing systems, which are peer to each other but would be cooperative in providing better services and security by coordinating their activities and sharing their resource on demand. Essentially, federated cloud computing aims to achieve the "$1 + 1 > 2$" effect at the cloud level, namely that multiple clouds can collectively accomplish tasks that are otherwise infeasible, or even impossible, to achieve by the clouds individually.

## 2.2 Some Killer Applications of (Federated) Mobile Cloud Computing

Now we discuss some killer applications of federated mobile cloud computing, which are infeasible or even impossible without simultaneously taking advantage of both mobile computing and cloud computing.

**Saving lives.**

People often talk when they walk. This can be dangerous especially when they cross roads. It is possible to use the smartphone camera to detect the cars that are approaching the walker (based on the appropriate image processing technology), and then alert the walker [21]. As another example, there are reports that nearly 50% high schoolers text when they are driving [22]. One possible solution is to use smartphones to monitor the moving speed of smartphones while disabling the texting function when the phone is moving at a high speed (e.g., under parent control) [23].

**Better lives.**

Suppose Jon, a child attending the Macy's Thanksgiving day parade, is lost in the crowd. How can his parents find him? One solution is, possibly through the assistance of the police on the spot, to ask the people attending the parade to take pictures of children they spot. The pictures are then sent to the cloud, which uses some appropriate image recognition algorithms to automatically identify which pictures would be likely the lost Jon. The cloud can send this feedback to the people who took those pictures, who can then work with the police and Jon's parents to find Jon. This application was proposed in [24]. Another example, which was first proposed in [11], is that when you travel to a foreign country for sightseeing. You may see many signs you do not understand because you do not speak that foreign language. With your smartphone camera, pictures of street signs can be automatically translated and overlayed in the language of your choice. In this case, federation of clouds cloud enhance the user experience by shortening the response time (e.g., using the compute resources that are physically closer to the spot).

**Situational authentication.**

Consider a battlefield. Suppose a solider is captured by the enemy, who would want to use the smartphone of the solider to access the data center in the cloud. Given that the soldier's life is at stake, the solider would have to give the enemy whatever means that are used for authenticating the solider to the cloud (e.g., password, fin-

gerprints, biometrics). As a consequence, the enemy can perfectly impersonate the solider in terms of accessing the data centers.

With mobile cloud computing, the cloud can enforce the following type of *situational authentication*: The cloud can, based on some appropriate policy, ask the solider to take pictures of the people (or cryptographic endorsement by other soldiers whom the solider is staying with) or background scenes surrounding him. By image processing in the cloud, it is possible to check whether the people surrounding the solider are some of the people who are supposed to be with the soldier. This would detect, at least in some scenarios, that the solider is already at the hands of the enemy and and requests from the soldier's smartphone, despite that is can pass all technical authentications, should be disabled. This method of authentication may be augmented with other situational parameters such as location, which can then be factored into the access control decision. Location in particular can be measured by either the mobile device itself (e.g., GPS), or by the radio devices it is communicating with (i.e., radio location).

**Rapid information/intelligence/resource sharing.**

Consider two clouds that are federated. It is important that they are willing to share information or intelligence in terms of the attacks against them. This is because, for example, an Advanced Persistent Threat (APT) or zero-day attack detected at one cloud can be rapidly shared with the other cloud, which can possibly adopt the appropriate countermeasures to cope with the attacks. This is especially true for sharing information about attacks against mobile devices, because the compromise of cryptographic credentials in a mobile device can allow the attacker to perfectly impersonate the victim user to the multiple cloud services the user is a customer of. Moreover, the compromised mobile devices can become a powerful attack tool against each other and directly or indirectly against the clouds themselves. Another advantage of federation is that they can share resources, especially the compute and storage resources at the cloudlets — an architectural component that will be elaborated later — to better serve the could users. For example, when one cloud's user travels to a new place where the closest cloudlet is operated by another cloud, the user can use the other cloud's cloudlet infrastructure via local WiFi communications.

**Healthcare.**

In healthcare applications, a patient's human body may have multiple embedded sensors (also known as Implantable Medical Devices or IMDs [25, 26, 27]) that monitor one's health conditions. The sensors have very limited communication power and battery supplies. Compared with traditional sensors, healthcare sensors are even more constrained because, for example, replacing battery or an embedded sensor would require invasive surgery [25]. It is still under active research how to se-

cure such systems, including facilitating the security communication between such devices.

The monitored data might need to be shared with the patient, the patient's doctor, or the emergency doctor. For this purpose, it would be natural to use patient smartphones as the proxies between the sensors and the cloud. For example, when the monitored parameters are too high or too low, the sensors can let an appropriate smartphone app know so that the smartphone can automatically alert the patient and possibly automatically contact the emergency room or the patient's doctor with the reads as well. Indeed, there are several advantages for using smartphones as a proxy of the body sensors. First, the communication channel between the sensors and the smartphones is short, meaning that the communication-caused power consumption at the embedded body sensors is small. Second, the asymmetry of communication, compute and storage capacities between the sensors and the smartphones can be exploited to design security solutions to better protecting the patient's private health data. Third, a patient smartphone can manage all the sensors that have been embedded into the patient's body. This type of centralized management would be beneficiary, especially because the patient's private health data would need to be stored in the cloud anyway.

## *2.3 Modeling Federated Mobile Cloud Computing*

Figure 1 highlights the model of federated mobile cloud computing. The model consists of four types of functional components: mobile devices or smartphones, wireless communication and the corresponding cloudlets (where wireless communications can be based on 3G/4G/LTE and wireless LANs), wired communication infrastructure (for connecting the wireless communication base stations and access points to the clouds), cloud data centers and their associated distributed federation entities (where the federation entities are responsible for sharing information and coordinating activities between the clouds).

In the model, the mobile devices or smartphones are manufactured by different vendors and owned by different users or enterprises. The wireless communication facilities and the corresponding cloudlets are operated and maintained by different service providers. The wired communication infrastructure is provided by other third party vendors. The cloud data centers are operated by different cloud vendors for various cloud services, including PaaS (Platform as a Service), IaaS (Infrastructure as a Service), and SaaS (Software as a Service).

We want to highlight the importance of deploying cloudlets in (federated) mobile cloud computing. The main motivation for introducing cloudlet is to make compute resources closer to the mobile users so as to reduce the unnecessary communication overhead [9, 28, 29]. Moreover, the wireless service vendors might have the incentives to deploy compute and storage devices at the cloudlets to reduce the consumption of their own resources in the wireless access network. As such, cloudlets are a useful architectural component that can bridge the resource gap between cloud

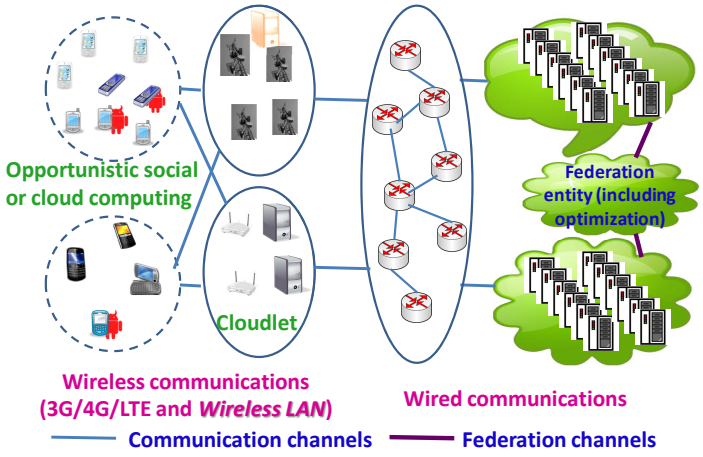


**Fig. 1** Model of federated mobile cloud computing

computing and mobile computing [30]. This explains why we believe cloudlets will play an important role in (federated) mobile cloud computing.

## 3 Security Architecture for Federated Mobile Cloud Computing

In this section we first discuss the design challenges of a security architecture for federated mobile cloud computing systems, with emphasis on the unique and emerging threats against federated mobile cloud computing. We then discuss the set of guiding principles that we have derived from these challenges. Finally, the security architecture itself is presented, along with running examples that show how the security architecture can be instantiated to serve some representative needs.

### *3.1 Security Architecture Design Challenges*

Now we elaborate the challenges that are encountered in designing security architecture for federated mobile cloud computing. To the best of our knowledge, these challenges are not systematically examined until now.

**Challenge 1: Unique threats.**

First and foremost in the design of any security architecture should be a detailed and thoughtful consideration of the threats it will face. We want to assure the functioning of federated mobile cloud computing, as however specified, despite the following kinds of attacks. We classify threats against federated mobile cloud computing into the following categories.

**Threats against the cloud data centers and their associated federation entities.**

Attacks against the cloud-end can be launched by compromised smartphones and/or compromised cloudlets. The attacks may attempt to compromise the confidentiality and integrity of the data stored in the cloud data centers. In return, compromised cloud data centers can host malicious websites to attack the innocent but vulnerable smartphones.

The state of the art protection of the cloud-end systems is summarized as follows. There has been security architectures for protecting the clouds, including [31, 32, 33, 34], which are however mainly geared towards securing the cloud-end. For example, the security architecture presented in [31] systematically accommodates defenses at three different layers: application, network, and system. Application-layer defense aims to detect malicious websites that are hosted by the clouds, ideally as early as possible. This is important because malicious websites have become a major vector of cyber attacks (e.g., hosting botnet C2 severs [35], cracking passwords [36], spamming in cloud-like settings). Network-layer defense includes the use of honeypots in some novel ways, such as blending the honeypot IP addresses into production networks and shuffling the honeypot IP addresses to make it hard to evade by the attacker. System-layer defense aims to cope with Advanced Persistent Threats (APT) or APT-like sophisticated and stealthy attacks, by enhancing VM image management and VMM security [37, 38, 39, 40]. There are also some security architectures for securing the user-end mobile devices [5, 41]. Moreover, there have been discussions on security architecture for mobile cloud computing [42]. However, to the best of our knowledge, there are no systematic security architecture for (federated) mobile cloud computing systems.

**Threats against the smartphone devices.**

Smartphones may be compromised through many means. First, we consider those threats that relate to the smartphone devices themselves, as opposed to the software running thereon. By their nature, it is very easy for a smartphone to be lost, stolen or tampered with. As a result, physical access is a major and very powerful attack vector. Attackers with physical access can tamper with the system and do things like inserting a malicious Secure Digital (SD) card, manipulating the bootloader, flashing trojanized firmware, or accessing the device via privileged debugging inter-

faces. For example, Android lockscreens are notorious for the many ways in which they can be bypassed [43]. Finally, because of their inherent mobility, smartphones may connect to many different networks or serial (e.g., USB) links throughout the course of a typical day. Many of these connections, especially USB, are assumed to be trusted and can provide privileged access to the device and operating system [44]. Each outside connection represents an opportunity for an attacker to compromise the device, such that when it reconnects back to the mobile cloud computing system, the attacker's foothold is propagated into the cloud.

In addition to these threats, there are many others stemming from the rich and less-controlled ecosystem that exists for mobile devices or smartphones. For example, because of their discretionary access controls and strong dependence on the owners' non-expert administrative decisions, malware can be unknowingly installed and attack other apps and data on the devices. This category of vulnerabilities is discussed separately, below.

**Threats against the cloudlets.**

Attacks against cloudlets can be launched by compromised clouds and/or by compromised smartphones. Consequences of compromised cloudlets include: the users' data is compromised when processed by the cloudlets. This is particularly devastating when the computing tasks outsourced to the cloudlets involve some cryptographic credentials (e.g., private keys for decryption or digital signing). This aspect has not been studied with the due attention, despite that some of the solutions for securing the cloud-end may be naturally adopted (or adapted) to securing the cloudlets. The more tasks that are conducted at the cloudlets, the bigger attack surface at the cloudlets. Moreover, cloudlets enlarge the vulnerability surface of (federated) mobile cloud computing to insider threats because cloudlets are not present in common cloud computing systems.

