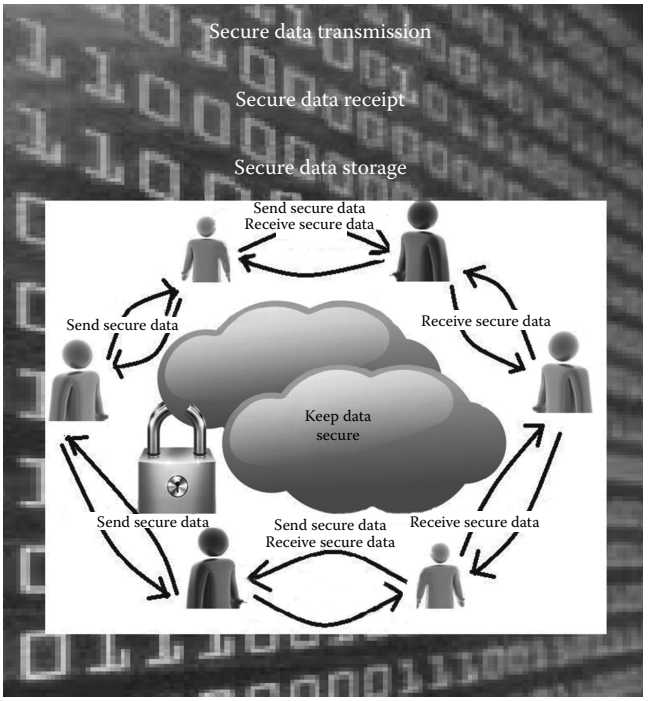
LECTURE NOTES

Mobile Technology and Cloud Computing

Session 12
Privacy and Security in Mobile Cloud Computing

9

Privacy and Security in Mobile Cloud Computing



ABSTRACT In mobile cloud computing, data are stored in the cloud. As both data access and storage are done outside the mobile device and into a remote cloud server, the user has no personal control over it. Therefore, security and privacy are the major challenges for the network security managers. With the rapid growth of social media and mobile web users (e.g., smartphone or tablet users), malicious threats have also increased. In this chapter, various aspects of security issues and solutions are analyzed in the field of mobile cloud computing.

KEY WORDS: *security, privacy, authentication, intrusion.*

9.1 Introduction

In this twenty-first century, mobile devices and wireless communication technologies are gaining popularity. Now, handheld devices such as smartphones, personal digital assistants (PDAs), and tablets are used to perform a number of tasks which were previously done by PCs. Such devices are resource-limited and do not provide the same efficiency and results as PCs. On the other hand, cloud has unlimited resources and works on "pay-as-you-consume" mode. A "cloud" is an elastic execution environment of resources involving multiple stakeholders and providing a metered service at multiple granularities for a specified level of quality of service. Cloud computing is a method of computing in which dynamically scalable and virtualized resources are provided as services over the Internet. Cloud computing consists of a number of servers and switches running in a physical layer to provide various cloud services, more commonly known as infrastructure as a service (IaaS), software as a service (SaaS), platform as a service (PaaS), data storage as a service (DaaS), communication as a service (CaaS), hardware as a service (HaaS), business as a service (BaaS), and security as a service (SecaaS) [2]. The cloud service providers (CSPs) are responsible for running, maintaining, managing, and upgrading cloud hardware to meet the increasing requirement of users. Mobile cloud computing (MCC) incorporates cloud computing into the mobile environment. MCC is a recent development in the field of mobile networks. Mobile computing is a term used to describe technologies that enable people to access network services anyplace, anytime, and anywhere. MCC refers to "an infrastructure where both data storage and the processing happen outside of the mobile device. Mobile cloud applications move the computing power and data storage away from mobile phones and into the cloud, bringing application and mobile computing to not just smartphone users but a much broader range of mobile subscribers" [1]. MCC provides simple and easy infrastructure for mobile applications and services. It enables users to flexibly utilize resources on demand, takes full advantage of cloud computing, and brings new types of services and facilities to users by providing ubiquitous service access. This integration of two different technologies, namely cloud computing and mobile networks, faces many technical challenges, such as low bandwidth, availability, heterogeneity, computing offloading, data accessing, security, privacy, and trust [1]. This chapter is mainly focused on privacy and security issues in MCC.

There has been a drastic increase in the use of smartphones in recent years. According to ABI Research predictions, the number of mobile cloud users will grow from 42.8 million (1.1% of total mobile users) in 2008 to 998 million (19% of total mobile users) in 2014 [2]. The security of smartphones is becoming increasingly important because they offer advanced services such as web browsing, instant massing, e-commerce, as well as personal and economical information storage. The proliferation of mobile malware increased by 46% in 2010 compared to 2009 [3]; 74% of chief information officers and IT executives are not willing to adopt cloud services because of the risks associated with security and privacy [2].