**Challenge 2: Data diversity.**

A diverse kinds of data is involved in federated mobile cloud computing. One kind of data is owned by the users that outsource their personal and private data to the clouds. In addition to that the users will access their own data, they may choose to share some of their data with some of the other users. Another kind of data is owned by the enterprises that outsource their business data to the clouds. The enterprises not only will need to allow their employees to maintain and process their data, but also might need to share some of their data with their business partners. Yet another kind of data is collected, and outsourced, by a group of mobile device users, who work together to accomplish some tasks (e.g., emergence rescue as discussed above). It is not even clear who owns this kind of data, let alone who should define access control policies for protecting this kind of data. Due to the diversity of the data that is involved in federated mobile cloud computing, it is both hard and pos-

sibly unwise to impose a single access control mechanism for managing the data in any security architecture. Instead, we will highlight the advantages and disadvantage of two kinds of access control mechanisms in the setting of federated mobile cloud computing, namely Mandatory Access Control (MAC) and Discretionary Access Control (DAC).

**Challenge 3: Application richness.**

The most attractive feature of mobile computing is perhaps the rich set of apps that are (almost) free to use. However, the apps are often developed by third-party developers that may not be trusted or even may be malicious. Despite that smartphone platform designers have attempted to impose strong restriction on the apps that can be shared through the so-called marketplace, only limited success can be expected especially because of the magnitude of the number of apps. As a consequence, smartphone malware might become, if not already, a major threat vector against (federated) mobile cloud computing.

To be more specific, let us consider the most popular smartphone platform, namely Android. Android was designed to enforce strong isolation and permission-based access control, but could not effectively deal with attacks that exploit indirect and implicit delegations. This makes Android especially subject to various privilege escalation attacks, such as:

- Permission abuse [45, 46, 47]: Malicious apps can abuse their permissions to steal or compromise unauthorized sensitive information.
- Confused deputy attacks [48]: A malicious and unauthorized app exploits the unprotected/vulnerable interface of a legitimate and authorized app to bypass the permission-based access control.
- Colluding attacks [49, 50, 51, 52]: Multiple malicious apps maliciously combine their authorizations or permissions to conduct activities that are not authorized with respect to each individual of them, perhaps via some covert channels.
- Bypassing Android's built-in defense [53]: A malicious app could exploit the underlying kernel-facilitated channels to bypass Android's built-in defenses.
- Battery drain attack: As demonstrated in [54], an app can intentionally consume the battery of a smartphone. While countermeasures based on detecting anomaly use of battery can defeat certain abnormal uses, it is conceivable that anomaly-based countermeasures can be evaded by sophisticated attacks as in the case of intrusion detection.

The state of the art protection of smartphones is summarized as follows. There have been many solutions that have been proposed to address specific problems, such as: detecting malicious apps in the market [55], enhancing privacy [56, 57], mitigating colluding attacks [53], and mitigating permission abuses [58]. Despite these efforts, we suspect that the attackers can still find ways to penetrate into the systems. Moreover, usability issue could hinder the deployment of even advanced defense techniques. There have been studies on identifying trade-offs be-

tween smartphone security and usability/convenience [59, 60], with emphasis on authenticating the human users to the smartphones. Our study assumes that such human-to-smartphone authentication is already enforced, either by education or by commandant as in the case of government/enterprise employees using smartphones to visit organizational resources.

**Challenge 4: Resource constraints.**

While smartphones are already powerful in terms of compute and storage capacities, they are limited by the short lifetime of batteries. This not only disincentivize the users for conducting battery-heavy computational or communication tasks for normal needs, but also disincentivize the users for deploying security tools that are computationally heavy. In particular, this can affect malware detection algorithms at the smartphone end. Although there have been mobile-specific versions of anti-malware tools, they are similar to their desktop counterparts especially in terms of power and resource consumption [61]. For example, it was reported that the procedure for initializing the malware signature database on Nokia N800 mobile devices would take 57 seconds and consume 40 MB memory [62]. It is reasonable that the users might turn off such security tools to save their battery consumption.

**Challenge 5: Federation.**

Federation of clouds imposes many barriers. For example, how can one cloud allow the customers of another cloud to use its resources while assuring security and proper billing? This is especially true when the clouds that are involved are heterogeneous, meaning that they may deploy different authentication methods. Consider that a cloud user, Alice, outsources her data to Cloud $X$. How can Alice share her outsourced data to Bob, who may be a user of Cloud $Y$, without exposing her data to clouds $X$ and $Y$? How can cloud $X$ and $Y$ rapidly share security information to make both $X$ and $Y$ more secure than when they do not form a federation?

**Summary.**

In Figure 2, we summarize the above exploration of the challenges that are encountered when designing security architecture for federated mobile cloud computing. Specifically, we note that federated cloud computing is harder to secure than cloud computing because of the above Challenge 4. Smartphone-based mobile computing is harder to secure than cloud computing because of the above Challenges 1-3. Smartphone-based mobile computing is harder to secure than mobile computing (MANET) because of the above Challenge 2. Indeed, smartphone-based mobile computing is also harder to manage than military MANETs because of the higher mobility uncertainty of smartphone users [63]. Federated mobile cloud computing

is harder to secure than federated cloud computing because of the above Challenges 1-4. Federated mobile cloud computing is harder to secure than smartphone-based mobile computing because of the above Challenge 4.

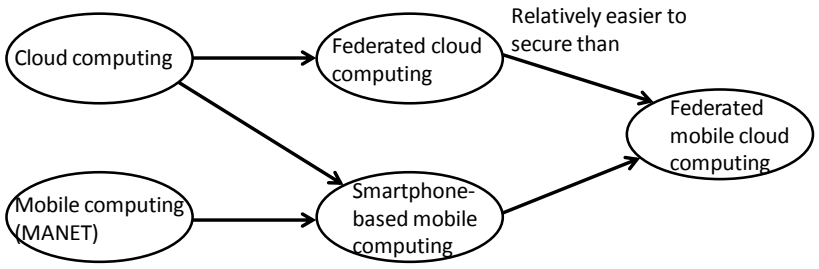


**Fig. 2** Relative hardness in assuring "secure *-computing": $A \rightarrow B$ means that it is harder to secure $B$ than to secure $A$. The relative hardness result is caused by the "$1+1>2$" effect from the attacker's perspective. For example, the rich third-party apps make smartphones a bigger threat to the clouds than the smaller number of applications that are developed by trusted parties for non-mobile cloud computing.

## *3.2 Security Architecture Design Guiding Principles*

In order to secure federated mobile cloud computing systems, despite the challenges and threats mentioned above, we need to have some design guiding principles. In what follows we discuss our guiding principles.

**Principle 1: A security architecture should be flexible and comprehensive, but not over flexible/comprehensive.**

A security architecture should not impose restrictions on the desirable features of mobile devices and clouds, and it should minimize introduction of undesirable features. For example, strengths of mobile devices include mobility, personalization and rich sensors, while weaknesses are that they are easily lost, stolen or tampered with, and that they have weaker security features than larger, fixed computing platforms. In contrast, clouds and data centers enjoy strong physical security, essentially unlimited storage and computation, and high reliability. However, clouds suffer from the fact that they are not necessarily located where they are needed, are shared by multiple interests, and may be inaccessible due to network outages or degradations.

A security architecture should be flexible, meaning that it can accommodate not only today's security technologies, but also tomorrow's security technologies that are yet to be developed. Moreover, a security architecture should be able to seam-

lessly integrate various "point" solutions that are often geared toward a very specific attack under a very specific premise.

However, a security architecture cannot, and should not attempt to consider all relevant services. Certain realistic assumptions about the environment and underlying substrates should be, or must be made. A high level security architecture may need to assume that attacks against the communications substrates, wireless communication and wired communication alike, will be addressed at a different layer of abstraction. If this is the case, then users must recognize that the architecture is only as good as this assumption. Alternatively, the architecture could be designed to operate *in spite of* security problems in the substrate. In fact, if the architecture relies on commercially-provided connectivity, storage or other resources, there is a good chance that the provider will be reluctant to provide customers with full transparency regarding the security of the service. Hence, these aspects must be regarded as a security "black box" and the architect must be very careful when making assumptions about them.

For our purposes, we assume that the underlying communication substrate can achieve the communication goals (i.e., the terms of the service level agreement), especially, end-to-end communication capabilities, despite the attacks (e.g., denial-of-services) against them. However, we do not assume anything about the confidentiality of the substrate.

The above discussion suggests us to consider the notion of *security services*, rather than concrete security mechanisms. This is because a security service can be fulfilled by many different security mechanisms, where new mechanisms may replace the old ones after some period of time. This flexibility allows to replace the security mechanism building-blocks in a plug-and-play fashion.

**Principle 2: A security architecture should facilitate security co-design.**

We advocate security and cryptography co-design, which are often separately designed. This is important because security can help detect compromised cryptographic keys, namely increasing trustworthiness of cryptographic assertions. Cryptography can help detect system compromises: Honey cryptographic credentials on smartphones and cloudlets can be used to detect system compromises. Moreover, there are practical solutions to verify the storage integrity of data outsourced to the cloud (see, for example, [15, 16, 17]). These cryptographic solutions can be exploited to detect very efficiently whether the outsourced data has been tamper with by the attacker who has penetrated into the cloud data centers, or by the insider administrator of the cloud data centers. Yet another example is to exploit system security for enhancing the trustworthiness of cryptographic digital signatures [64]. This is important because the trustworthiness of digital signatures is fundamental to essentially all digital communications. Finally, we mention that it is even possible to exploit anonymous communication techniques to "blind" the attackers so as to reduce their capability of launching adaptive attacks to launching random attacks [65]. This can lead to much better resilience against sophisticated attackers.

**Principle 3: Agility-comprehensiveness trade-off.**

Effective cyber defense requires to respond to cyber attacks rapidly. This means, among other things, that the decision-making process and the countermeasure deployment process must be agile. However, this also would mean that the decision-making process must be automated and cannot rely on comprehensive global situational information. That is, we need to achieve a good agility-comprehensiveness trade-off. A key concept that we introduce in the present paper is called *fast/slow two-tier OODA Loops*, which allows to achieve effective trade-offs between agility and comprehensiveness.

**Principle 4: There is no fundamental difference between private clouds and public clouds in federated mobile cloud computing.**

We observe that while private clouds (e.g., DoD clouds) can be trusted, they can be compromised as well. It is often believed that public clouds are not trusted, but private clouds are trusted. While this is true from certain perspectives, it is not true in general because private clouds, while trusted, can be compromised by insiders or by malware such as APT. Moreover, cloud computing is not, by default, more secure than the other distributed computing systems that have been deployed, but its uniformity may ease securing it [4]. Since no deployment of cloud computing is universally suitable for all DoD applications, sensitive and time-critical DoD applications should not be deployed in public clouds [4].

## *3.3 Security Architecture*

**Security architecture components and flows.**

Figure 3 describes the security components and data flows in the proposed security architecture. In this architecture, each smartphone, each cloudlet, and each cloud data center is associated with a *security control center*. The security control centers are responsible for communicating with the other control centers. For example, a control center of a smartphone can communicate with the control center of a cloudlet and can communicate with the control center of a cloud data center. In addition, each cloud maintains a *command and control* (C2) center. The C2 centers communicate with each other for sharing data and coordinating activities. Ideally, this C2 channel will be instantiated as a separate *out-of-band* connection that allows critical C2 functions to persist even when connectivity across the data channels is lost or degraded due to an attack. This separation of control and data planes could be virtual (e.g., a separate virtual private C2 network) or even physical (e.g., satellite communication C2 links) for high assurance applications.

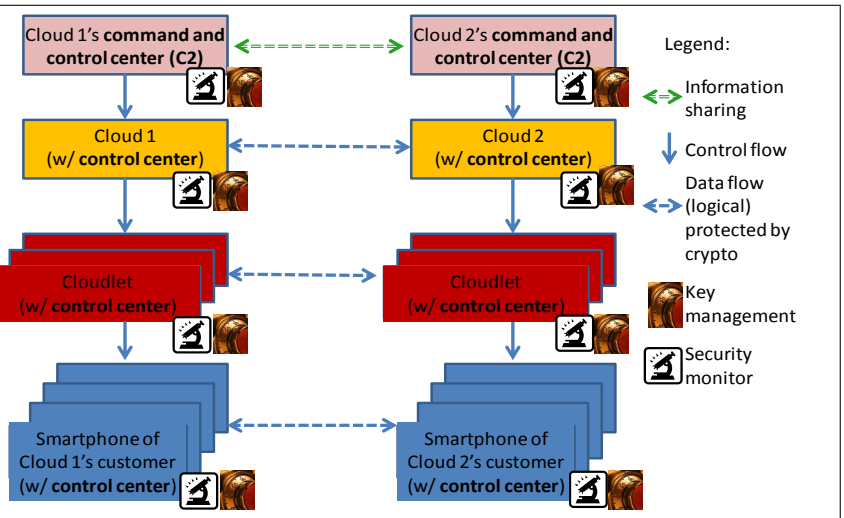


**Fig. 3** Security architecture centered on (i) the C2 centers at the clouds for federated coordination purpose, and (ii) the security control centers (also called control centers for short) deployed at the smartphones, the cloudlets, and the clouds (for intra-cloud security management): The C2 centers of different clouds can conduct peer communications with each other. The dashed arrows between the security control centers of different clouds, between the cloudlets, and between the smartphones represent the logical data flows, which may or may not need to go through the C2 centers. The typical scenario is the following: A secure, cryptography-protected logical communication channel, between the security control centers is first established by the C2 centers, which allows secure communication between the control centers of different clouds. If needed (e.g., when the smartphones cannot authenticate each other), the control centers can help the cloudlets and the smartphones establish authenticated communication channels.

The security control centers are middleware that reside in the software stack of the smartphones, the cloudlets, and the clouds (i.e., data centers). The functions of the security control centers are the following:

- The security control center at a cloud will coordinate the intra-cloud security activities with the cloudlets and smartphones that are connected to the cloud. This control center also communicate with the C2 center at the same cloud.
- The security control centers also serve as the activators in the security architecture. The security control center in each cloud is responsible coordinating the defense activities within the cloud itself, the defense countermeasures at the cloudlets, and the defense mechanisms at the smartphones of the corresponding cloud users. The security control centers also communicate with the C2 center of the same cloud. For example, the security control centers at the smartphones can deploy new patches.
- The security control centers monitor the activities of the affiliated components, and report the monitoring results to the appropriate entities such as the human user and the cloud control center. Since preventive defense cannot be perfect,

smartphones can be penetrated, while cloud computers or virtual machines or even hypervisors can be compromised. Monitoring for abnormal behaviors at the respective components at the right level of abstraction is an important means for detecting zero-day attacks and Advanced Persistent Threats (APTs). Security information sharing between the security control centers at the smartphones could lead to more effective detection of such advanced stealthy attacks.

**The C2 center middleware.**

The functions and uses of the C2 center middleware for federated clouds, which do not necessarily involve smartphones though, has been explored in [31]. The C2 center described in [31] consists of the following modules. The *information gathering* module aims to capture dynamic attack information from deployed defense mechanisms, including honeypot and forensic analysis and possibly crossing the application-network-kernel layers. The *information analysis* module, which is the core of the C2 centers, aims to learn and characterize the dynamic threats. The *information feedback* module decides where to deploy the countermeasures, which may be received from a peer C2 center at another cloud or generated by the cloud in question. The *information sharing* module aims to effectively share information about new attacks and countermeasures between the C2 centers. The functions of these modules will mainly remain the same as elaborated in [31]. For example, upon receiving countermeasures against a new attack from a federated cloud, the C2 center will instruct the deployment through the security control centers that are associated with the cloud. Nevertheless, extensions need to be done to collect and exchange situational information about the smartphones and the cloudlets, which were not necessarily considered in [31]. For example, the new attack information collected at the smartphones (e.g., new malware or zero-day attacks) can be rapidly shared between the federated clouds so as to deploy effective countermeasures before it is too late.

Ideally, the C2 center middleware has the following characteristics:

- Fully automated process: As cyber attacks evolve rapidly, we cannot rely on manual analysis and decision-making for cyber defense.
- Easy-to-use control panel: The functions are self-explaining and user friendly.
- Protected control channels: When the federated mobile cloud is under attack, it is critical that communication between the C2 centers persists. In very critical applications, this control channel may even be physically separate in order to completely decouple it from the cloud data channels.

**The security control center middleware.**

Figure 4 shows the security architecture middleware, which is implemented and deployed at each control center. In this architecture, we assume that the hypervisor or OS is extended to incorporate security mechanisms at such low level. The in-

corporated mechanisms can include the current standard practice of anti-malware software, which often operates at the OS kernel-level to detect suspicious activities. This detection layer is responsible for representing new attack information that can be rapidly shared with the other control centers. Tailored apps can directly utilize some services of the middleware, such as group key management. The control panel of the control centers offers the user of a smartphone, the administrator of a cloudlet or cloud to configure the security middleware. The security architecture middleware consists of the following sub-layers (from top to bottom):

1. Secure offloading.
2. Secure data sharing.
3. Fine-grained data protection.
4. Countermeasure deployment.
5. Access control.
6. Key management infrastructure.
7. Attack identification, reporting and countermeasures.

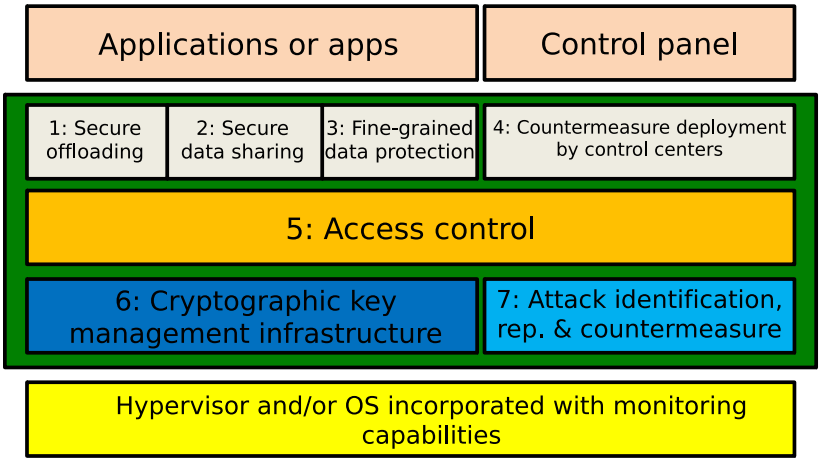


**Fig. 4** Security control center middleware at the smartphones, cloudlets and clouds (shown with green background). As we will elaborate, not all modules/services in the security control center middleware are deployed at every level of the architecture (i.e., smartphone, cloudlet, and cloud).

It would be ideal that the security control center middleware at the smartphones and cloudlets are automated and completely transparent to the smartphone users and the cloudlet administrators, respectively. However, to ease the privacy concerns of smartphone users and the cloudlet owners/administrators, there should be an ease-to-use control panel by which the users and cloudlet administrators can easily manage or maintain the security control center middleware as well as the deployment of countermeasures that are received from the cloud-end security control centers or received from the security control center middleware vendors.

In the following discussion, each of the seven middleware modules are discussed in more detail. Module numbers correspond to the numbered blocks in figure 4.

**Security control center middleware service/module 1: Secure offloading.**

This module is responsible for assuring securely offloading some computational tasks from the smartphone to the cloudlet or to the cloud. There are several kinds of offloading scenarios:

- Offloading general programs to cloudlets or clouds: There have been many studies on offloading general programs to the cloudlets or clouds, including the coarse-grained VM migration and the fine-grained process migration (e.g., the aforementioned systems such as ThinkAir [6], MAUI [7], CloneCloud [8]). However, existing studies did not consider the situation that the cloudlets or clouds that run the offloaded programs are compromised. While this can be alleviated by running the offloaded tasks in some trusted environments, future research should seek better solutions.
- Offloading cryptographic computations to cloudlets or clouds: There have been many studies on securely offloading cryptographic tasks. A feature of such offloading tasks is that the user (i.e., mobile device in this case) can verify that the cloudlets or clouds did not cheat by deviating from the execution of the desired cryptographic functions. This is often achieved by exploiting the homomorphism properties of the cryptographic functions in question. For example, there has been substantial new progress in verifiable outsourced computing: a user can verify that the cloud does not cheat when conducting SQL queries over an outsourced relational database [19]; a user can verify that the cloud does not cheat when conducting some queries over some outsourced encrypted data [20]. As mentioned earlier, a cheating cloud does not necessarily mean that the cloud is malicious by default. In contrast, the cloud can be a trusted one, but may be compromised later by sophisticated attacks and fall under the control of the adversary.
- Offloading malware detection to cloudlets or clouds: Current anti-malware techniques that are deployed on desktop platforms might not be appropriate for deploying on smartphone platforms. This is because scanning of smartphones with respect to a large number of malware signatures can consume too much smartphone resources, including the batteries. Therefore, there have been proposals for offloading virus scanning services to the cloud. For example, the ThinAV system [66] is a cloud-based lightweight anti-malware service for Android via a small client running at the Android end that can take advantage of cloud-based anti-malware services (e.g., `virustotal.com`). However, the key issue here is to protect privacy because scanning for signatures, or even dynamic analysis of app execution behavior, may expose smartphone users' data to the clouds.
- Offloading heavyweight security analysis tasks to cloudlets or clouds: Although smartphones may be more vulnerable than their fixed counterparts, it is difficult to compensate for this with extra on-board security functionality because of the lightweight nature of the smartphone platform. Because our architecture in-

cludes cloud resources, it makes good sense to leverage these resources not only for applications, but also for the security functions of module 7 (discussed later). Several proposals have been made for offloading security analysis functions to tightly integrated cloud resources. For example, Paranoid Android allows the smartphone user to benefit from the application of multiple simultaneous analysis functions by hosting an exact replica of the device in the cloud [67]. These functions could include real-time file scanning, multi-sensor correlation, sophisticated information flow analysis, and behavior characterization.

This module is deployed at the smartphones, the cloudlets and/or the clouds, dependent upon the applications.

**Security control center middleware service/module 2: Secure data sharing.**

This module is responsible for securely sharing data as requested by the apps or the control panel. In order to control the use of the shared data, there are two methods:

- Explicitly enforcing access control: In this case, the principals (users, programs or apps) must get explicit authorization before having access to the data in question. Traditional access control methods, including Discretionary Access Control (DAC), Mandatory Access Control (MAC), and Role-based Access Control (RBAC), fall into this category. This method is particularly appropriate for pull-based data sharing, where data seekers/consumers contact the data owners/administrators for requesting to access the data. This kinds of data sharing often requires the execution of a cryptographic authentication protocol, which is often accompanied with a key exchange function by which a security channel can be established for securely transmitting the data.
- Implicitly enforcing access control: In this case, the principals get their due credentials, which may correspond to their identities in DAC, clearance and category in MAC or roles in RBAC. The data owner will encrypt the data according to an appropriate policy such that the encrypted data can be arbitrarily disseminated, because only the users whose credentials satisfy the policy can decrypt the encrypted data. This method is particularly appropriate for push-based data sharing, where the data owners/distributors can publish the encrypted data (e.g., through a broadcast channel) at any time as needed. This kind of data sharing can be achieved using the cryptographic technique known as attribute-based encryption (see Cryptographic Infrastructure below), where attributes correspond to the credentials mentioned above.

This module is deployed at the smartphones and the clouds, and the cloudlets when they participate in helping conduct data sharing.

**Security control center middleware service/module 3: Fine-grained encryption of private data.**

This module is responsible for fine-grained encryption of private data, especially the private data stored on smartphones. Fine-grained encryption aims to mitigate the damage caused by malware that has penetrated into the smartphones. One appealing idea is to enforce mission-based access control, meaning that the private data stored in smartphones is partitioned into portions with respect to different missions. When a certain data item needs to be accessed or processed, the smartphone needs to work together with the cloud in order to the decrypt the data item. This can be achieved by, for example, splitting the private decryption keys into two shares such that one share is held by the smartphone and the other share is kept by the cloud [68]. Further enhanced security can be achieved using some more advanced techniques [69, 70], which can mitigate the damage caused by the compromise of private keys at the smartphone end. Moreover, the aforementioned situational authentication technique can be used to enhance the authentication of the smartphone to the cloud, so as to mitigate the damage caused by that the adversary physically possesses the smartphone in question.

A variant of the above mission-based fine-grained access control is to enforce fine-grained protection of data pushed to the smartphones. For example, consider a squad of soldiers in a battlefield. When the commander sends an order to the squad, the commander may want to ensure that only when multiple soldiers are present, the encrypted order can be decrypted (i.e., no single solider can decrypt the order by itself). This is relevant because the mission may indeed need multiple soldiers to accomplish. In this case, the order can be encrypted using threshold cryptography that ensures that a certain group of soldiers must work together in order to decrypt. Another related idea is to enforce location-based access control of data. For example, a data item in a smartphone can be encrypted so that it can be decrypted only when (for example) the smartphone is in certain service zones [71, 72, 73].

**Security control center middleware service/module 4: Automatic countermeasure deployment.**

The module aims to automatically deploy countermeasures at the smartphones, cloudlets and clouds. The automated deployment is meant to keep up with the speed of cyber attacks, because manual deployment is often too slow to be effective. As an example, at the smartphone end, Google is already able to force the installation/deinstallation of apps/software on Android [74]. This can be enforced at the "control center" at the smartphones. This remote installation/deinstallation manifests the function of control centers at mobile devices. The control centers can further be extended to patch known vulnerabilities, install security countermeasures received from the cloud control center or the cloud C2 center. An example of this is, at the cloudlet end, Microsoft's capability to automatically upload security patches, with little user involvement. This same automatic update technology can be equally

applied to update, for example, virtual machines images at the cloud end. Therefore, this module can be deployed at the smartphone end, the cloudlet end, and the cloud end.

**Security control center middleware service/module 5: Access control.**

This module is responsible for enforcing access control that is needed to protect the data and resources that may be accessed by various applications or apps. Access control may be involved at multiple levels of the entire federated mobile cloud computing system, crossing the smartphone end, the cloudlet end and the cloud end. Access control can take different forms depending on the specific access control needs at a specific level, so that enforcements at the multiple levels can collectively achieve the desired objectives.

At the cloudlet end and the cloud end, the access control module resembles those that typically protect today's cloud-based systems, such as Amazon Web Services (AWS) [75], which are often coupled with rich operating system and application-level controls running inside the cloud and cloudlet virtual machines. For example, access to confidential data may be guarded by operating system kernel and file system mechanisms, which may further rely on the support provided by the underlying hypervisor or virtual machine monitor. Access control in such platforms is relatively better understood because of the many years of research effort.

Unfortunately, access control for smartphones is nowhere near as mature or comprehensive, and thus will require this module to address a broad range of concerns if the overall architecture is to remain secure. Today's smartphones have evolved to an open platform that primarily employs discretionary access control to run third-party apps alongside system and platform applications. Thus, smartphones are extremely vulnerable to owners' poor administrative decisions as well as malicious apps that may contain malware, spyware, etc. While all major smartphone platforms are vulnerable in these ways, those based on the Android Open Source Project (AOSP)[2] have been extensively studied and will form the basis for the remainder of this discussion.

Despite the fact that Android apps are isolated as separate and independent users by the kernel, there are still many ways that two malicious apps can establish a covert channel to abuse data/resource that otherwise cannot by accessed by either one individually [51]. Moreover, some of these kinds of covert channels cannot be detected by existing information-flow tracking/detection tools [51]. Flawed apps may also configure their DAC improperly, resulting in information leakage and compromise, as was the case with the Skype app for Android [76]. In addition, because of the underlying inter-process communication (IPC) model that Android is built upon, known as Binder[3], vulnerabilities in any app's public interfaces, including

[2] `http://source.android.com/`

[3] `http://developer.android.com/reference/android/os/Binder.html`

system and platform apps/services, can be used by malicious apps to gain privileges not explicitly assigned [77].

The above problems with smartphone security are just the tip of the iceberg, yet many can be traced back to mistakes by owners' and/or developers' use of DAC. Even when employed properly, the fundamental shortcomings of the DAC-only model [78] create a situation that seriously undermines the security of the mobile end-point, and in turn, the security of the entire federated mobile cloud computing system. In a tightly-integrated mobile cloud architecture such as the one described here, these end-point vulnerabilities, if not addressed, can have serious ramifications for the entire mobile cloud architecture. Adding a flexible mandatory access control (MAC) framework to mobile devices is the first big step that should be taken to strengthen this weak link.

Fortunately, several large projects have recently made great strides in realizing effective MAC on Android devices. Security Enhancements (SE) for Android [79] is an Android-specific adaption of Security-Enhanced Linux (SELinux).[4] Just as with current versions of SELinux, SE for Android's enhanced kernel enforces, by way of a Linux Security Module (LSM) [80], a MAC policy that can mitigate many Android attacks that utilize the root user id (UID) or improperly secured access to shared resources such as the SD card. In SE for Android, the policy source is compiled into a binary monolithic policy file when the device image is built. All subjects and objects are labeled with a security context, which is used along with the policy's rules to grant or deny a certain subject's requested action on a particular object. In addition to the standard SELinux classes such as file, socket, etc., SE for Android introduces new classes to account for Android-specific features including `binder`, `zygote`, and `property_service`.

Besides the SELinux-like kernel MAC support, the SE for Android project includes several different types of middleware MAC (MMAC) mechanisms that would be appropriate for inclusion in this module.

- Install MAC: Based on its signature, an app's requested permissions are checked at boot time. If not specifically allowed by the policy, the app is not installed.
- Intent MAC (experimental): Intents sent from one app component to another, and between apps are denied unless specifically allowed by policy.
- Content Provider MAC (experimental): An app's access to ContentProviders is denied unless specifically allowed by policy.
- Revoke Permissions (experimental): An app's permissions may be revoked at run-time.

The kernel MAC and install MAC features of SE for Android have been mainstreamed into AOSP [81]. As of this writing, the other middleware features are still experimental.

Building on SE for Android, FlaskDroid [82] is a comprehensive, modular, flexible MAC implementation that addresses MAC features across the entire Android system. Although it uses SE for Android's SELinux kernel, FlaskDroid follows the

[4] `http://selinuxproject.org/page/SEAndroid`

Flask [83] security architecture to address the remainder of the Android system. The result is a viable MAC solution for the smartphone endpoint that maps directly onto this module within the federated mobile cloud computing architecture. Notably, FlaskDroid provides support for protecting the interests of multiple stakeholders in a distributed open information architecture such as ours. For federated mobile cloud computing, this is important because the architecture must consolidate the often-conflicting interests of the smartphone owner, cloudlet/cloud owner, and data owner. For example, FlaskDroid's policy language has provisions for stakeholder priority as described in [84], e.g., higher-ranked stakeholder policies override lower ones.

Another aspect of access control that is important for the endpoint enforcement is end-user identification and authentication. Typical smartphone single-factor authentication techniques, such as PIN, password, and pattern will not be adequate except for the most casual applications. Fortunately, strong two- and three-factor approaches which can be tightly coupled to the architecture's MAC policy are readily available.[85, 86, 87, 88] Moreover, flexibility can be achieved by allowing increased functionality when increasingly strong authentication points are satisfied.

**Security control center middleware service/module 6: Cryptographic key management infrastructure.**

In order to enable secure communication between the control centers and facilitate secure data sharing and offloading, there is a need for a cryptographic key management infrastructure whereby the communications can be protected with virtual (i.e., cryptography-protected) authenticated private channels. More specifically, this module will facilitate secure communications between the smartphones, between the smartphones and the cloudlets, between the smartphones and the clouds, between the cloudlets, and between the cloudlets and the clouds. In terms of the goals for secure communications, there are two types:

- Secure two-party communication: This allows two participants, which can be smartphones, cloudlets and cloud VMs, to conduct authenticated private communication so that no third participant can impersonate any of them and no third participant can decrypt the plaintext messages. This is often used to support pull-based secure data sharing described above.
- Secure group communication: This allows three or more participants, which can be smartphones, cloudlets and cloud VMs, to conduct authenticated private communication. This is also relevant, for example, in combat fields where the commander needs to send orders to a group of soldiers in real-time. This is often used to support push-based secure data sharing between a group of entities.

In terms of the means that can facilitate the above secure two-party and group communications, there are several kinds of cryptographic techniques:

- KDC-based solution: A Key Distribution Center (KDC) generates and distributes cryptographic keys for the participants, where the distribution of keys is conducted over appropriate authenticated private channels that can be instantiated

on cryptographic protocols. In principle, this solution can support both secure two-party communication, but it is perhaps more useful for supporting secure group communication because of its appealing performance in the key management procedure, especially when the group membership is dynamic (i.e., members may join or leave the group on demand). We refer to, for example, [89, 90, 91, 92] for provably secure mechanisms for secure stateful (where the revocation of group membership is explicitly conducted) and stateless (where the revocation of group membership is implicitly conducted) group communication schemes.

- PKI-based solution: Public Key Infrastructure (PKI) is an important service for supporting secure communications. PKI has been widely deployed, especially within the DoD enterprise. This infrastructure has its own management modules that take care of the issuance and revocation of public key certificates. This solution is especially suitable for generating a session key on demand. There are many provably secure cryptographic protocols for using PKI to generate session keys, such as [93, 94, 95] and for generating group communication keys, such as [96, 97].
- Identity-based solution: Identity-based cryptography is a variant of PKI because each participant's identity can serve as the participant's public key. Identities for such purpose can be email addresses. There is an authority that generates the corresponding private keys for the participants. This solution is therefore also appropriate for DoD-like enterprises because the employees' email addresses are assigned by some authority. We refer to [98, 99] and their numerous follow-on provably secure identity-based cryptosystems.
- Attribute-based solution: Attribute-based cryptography also can be seen as a variant of PKI. In particular, attribute-based cryptography is perhaps more appropriate for implementing mandatory access control and role-based access control because a participant is assigned with multiple attributes or roles, which can correspond to (for example) the clearance the participant holds. Moreover, attribute-based encryption is especially suitable for push-based information sharing, where the data owner can use attribute-based cryptosystem to encrypt the data and publishes the ciphertexts so that only the participants who possess the (private keys corresponding to the) desired set of attributes can decrypt the ciphertext. This data sharing paradigm does *not* require the data owner to know the identities of the recipients; whereas, the above three solutions would explicitly or implicitly require the data owner to know the identities of the recipients. We refer to [100, 101, 102] for some example attribute-based cryptosystems.

No single method would meet the needs of all applications. Our flexible security architecture allows the security architect to adopt the appropriate cryptographic key infrastructure to incorporate into the security architecture.

**Security control center middleware service/module 7: Attack identification, representation, and countermeasure.**

This module is responsible for identifying attacks, possibly based on the output from anti-malware tools, intrusion detection systems, and the monitors that are embedded into the hypervisor and/or OS. For example, we expect that VM introspection [103], which is to examine active VM states during execution time for security purposes, will become a widely used defense tool that can be deployed across the smartphones, cloudlets and cloud data centers because of the virtualization technology that is becoming available at all of these platforms (if not already). The identified attacks need to be represented in a fashion that will ease the sharing of the attack information. If possible, countermeasures against the newly identified attacks need to be generated, either manually or automatically.

There is a tension between the resource consumption and the security needs, especially because security mechanisms such as anti-malware tools may consume a great deal of battery on smartphones. Although it has been shown that checking integrity of kernel-code pages does not incur significant battery consumption, checking integrity of data structures in the kernel can consume a significant amount of battery [104]. For this reason, current anti-malware techniques that are deployed on desktop platforms would not be appropriate for deploying on smartphone platforms because scanning of smartphones with respect to a large malware signature database is prohibitively expensive in terms of battery consumption. Although the idea of offloading malware scanning to the cloud is promising, as exemplified by the ThinAV system [66] and the cloud-based anti-malware services such as `virustotal.com`, we need to cope with two issues: the privacy concerns and the battery-intense smartphone-to-cloud communications. Paranoid Android [67] attempts to address these trade-offs by hosting an exact replica of the smartphone in the cloud, and applying intensive security analyses and multiple detection techniques to the cloud replica, rather than locally on the smartphone itself.

Another method to alleviate the tension is by ranking the degree of suspicion of apps via the *multiple correspondence analysis*, by which really suspicious apps can be rendered to further more detailed and resource-consuming analysis [105]. The detection can be multiple times faster than the signature-based detection method. More studies are needed in order to rapidly detect new and zero-day attacks. This kind of agility is key to effective cyber defense.

Currently, the representation of attacks and generations of countermeasures are often manual. In order to be effective, we need to study automated generation of countermeasures. This is a promising research direction, and we have some initial ideas that we plan to explore in the near future.

**Automated Two-Tier OODA Loop of Security C2 and Control Centers.**

In the above we have described the function of the C2 centers and the Security Control Centers. Now we discuss how these centers seamlessly work together with each

other. As depicted in Figure 5, a feature of the security architecture is that it facilitates a novel fast/slow two-tier Observation-Orientation-Decision-Action (OODA) Loops.[5] The fast or agile OODA Loop corresponds to the observation, orientation, decision and action activities at each individual security control centers. Agility is achieved because the decisions and deployment of countermeasures are contained to a smartphone, a cloudlet, or a cloud data center. On the other hand, the slow OODA Loop corresponds to the observation, orientation, decision and action activities at a global scale. More specifically, the C2 centers not only analyze the data and information collected from the smartphones, cloudlets and data centers associated to the respective clouds in question, but also jointly work with each other to analyze the global situational awareness and decision-making.

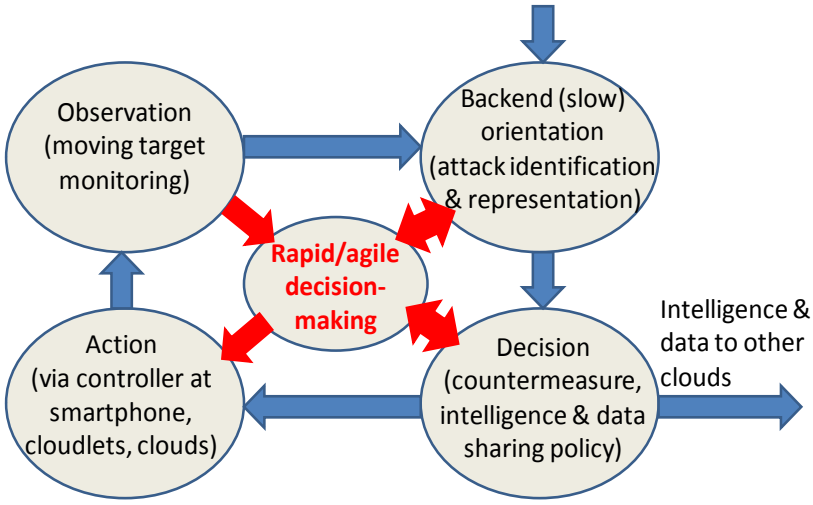


**Fig. 5** The fast/slow two-tier defense OODA Loop: The local security control center rapidly responds to detected attacks (the fast or agile OODA Loop), and the C2 center conducts global and comprehensive analysis that corresponds to the slow OODA Loop.

## *3.4 Putting the Pieces Together: Running Examples*

Now we present two examples by which we show how the security architecture can incorporate the various security services together to accomplish some tasks or missions, including some of the killer applications mentioned above.

[5] For background information about OODA Loop, we refer to `http://en.wikipedia.org/wiki/OODA_loop`.

**Security-enhanced situational authentication.**

Situational authentication is a defense mechanism that can mitigate the damage when the cryptographic authentication mechanisms have been compromised. Imagine that in a battlefield a soldier needs some information for deciding which direction to go. The information may be obtained from the data center back in the cloud and/or from the other soldiers in the same battlefield. One concern is that the information may have been compromised, either in the cloud or maliciously entered into the system by a compromised mobile device. Therefore, it is important to evaluate the trustworthiness of the information item. Several approaches can suffice this purpose. In the case where there is no reputation service, information trustworthiness can be evaluated based on the provenance of the information item, that is tightly coupled with cryptographic mechanisms [106, 107, 64]. Note that the cryptographic attestation of data provenance can be viable, despite that some of the mobile devices or the cloud might have been compromised, because the cryptographic keys for attesting the provenance information may be out of these systems and are not compromised thereof. Moreover, situational authentication can further enhance the trustworthiness of the data item because the particular data item is attested by multiple sources that can be authenticated by the information recipient.

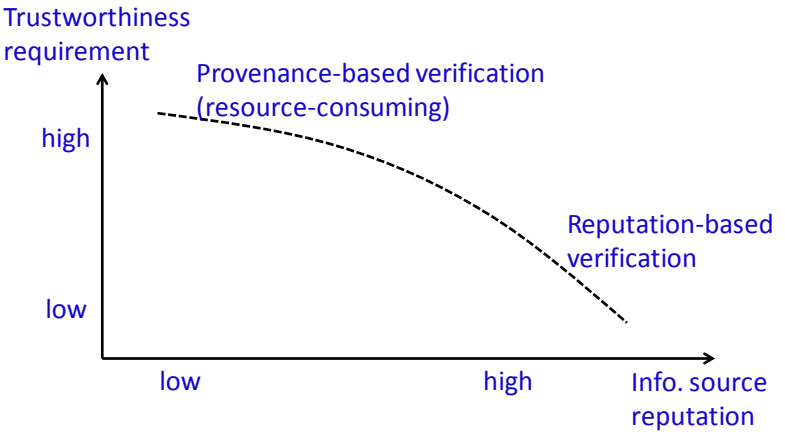


**Fig. 6** Information trustworthiness management service for achieving better trade-off between the efficiency of provenance verification and the trustworthiness of information.

However, verifying provenance information can be resource consuming, especially when the provenance information is associated to multiple users who have processed the information item. Therefore, we need some alternatives to achieve good trade-off between information trustworthiness verification and efficiency. The first alternate solution is to offload the provenance verification operations to the clouds or the cloudlets. This solution has the drawback that if the cloudlet in ques-

tion is compromised, the provenance verification operation is compromised. Unless there are multiple cloudlets and the smartphone-cloudlets communications do not consume more battery than the computation at the smartphone, we would not recommend this solution. The second alternate solution is to resort to some reputation system, meaning that information received from more reputed sources can be accepted as more trustworthy than the information received from the less reputed sources. In general, the degree of trustworthiness of received information may be some appropriate function of the reputation(s) of the information source(s). As illustrated in Figure 6, to achieve a certain level of trustworthiness, full-fledged verification of provenance information may be imposed against low-reputation sources (meaning consumption of more computational resources), and selected verification of provenance information may be conducted against high-reputation sources (meaning consumption of lesser computational resources).

**Secure rapid information/intelligence sharing.**

When a new attack is detected at a smartphone, cloudlet or cloud data center, information about the attack needs to be shared between the security control centers and further the C2 centers. The sharing of such information requires authenticated private channels, by which the information recipients can verify the information/intelligence in question indeed comes from a trusted party. This also makes the provenance information about the information/intelligence source non-repudiable. These needs can be easily supported by the cryptographic key management infrastructure.

The tiered federated mobile cloud computing architecture also supports the notion of having information available only when it's most needed and relevant. For example, information about "last mile" security events can be maintained and shared in real-time at/among the smartphone and/or cloudlet levels, while only being present in the cloud for archival purposes. Transient information that is only relevant momentarily or for particular smartphone locations is confined to the mobile endpoint devices and securely shared directly among them.

## 4 Related Work

We refer to the following survey for more information about mobile cloud computing [108, 42, 109, 5, 110]. For example, [109] presents a taxonomy of mobile cloud computing based on operational issues (e.g., offloading tasks to clouds), end-user issues (e.g., incentivizing users to contribute their smartphones to formulate a cloud computing substrate), service/application issues (e.g., measuring service performance), security issues (e.g., attacks against data stored in the clouds), context issues (e.g., context-based service/resource provisioning), and data management issues (e.g., data interoperability and portability). On the other hand, [110] discusses

the security services at different layers. Specifically, they discussed the following: secure cloud applications would require privacy protection, authentication, and data integrity verification, etc; secure cloud process hosting services would require VM secure migration, VM isolation etc. However, these prior works do not present any systematic security architecture for securing the user-end of (federated) mobile cloud computing. To the best of our knowledge, we are the first to systematically investigate and design security architecture for federated mobile cloud computing. Nevertheless, there have been some studies on security architecture for the cloud-end and certainly many studies on security mechanisms, as we will review below.

**Related prior studies of cloud security architectures.**

First, a security architecture is presented in [31]. This security architecture focuses on the securing the cloud end or cloud data centers, by systematically integrating defense mechanisms crossing multiple layers — application, network, and system. The security architecture proposed in the present paper deals with the mobile devices, the cloudlets and the cloud data centers, and therefore can be seamlessly integrated with the security architecture proposed in [31]. Indeed, the concept of C2 centers was introduced in [31], and further plays an important role in the security architecture of the present paper.

Second, the twin-cloud security architecture [111] targets secure outsourcing of data and arbitrary computations to an untrusted commodity cloud. The idea is to let a user communicate with a trusted cloud, which serves as a proxy of the user by verifying the data stored in, and the operation results returned by, the untrusted commodity cloud. It can further split security-critical operations from the others so that the security-critical operations are conducted in the trusted cloud.

Third, the cloud security architecture presented in [32] aims to achieve the following security objectives simultaneously: improving resilience of system components that are critical to the current applications in question; learn to adapt to the evolving threats; enforce moving-target defense. Its components include the following: a substrate for distributed monitoring and cross-checking at multiple levels of abstraction to differentiate normal from anomalous behaviors; artificial mechanisms for diversifying the instruction sets; a fine-grained information tracking system; a deception mechanism for generating, injecting and tracking information; fast process migration.

Fourth, the “Mobile Security Reference Architecture” published by the Federal CIO Council in May 2013 [41] focuses on the security of using commodity mobile devices and infrastructures that are used to access Federal Government resources. Mobile devices include smartphones and tablet computers (rather than laptops). It does not fully address security of mobile devices and supporting networks that are used to access services for National Security and Emergency Preparedness needs. We hope that the present investigation and the security architecture can be adopted by the federal government as a starting point of the ultimate reference security architecture for federated mobile cloud computing.

Fifty, the "NIST Cloud Computing Security Reference Architecture" published in May 2013 [33] is for the cloud-end security. A key feature of the architecture is the concept of "broker" which achieves the following: secure service aggregation — supporting the integration and fusion of multiple services into some new services; secure service arbitrage — choosing the desired services from multiple candidate providers; secure service intermediation — enhancing services via, for example, managing access to cloud services, identity management, enhanced security; secure cloud ecosystem orchestration.

**Related prior studies of security mechanisms.**

At the mechanisms level, there have been many studies, such as: cloud-based detection of malware in mobile devices [112]; context-sensitive access control [113, 114, 115, 116]; secure logging system for mobile devices [117] via elliptic curve cryptography; secure storage and retrieval of encrypted data outsourced to the cloud [118] while delegating the heavy encryption and decryption operations to the cloud (but assuming the cloud will conduct the operations faithfully). Our security architecture is flexible enough to accommodate such mechanisms and the others discussed in the main body of the text.

# 5 Conclusion

## *5.1 Summary*

We have systematically examined the threat model against, and security requirements of, federated mobile clouds computing. In particular, we focused on the unique features of federated mobile cloud computing and the unique security needs thereof. In particular, we are the first to put forth that mobile cloud computing can aim to achieve the "$1 + 1 > 2$" effect from the perspective of functions (i.e., mobile cloud computing can achieve functions that are infeasible or even impossible to achieve by mobile computing or cloud computing alone). Despite that mobile cloud computing has an enlarged vulnerability surface than mobile computing and cloud computing, respectively, it is important to achieve the "$1 + 1 > 2$" effect in terms of defense, meaning the exploitation of cloud computing to help secure mobile computing and *vice versa*. We also highlighted that private clouds are not necessarily more secure than public clouds.

We have presented a systematic security architecture for federated mobile cloud computing, where the cloud-end can be public clouds, private clouds, hybrid clouds, or social clouds. To the best of our knowledge, this is the first security architecture for federated mobile cloud computing. The proposed security architecture supports a three-tiered system structure of federated mobile cloud computing: the mobile de-

vices, the cloudlets, and the clouds. Correspondingly, the proposed security architecture can be implemented as a middleware that can be deployed at the mobile devices, the cloudlets, and the clouds. The proposed security architecture can accommodate a diverse set of security services. We not only discuss the security services of the security architecture components, but also explore approaches to fulfilling the security services. For example, we propose the novel concept of *situational authentication*, which is different from existing authentication methods based on "what you know" (e.g., password or cryptographic private key), "what you have" (e.g., a hardware token), "who you are" (e.g., fingerprints and biometrics). Instead, situational authentication is based on "whom you are with" (e.g., the people surrounding you) or "where you are" (e.g., your location or elements of your environment), and can provide another layer of assurance when a mobile device (including the associated password or any other authentication method) is under the adversary control.

## *5.2 Extensibility of the Architecture*

Our security architecture primarily focuses on the security aspect of federated mobile cloud computing. For example, the proposed security architecture can be naturally and seamlessly extended to incorporate the security architecture for securing the cloud end [31], which include interesting services at the application, network and system layers. It is possible to extend our security architecture to accommodate other cloud-end security architectures as well. In what follows we discuss the possibly extensions to accommodate the services that are orthogonal, but relevant, to the security aspect, which is the focus of the present chapter.

The first extension would be to accommodate the *incentivization* aspect. Specifically, we assumed that the clouds have the incentive to formulate a federation, which is natural for applications such as DoD clouds. For other scenarios, we may need to design mechanisms to incentivize the formulation of federations. Moreover, we also need to incentivize the smartphone users for participating in mobile cloud computing (see, e.g., [119, 120, 63]).

The second extension would be to accommodate the *privacy* aspect, namely the privacy of the mobile users that have contributed to the participatory or opportunistic sensing. This is relevant not only in civilian applications setting [121], but also in military applications because knowing which solider has contributed what sensing information in the past (i.e., mobility history) could allow one to infer useful information.

The third extension would be to accommodate the *reputation* aspect, which is related to the trustworthiness of the data contributed by the users. (Recall that we discussed how reputation may be exploited to reduce the verification cost of provenance data, which is nevertheless orthogonal to the extension mentioned here.) This is especially relevant for participatory and opportunistic applications because the users may have the incentives to abuse the functions. This aspect is often coupled

with the privacy aspect because we need to protect both the reputation and the privacy of the users [122].

**Acknowledgements** This material is based upon work supported in part by Air Force Research Laboratory under contract number FA8750-?? and National Science Foundation under Grant No. 1111925. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the author(s) and do not necessarily reflect the views of the funding agency.

## References

1. M. Conti and M. Kumar, "Opportunities in opportunistic computing," *Computer*, vol. 43, pp. 42–50, Jan. 2010.
2. N. D. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A. T. Campbell, "A survey of mobile phone sensing," *Comm. Mag.*, vol. 48, no. 9, pp. 140–150, 2010.
3. P. Mell and T. Grance, "The NIST definition of cloud computing." `http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf`, September 2011.
4. Defense Science Board, "Cyber security and reliability in a digital cloud." `http://www.acq.osd.mil/dsb/reports/CyberCloud.pdf`, Jan. 2013.
5. F. Liu, P. Shu, H. Jin, L. Ding, J. Yu, D. Niu, and B. Li, "Gearing resource-poor mobile devices with powerful clouds: architectures, challenges, and applications," *IEEE Wireless Commun.*, vol. 20, no. 3, pp. 1–0, 2013.
6. S. Kosta, A. Aucinas, P. Hui, R. Mortier, and X. Zhang, "Thinkair: Dynamic resource allocation and parallel execution in the cloud for mobile code offloading," in *Proceedings of the IEEE INFOCOM 2012, Orlando, FL, USA, March 25-30, 2012* (A. G. Greenberg and K. Sohraby, eds.), pp. 945–953, IEEE, 2012.
7. E. Cuervo, A. Balasubramanian, D.-k. Cho, A. Wolman, S. Saroiu, R. Chandra, and P. Bahl, "Maui: making smartphones last longer with code offload," in *Proceedings of the 8th international conference on Mobile systems, applications, and services*, MobiSys '10, (New York, NY, USA), pp. 49–62, ACM, 2010.
8. B.-G. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti, "Clonecloud: elastic execution between mobile device and cloud," in *Proceedings of the sixth conference on Computer systems*, EuroSys '11, (New York, NY, USA), pp. 301–314, ACM, 2011.
9. M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The case for vm-based cloudlets in mobile computing," *IEEE Pervasive Computing*, vol. 8, pp. 14–23, Oct. 2009.
10. X. Zhang, J. Schiffman, S. Gibbs, A. Kunjithapatham, and S. Jeong, "Securing elastic applications on mobile devices for cloud computing," in *Proceedings of the 2009 ACM workshop on Cloud computing security*, CCSW '09, pp. 127–134, 2009.
11. X. Zhang, A. Kunjithapatham, S. Jeong, and S. Gibbs, "Towards an elastic application model for augmenting the computing capabilities of mobile devices with cloud computing," *Mob. Netw. Appl.*, vol. 16, pp. 270–284, June 2011.
12. E. Miluzzo, R. Cáceres, and Y.-F. Chen, "Vision: mclouds - computing on clouds of mobile devices," in *Proceedings of the third ACM workshop on Mobile cloud computing and services (MCS'12)*, (New York, NY, USA), pp. 9–14, ACM, 2012.
13. S. Čapkun, J.-P. Hubaux, and L. Buttyán, "Mobility helps security in ad hoc networks," in *Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*, MobiHoc '03, (New York, NY, USA), pp. 46–56, ACM, 2003.
14. S. Xu and S. Capkun, "Distributed and secure bootstrapping of mobile ad hoc networks: Framework and constructions," *ACM Trans. Inf. Syst. Secur.*, vol. 12, no. 1, 2008.
15. A. Juels and B. S. Kaliski, Jr., "Pors: proofs of retrievability for large files," in *Proceedings of the 14th ACM conference on Computer and communications security*, CCS '07, (New York, NY, USA), pp. 584–597, ACM, 2007.

16. G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable data possession at untrusted stores," in *Proceedings of the 14th ACM conference on Computer and communications security*, CCS '07, (New York, NY, USA), pp. 598–609, ACM, 2007.
17. Q. Zheng and S. Xu, "Fair and dynamic proofs of retrievability," in *First ACM Conference on Data and Application Security and Privacy (CODASPY 2011) San Antonio, TX, USA, February 21-23, 2011, Proceedings* (R. S. Sandhu and E. Bertino, eds.), pp. 237–248, ACM, 2011.
18. Q. Zheng and S. Xu, "Secure and efficient proof of storage with deduplication," in *Second ACM Conference on Data and Application Security and Privacy (CODASPY 2012), San Antonio, TX, USA, February 7-9, 2012* (E. Bertino and R. S. Sandhu, eds.), pp. 1–12, ACM, 2012.
19. Q. Zheng, S. Xu, and G. Ateniese, "Efficient query integrity for outsourced dynamic databases," in *Proceedings of the 2012 ACM Workshop on Cloud computing security (CCSW 2012), Raleigh, NC, USA, October 19, 2012* (T. Yu, S. Capkun, and S. Kamara, eds.), pp. 71–82, ACM, 2012.
20. Q. Zheng, S. Xu, and G. Ateniese, "Verifiable attribute-based keyword search over outsourced encrypted data." Cryptology ePrint Archive, Report 2013/462, 2013. `http://eprint.iacr.org/`.
21. T. Wang, G. Cardone, A. Corradi, L. Torresani, and A. T. Campbell, "Walksafe: a pedestrian safety app for mobile phone users who walk and talk while crossing roads," in *Proceedings of the Twelfth Workshop on Mobile Computing Systems and Applications (HotMobile'12)*, pp. 5:1–5:6, 2012.
22. Health News, "Nearly half of high schoolers text while driving: Survey." `http://www.nlm.nih.gov/medlineplus/news/fullstory_136763.html`, May 13, 2013.
23. D. Huang, Z. Zhou, L. Xu, T. Xing, and Y. Zhong, "Secure data processing framework for mobile cloud computing," in *Computer Communications Workshops (INFOCOM WKSHPS), 2011 IEEE Conference on*, pp. 614–618, 2011.
24. M. Satyanarayanan, "Mobile computing: the next decade," in *Proceedings of the 1st ACM Workshop on Mobile Cloud Computing and Services: Social Networks and Beyond (MCS'10)*, (New York, NY, USA), pp. 5:1–5:6, ACM, 2010.
25. M. Rostami, W. Burleson, F. Koushanfar, and A. Juels, "Balancing security and utility in medical devices?," in *Proceedings of the 50th Annual Design Automation Conference*, DAC '13, (New York, NY, USA), pp. 13:1–13:6, ACM, 2013.
26. S. Gollakota, H. Hassanieh, B. Ransford, D. Katabi, and K. Fu, "They can hear your heartbeats: non-invasive security for implantable medical devices," in *Proceedings of the ACM SIGCOMM 2011 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, Toronto, ON, Canada, August 15-19, 2011* (S. Keshav, J. Liebeherr, J. W. Byers, and J. C. Mogul, eds.), pp. 2–13, ACM, 2011.
27. S. S. Clark and K. Fu, "Recent results in computer security for medical devices," in *Wireless Mobile Communication and Healthcare - Second International ICST Conference, MobiHealth 2011, Kos Island, Greece, October 5-7, 2011. Revised Selected Papers* (K. S. Nikita, J. C. Lin, D. I. Fotiadis, and M. T. A. Waldmeyer, eds.), vol. 83 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pp. 111–118, Springer, 2011.
28. T. Verbelen, P. Simoens, F. De Turck, and B. Dhoedt, "Cloudlets: bringing the cloud to the mobile user," in *Proceedings of the third ACM workshop on Mobile cloud computing and services*, MCS'12, (New York, NY, USA), pp. 29–36, ACM, 2012.
29. J. Li, K. Bu, X. Liu, and B. Xiao, "Enda: embracing network inconsistency for dynamic application offloading in mobile cloud computing," in *Proceedings of the second ACM SIGCOMM workshop on Mobile cloud computing*, MCC '13, (New York, NY, USA), pp. 39–44, ACM, 2013.
30. P. Bahl, R. Y. Han, L. E. Li, and M. Satyanarayanan, "Advancing the state of mobile cloud computing," in *Proceedings of the third ACM workshop on Mobile cloud computing and services*, MCS '12, pp. 21–28, 2012.

31. W. Luo, L. Xu, Z. Zhan, Q. Zheng, and S. Xu, "Federated cloud security architecture for secure and agile clouds," 2013.
32. A. D. Keromytis, R. Geambasu, S. Sethumadhavan, S. J. Stolfo, J. Yang, A. Benameur, M. Dacier, M. Elder, D. Kienzle, and A. Stavrou, "The meerkats cloud security architecture," in *Proceedings of the 32nd International Conference on Distributed Computing Systems Workshops (ICDCSW'12)*, pp. 446–450, 2012.
33. NIST Cloud Computing Security Working Group, "NIST cloud computing security reference architecture (NIST special publication 500-299)." `http://collaborate.nist.gov/twiki-cloud-computing/pub/CloudComputing/CloudSecurity/NIST_Security_Reference_Architecture_2013.05.15_v1.0.pdf`, 2013.
34. S. Xu and M. Yung, "Socialclouds: Concept, security architecture and some mechanisms," in *First International Conference on Trusted Systems (INTRUST)*, pp. 104–128, 2009.
35. E. J. Kartaltepe, J. A. Morales, S. Xu, and R. Sandhu, "Social network-based botnet command-and-control: emerging threats and countermeasures," in *Proceedings of the 8th international conference on Applied cryptography and network security*, ACNS'10, (Berlin, Heidelberg), pp. 511–528, Springer-Verlag, 2010.
36. "Zeus botnet finds hold in amazon cloud." `http://www.securityfocus.com/brief/1046`, 2009.
37. Z. Wang and X. Jiang, "Hypersafe: A lightweight approach to provide lifetime hypervisor control-flow integrity," in *Proceedings of the 2010 IEEE Symposium on Security and Privacy*, SP '10, (Washington, DC, USA), pp. 380–395, IEEE Computer Society, 2010.
38. A. M. Azab, P. Ning, Z. Wang, X. Jiang, X. Zhang, and N. C. Skalsky, "Hypersentry: enabling stealthy in-context measurement of hypervisor integrity," in *Proceedings of the 17th ACM conference on Computer and communications security*, CCS '10, (New York, NY, USA), pp. 38–49, ACM, 2010.
39. U. Steinberg and B. Kauer, "Nova: a microhypervisor-based secure virtualization architecture," in *Proceedings of the 5th European conference on Computer systems*, EuroSys '10, (New York, NY, USA), pp. 209–222, ACM, 2010.
40. J. Szefer, E. Keller, R. B. Lee, and J. Rexford, "Eliminating the hypervisor attack surface for a more secure cloud," in *Proceedings of the 18th ACM conference on Computer and communications security*, CCS '11, (New York, NY, USA), pp. 401–412, ACM, 2011.
41. Federal CIO Council, "Mobile security reference architecture." `https://cio.gov/wp-content/uploads/downloads/2013/05/Mobile-Security-Reference-Architecture.pdf`, May 23, 2013.
42. H. T. Dinh, C. Lee, D. Niyato, and P. Wang, "A survey of mobile cloud computing: architecture, applications, and approaches," *Wireless Communications and Mobile Computing*, pp. n/a–n/a, 2011.
43. A. Gonsalves, "Android lock screen bypass highlights mobile risk," *CSO Online*, July 2013.
44. B. Lau, "Mactans: Injecting malware into iOS devices via malicious chargers," in *BlackHat USA*, UBM Tech, 2013.
45. A. P. Felt, M. Finifter, E. Chin, S. Hanna, and D. Wagner, "A survey of mobile malware in the wild," in *Proceedings of the 1st ACM workshop on Security and privacy in smartphones and mobile devices*, SPSM '11, (New York, NY, USA), pp. 3–14, ACM, 2011.
46. Y. Zhou, Z. Wang, W. Zhou, and X. Jiang, "Hey, you, get off of my market: Detecting malicious apps in official and alternative android markets," in *Proceedings of the 19th Network and Distributed System Security Symposium (NDSS 2012), San Diego, CA, February 2012*, 2012.
47. Y. Zhou and X. Jiang, "Dissecting android malware: Characterization and evolution," in *IEEE Symposium on Security and Privacy, SP 2012, 21-23 May 2012, San Francisco, California, USA*, pp. 95–109, IEEE Computer Society, 2012.
48. N. Hardy, "The confused deputy: (or why capabilities might have been invented)," *SIGOPS Oper. Syst. Rev.*, vol. 22, pp. 36–38, Oct. 1988.
49. L. Davi, A. Dmitrienko, A.-R. Sadeghi, and M. Winandy, "Privilege escalation attacks on android," in *Proceedings of the 13th international conference on Information security*, ISC'10, (Berlin, Heidelberg), pp. 346–360, Springer-Verlag, 2011.

50. R. Schlegel, K. Zhang, X. yong Zhou, M. Intwala, A. Kapadia, and X. Wang, "Soundcomber: A stealthy and context-aware sound trojan for smartphones.," in *NDSS*, The Internet Society, 2011.
51. C. Marforio, H. Ritzdorf, A. Francillon, and S. Capkun, "Analysis of the communication between colluding applications on modern smartphones," in *Proceedings of the 28th Annual Computer Security Applications Conference*, ACSAC'12, pp. 51–60, 2012.
52. C. Marforio, A. Francillon, and S. Čapkun, "Application collusion attack on the permission-based security model and its implications for modern smartphone systems," Tech. Rep. 724, ETH Zurich, April 2011.
53. S. Bugiel, L. Davi, A. Dmitrienko, T. Fischer, A.-R. Sadeghi, and B. Shastry, "Towards taming privilege-escalation attacks on Android," in *19th Annual Network & Distributed System Security Symposium (NDSS'12)*, Feb 2012.
54. X. Ma, P. Huang, X. Jin, P. Wang, S. Park, D. Shen, Y. Zhou, L. K. Saul, and G. M. Voelker, "edoctor: automatically diagnosing abnormal battery drain issues on smartphones," in *Proceedings of the 10th USENIX conference on Networked Systems Design and Implementation (NSDI'13)*, (Berkeley, CA, USA), pp. 57–70, USENIX Association, 2013.
55. W. Zhou, Y. Zhou, X. Jiang, and P. Ning, "Detecting repackaged smartphone applications in third-party android marketplaces," in *Proceedings of the second ACM conference on Data and Application Security and Privacy*, CODASPY '12, (New York, NY, USA), pp. 317–326, ACM, 2012.
56. W. Enck, P. Gilbert, B.-G. Chun, L. P. Cox, J. Jung, P. McDaniel, and A. N. Sheth, "Taintdroid: an information-flow tracking system for realtime privacy monitoring on smartphones," in *Proceedings of the 9th USENIX conference on Operating systems design and implementation*, OSDI'10, (Berkeley, CA, USA), pp. 1–6, USENIX Association, 2010.
57. P. Hornyack, S. Han, J. Jung, S. E. Schechter, and D. Wetherall, "These aren't the droids you're looking for: retrofitting android to protect data from imperious applications," in *Proceedings of the 18th ACM Conference on Computer and Communications Security, CCS 2011, Chicago, Illinois, USA, October 17-21, 2011* (Y. Chen, G. Danezis, and V. Shmatikov, eds.), pp. 639–652, ACM, 2011.
58. W. Luo, S. Xu, and X. Jiang, "Real-time detection and prevention of android sms permission abuses," in *Proceedings of the first international workshop on Security in embedded systems and smartphones*, SESP '13, (New York, NY, USA), pp. 11–18, ACM, 2013.
59. I. T. Fischer, C. Kuo, L. Huang, and M. Frank, "smartphones: not smart enough?," in *Proceedings of the second ACM workshop on Security and privacy in smartphones and mobile devices*, SPSM '12, (New York, NY, USA), pp. 27–32, ACM, 2012.
60. O. Riva, C. Qin, K. Strauss, and D. Lymberopoulos, "Progressive authentication: deciding when to authenticate on mobile phones," in *Proceedings of the 21st USENIX conference on Security symposium*, (Berkeley, CA, USA), pp. 15–15, USENIX Association, 2012.
61. J. Oberheide and F. Jahanian, "When mobile is harder than fixed (and vice versa): demystifying security challenges in mobile environments," in *Proceedings of the Eleventh Workshop on Mobile Computing Systems & Applications*, HotMobile '10, pp. 43–48, 2010.
62. J. Oberheide, K. Veeraraghavan, E. Cooke, J. Flinn, and F. Jahanian, "Virtualized in-cloud security services for mobile devices," in *Proceedings of the First Workshop on Virtualization in Mobile Computing (MobiVirt'08)*, (New York, NY, USA), pp. 31–35, ACM, 2008.
63. X. Sheng, J. Tang, X. Xiao, and G. Xue, "Sensing as a service: Challenges, solutions and future directions," *Sensors Journal, IEEE*, vol. 13, no. 10, pp. 3733–3741, 2013.
64. W. Dai, T. P. Parker, H. Jin, and S. Xu, "Enhancing data trustworthiness via assured digital signing," *IEEE Trans. Dependable Sec. Comput.*, vol. 9, no. 6, pp. 838–851, 2012.
65. S. Xu, X. Li, T. P. Parker, and X. Wang, "Exploiting trust-based social networks for distributed protection of sensitive data," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 1, pp. 39–52, 2011.
66. C. Jarabek, D. Barrera, and J. Aycock, "Thinav: truly lightweight mobile cloud-based anti-malware," in *Proceedings of the 28th Annual Computer Security Applications Conference (ACSAC'12)*, (New York, NY, USA), pp. 209–218, ACM, 2012.

67. G. Portokalidis, P. Homburg, K. Anagnostakis, and H. Bos, "Paranoid Android: versatile protection for smartphones," in *Proceedings of the* $26^{th}$ *Annual Computer Security Applications Conference*, ACSAC '10, (New York, NY, USA), pp. 347–356, ACM, 2010.
68. D. Boneh, X. Ding, and G. Tsudik, "Fine-grained control of security capabilities," *ACM Trans. Internet Techn.*, vol. 4, no. 1, pp. 60–82, 2004.
69. Y. Dodis, J. Katz, S. Xu, and M. Yung, "Key-insulated public key cryptosystems," in *Advances in Cryptology - EUROCRYPT 2002, International Conference on the Theory and Applications of Cryptographic Techniques, Amsterdam, The Netherlands, April 28 - May 2, 2002, Proceedings* (L. R. Knudsen, ed.), vol. 2332 of *Lecture Notes in Computer Science*, pp. 65–82, Springer, 2002.
70. Y. Dodis, W. Luo, S. Xu, and M. Yung, "Key-insulated symmetric key cryptography and mitigating attacks against cryptographic cloud software," in *7th ACM Symposium on Information, Compuer and Communications Security, ASIACCS '12, Seoul, Korea, May 2-4, 2012* (H. Y. Youm and Y. Won, eds.), pp. 57–58, ACM, 2012.
71. N. Santos, R. Rodrigues, K. P. Gummadi, and S. Saroiu, "Policy-sealed data: a new abstraction for building trusted cloud services," in *Proceedings of the 21st USENIX conference on Security symposium*, Security'12, (Berkeley, CA, USA), pp. 10–10, USENIX Association, 2012.
72. Y. Zhu, D. Ma, D. Huang, and C. Hu, "Enabling secure location-based services in mobile cloud computing," in *Proceedings of the second ACM SIGCOMM workshop on Mobile cloud computing*, MCC '13, (New York, NY, USA), pp. 27–32, ACM, 2013.
73. Y. Agarwal and M. Hall, "Protectmyprivacy: detecting and mitigating privacy leaks on ios devices using crowdsourcing," in *Proceeding of the 11th annual international conference on Mobile systems, applications, and services*, MobiSys '13, (New York, NY, USA), pp. 97–110, ACM, 2013.
74. Y. Nadji, J. Giffin, and P. Traynor, "Automated remote repair for mobile malware," in *Proceedings of the 27th Annual Computer Security Applications Conference (ACSAC'11)*, (New York, NY, USA), pp. 413–422, ACM, 2011.
75. Amazon.com, "Amazon web services: Overview of security processes." `http://media.amazonwebservices.com/pdf/AWS_Security_Whitepaper.pdf`, June 2013.
76. S. Hollister, "Skype for android vulnerable to hack that compromises personal info," *engadget*, April 14, 2011.
77. A. P. Felt, H. J. Wang, A. Moshchuk, S. Hanna, and E. Chin, "Permission re-delegation: attacks and defenses," in *Proceedings of the* $20^{th}$ *USENIX conference on Security*, SEC'11, (Berkeley, CA, USA), pp. 22–22, USENIX Association, 2011.
78. P. A. Loscocco, S. D. Smalley, P. A. Muckelbauer, R. C. Taylor, S. J. Turner, and J. F. Farrell, "The inevitability of failure: The flawed assumption of security in modern computing environments," in *In Proceedings of the* $21^{st}$ *National Information Systems Security Conference*, pp. 303–314, 1998.
79. S. Smalley and R. Craig, "Security enhanced (SE) android: Bringing flexible MAC to Android," in *Proc. 20th Annual Network & Distributed System Security Symposium*, 2013.
80. S. Smalley, C. Vance, and W. Salamon, "Implementing SELinux as a Linux Security Module," Feb 2006.
81. Android Open Source Project, "Security-enhanced Linux." `http://source.android.com/devices/tech/security/se-linux.html`.
82. S. Bugiel, S. Heuser, and A.-R. Sadeghi, "Flexible and fine-grained mandatory access control on Android for diverse security and privacy policies," in *22nd USENIX Security Symposium (USENIX Security'13)*, USENIX, 2013.
83. R. Spencer, S. Smalley, P. Loscocco, M. Hibler, D. Andersen, and J. Lepreau, "The flask security architecture: System support for diverse security policies," in *Proceedings of the* $8^{th}$ *USENIX Security Symposium*, pp. 123–129, Aug 1999.
84. V. Rao and T. Jaeger, "Dynamic mandatory access control for multiple stakeholders," in *Proceedings of the* $14^{th}$ *ACM symposium on Access control models and technologies*, SACMAT '09, (New York, NY, USA), pp. 53–62, ACM, 2009.

85. S. Shankland, "Using NFC, IBM brings dual-factor authentication to mobile," *c|net*, October 18, 2013.
86. G. Intelligence, "Two-factor authentication goes mobile," tech. rep., London, UK, September 2012.
87. J.-Y. Hu, C.-C. Sueng, W.-H. Liao, and C. Ho, "Android-based mobile payment service protected by 3-factor authentication and virtual private ad hoc networking," in *Computing, Communications and Applications Conference (ComComAp), 2012*, pp. 111–116, 2012.
88. F. Corella, "Convenient one-, two- and three-factor authentication for mobile devices," July 30, 2012.
89. C. K. Wong, M. Gouda, and S. S. Lam, "Secure group communications using key graphs," *IEEE/ACM Trans. Netw.*, vol. 8, pp. 16–30, Feb. 2000.
90. D. Wallner, E. Harder, and R. Agee, "Key management for multicast: Issues and architectures," 1999.
91. D. Naor, M. Naor, and J. B. Lotspiech, "Revocation and tracing schemes for stateless receivers," in *Proceedings of the 21st Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '01, (London, UK, UK), pp. 41–62, Springer-Verlag, 2001.
92. S. Xu, "On the security of group communication schemes," *Journal of Computer Security*, vol. 15, no. 1, pp. 129–169, 2007.
93. H. Krawczyk, K. G. Paterson, and H. Wee, "On the security of the tls protocol: A systematic analysis," *IACR Cryptology ePrint Archive*, vol. 2013, p. 339, 2013.
94. H. Krawczyk, "Cryptographic extraction and key derivation: The hkdf scheme," in *CRYPTO*, vol. 6223 of *Lecture Notes in Computer Science*, pp. 631–648, Springer, 2010.
95. H. Krawczyk, "Sigma: The 'sign-and-mac' approach to authenticated diffie-hellman and its use in the ike-protocols," in *CRYPTO*, vol. 2729 of *Lecture Notes in Computer Science*, pp. 400–425, Springer, 2003.
96. S. Jarecki, J. Kim, and G. Tsudik, "Flexible robust group key agreement," *IEEE Trans. Parallel Distrib. Syst.*, vol. 22, no. 5, pp. 879–886, 2011.
97. R. Gennaro, S. Jarecki, H. Krawczyk, and T. Rabin, "Secure distributed key generation for discrete-log based cryptosystems," *J. Cryptology*, vol. 20, no. 1, pp. 51–83, 2007.
98. A. Shamir, "Identity-based cryptosystems and signature schemes," in *CRYPTO*, vol. 196 of *Lecture Notes in Computer Science*, pp. 47–53, Springer, 1984.
99. D. Boneh and M. K. Franklin, "Identity-based encryption from the weil pairing," *SIAM J. Comput.*, vol. 32, no. 3, pp. 586–615, 2003.
100. S. Hohenberger and B. Waters, "Attribute-based encryption with fast decryption," in *Public Key Cryptography*, vol. 7778 of *Lecture Notes in Computer Science*, pp. 162–179, Springer, 2013.
101. A. B. Lewko and B. Waters, "New proof methods for attribute-based encryption: Achieving full security through selective techniques," in *CRYPTO*, vol. 7417 of *Lecture Notes in Computer Science*, pp. 180–198, Springer, 2012.
102. M. Pirretti, P. Traynor, P. McDaniel, and B. Waters, "Secure attribute-based systems," *Journal of Computer Security*, vol. 18, no. 5, pp. 799–837, 2010.
103. T. Garfinkel and M. Rosenblum, "A virtual machine introspection based architecture for intrusion detection," in *In Proc. Network and Distributed Systems Security Symposium*, pp. 191–206, 2003.
104. J. Bickford, H. A. Lagar-Cavilla, A. Varshavsky, V. Ganapathy, and L. Iftode, "Security versus energy tradeoffs in host-based mobile malware detection," in *Proceedings of the 9th international conference on Mobile systems, applications, and services (MobiSys'11)*, (New York, NY, USA), pp. 225–238, ACM, 2011.
105. S. Chakradeo, B. Reaves, P. Traynor, and W. Enck, "Mast: triage for market-scale mobile malware analysis," in *Proceedings of the sixth ACM conference on Security and privacy in wireless and mobile networks (WiSec'13)*, (New York, NY, USA), pp. 13–24, ACM, 2013.
106. S. Xu, H. Qian, F. Wang, Z. Zhan, E. Bertino, and R. S. Sandhu, "Trustworthy information: Concepts and mechanisms," in *Web-Age Information Management, 11th International Conference, WAIM 2010, Jiuzhaigou, China, July 15-17, 2010. Proceedings* (L. Chen, C. Tang, J. Yang, and Y. Gao, eds.), vol. 6184 of *Lecture Notes in Computer Science*, pp. 398–404, Springer, 2010.

107. H. Qian and S. Xu, "Non-interactive editable signatures for assured data provenance," in *First ACM Conference on Data and Application Security and Privacy, CODASPY 2011, San Antonio, TX, USA, February 21-23, 2011, Proceedings* (R. S. Sandhu and E. Bertino, eds.), pp. 145–156, ACM, 2011.
108. Z. Sanaei, S. Abolfazli, A. Gani, and R. Buyya, "Heterogeneity in mobile cloud computing: Taxonomy and open challenges," *Communications Surveys Tutorials, IEEE*, vol. PP, no. 99, pp. 1–24, 2013.
109. N. Fernando, S. W. Loke, and W. Rahayu, "Mobile cloud computing: A survey," *Future Generation Computer Systems*, vol. 29, no. 1, pp. 84 – 106, 2013.
110. A. N. Khan, M. M. Kiah, S. U. Khan, and S. A. Madani, "Towards secure mobile cloud computing: A survey," *Future Generation Computer Systems*, vol. 29, no. 5, pp. 1278 – 1299, 2013.
111. S. Bugiel, S. Nürnberger, A.-R. Sadeghi, and T. Schneider, "Twin clouds: secure cloud computing with low latency," in *Proceedings of the 12th IFIP TC 6/TC 11 international conference on Communications and multimedia security (CMS'11)*, (Berlin, Heidelberg), pp. 32–44, Springer-Verlag, 2011.
112. J. Oberheide, E. Cooke, and F. Jahanian, "CloudAV: N-Version Antivirus in the Network Cloud," in *Proceedings of the 17th USENIX Security Symposium*, (San Jose, CA), July 2008.
113. R. J. Hulsebosch, A. H. Salden, M. S. Bargh, P. W. G. Ebben, and J. Reitsma, "Context sensitive access control," in *Proceedings of the tenth ACM symposium on Access control models and technologies*, SACMAT '05, (New York, NY, USA), pp. 111–119, ACM, 2005.
114. M. L. Damiani, E. Bertino, B. Catania, and P. Perlasca, "GEO-RBAC: A spatially aware RBAC," *ACM Trans. Inf. Syst. Secur.*, vol. 10, Feb. 2007.
115. F. Roesner, T. Kohno, A. Moshchuk, B. Parno, H. Wang, and C. Cowan, "User-driven access control: Rethinking permission granting in modern operating systems," in *Security and Privacy (SP), 2012 IEEE Symposium on*, pp. 224–238, 2012.
116. M. Conti, V. T. N. Nguyen, and B. Crispo, "CRePE: context-related policy enforcement for Android," in *Proceedings of the* $13^{th}$ *international conference on Information security*, ISC'10, (Berlin, Heidelberg), pp. 331–345, Springer-Verlag, 2011.
117. A. A. Yavuz, P. Ning, and M. K. Reiter, "Baf and fi-baf: Efficient and publicly verifiable cryptographic schemes for secure logging in resource-constrained systems," *ACM Trans. Inf. Syst. Secur.*, vol. 15, pp. 9:1–9:28, July 2012.
118. Z. Zhou and D. Huang, "Efficient and secure data storage operations for mobile cloud computing," in *Network and service management (cnsm), 2012 8th international conference and 2012 workshop on systems virtualiztion management (svm)*, pp. 37–45, 2012.
119. D. Yang, G. Xue, X. Fang, and J. Tang, "Crowdsourcing to smartphones: incentive mechanism design for mobile phone sensing," in *Proceedings of the 18th annual international conference on Mobile computing and networking*, Mobicom '12, (New York, NY, USA), pp. 173–184, ACM, 2012.
120. Q. Li and G. Cao, "Providing privacy-aware incentives for mobile sensing," in *Pervasive Computing and Communications (PerCom), 2013 IEEE International Conference on*, pp. 76–84, 2013.
121. C. Cornelius, A. Kapadia, D. Kotz, D. Peebles, M. Shin, and N. Triandopoulos, "Anonysense: privacy-aware people-centric sensing," in *Proceedings of the 6th international conference on Mobile systems, applications, and services*, MobiSys'08, (New York, NY, USA), pp. 211–224, ACM, 2008.
122. D. Christin, C. Rokopf, M. Hollick, L. A. Martucci, and S. S. Kanhere, "Incognisense: An anonymity-preserving reputation framework for participatory sensing applications," *Pervasive and Mobile Computing*, vol. 9, no. 3, pp. 353 – 371, 2013. Special Issue: Selected Papers from the 2012 IEEE International Conference on Pervasive Computing and Communications (PerCom 2012).

# Index