9.2 Security Needed in Different Levels for Securing Mobile Cloud Computing

In the mobile cloud environment, the user employs his or her mobile devices, such as smartphones, tablets, PDAs, and so on, to store data in the cloud with the use of communication channels. Gharehchopogh et al. [4] discuss security of mobile device data using cloud, with

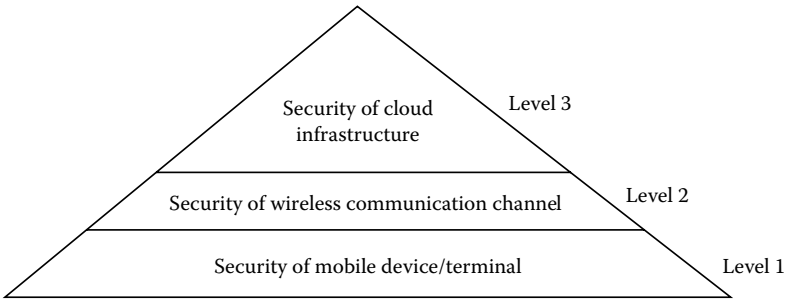


FIGURE 9.1
Security levels in mobile cloud computing.

the main focus on securing the cloud environment. In Ko et al. [5], security issues are divided into two categories, namely mobile network user's security and cloud security. In [6–8], the risk and issues related to mobile cloud computing are divided into three levels: mobile device/terminal, mobile network or wireless channel, and the cloud. Hence, we can say that the security and privacy risk should be deeply analyzed in three aspects:

1. Security of mobile device/terminal
2. Security of wireless communication channel
3. Security of cloud infrastructure

Three different levels of security are needed, as shown in Figure 9.1.

9.2.1 Level 1: Security Issues in Mobile Devices

In Level 1, the focus is on security and risk issues related to mobile devices. In [4–8], the main concern is the security of handheld devices. These devices have open operating systems, third-party applications, and wireless access to the Internet anywhere, anytime. Today, mobile phones are getting smarter with the advancement of hardware performance, technology, and communication bandwidth (3G, 4G, WiMax, etc), not only providing voice calls or Internet access but also applications and services that are possible by PCs and laptops. Thus, smartphones are now used as an enterprise tool in businesses, which helps increase the productivity of employees by allowing them to interact with their customers, partners, and colleagues. As smartphones are capable of supporting services and applications equal to PCs and desktops, it is vulnerable to the threats and risks as in PCs and desktops. Security issues in mobile devices with examples are given in Table 9.1. They mainly include malware, worms, Trojan horse, vulnerable applications, and OS. Even data can be compromised if the device is stolen, lost, or tampered with.

TABLE 9.1
Security Issues in Mobile Devices with Examples

Security Levels	Security Issues	Examples
Level 1: Mobile devices/ terminal	Information-stealing malwares, spam, phishing, data loss from lost or stolen devices, data leakage from poorly-written applications, vulnerabilities in hardware or OS, unsecured Bluetooth or Wi-Fi	Zimto and NickspyTrojans are information-stealing malwares, fake websites, digital wallet hacking, unwanted message from unknown vendors.

9.2.1.1 Approaches to Mitigate Security Issues Related to Mobile Devices

There are different approaches to lessen the security issues related to mobile devices. Anti-malware programs are run on the devices to identify and delete Trojan horse, viruses, and worms. Periodically updating the OS and downloading applications from known vendors such as Google, Apple, and Microsoft can also help. Also, unexplained links should not be tried, receiving data transmission from strange phones should be avoided, new, unauthorized software should not be installed, and the interface of Bluetooth, Wi-Fi, and so on, should be shut down. If a device is stolen or lost, there must be some remote data wiping technique so that the data cannot be misused. The mobile device can also use hardware-based encryption techniques for internal and external memory support.

9.2.2 Level 2: Security Issues in Communication Channels

Level 2 deals with issues related to securing the wireless communication channel [6–8] between the mobile and cloud servers. Mobile devices access their resources and services through communication channels from cloud servers. This increases the number of WAP (wireless application protocol) gateways and IMS (IP multimedia subsystem) equipment in the IP network, giving rise to many new security threats in the mobile Internet. These mobile terminals access phone service, short message services (SMS), and other Internet services using 3G, Wi-Fi, WiMax, and Bluetooth. Such broad access methods result in an increase in security risks associated with networks, causing information leakage and malicious attacks. A number of attacks have been identified in the communication between wireless mobile devices and the cloud environment. When mobile devices communicate with the cloud, they are more vulnerable to communication threats. With rapid increase in cloud usage, there is an increase in the number of security issues related to the communication channel. Even the free Wi-Fi connections in public places (e.g., airport, cafés) are prone to attacks because of the weak encryption techniques used in Wi-Fi. The attacker can break wireless interface and steal sensitive data. Illegal terminals can access the network with fake ID and carry out malicious activities. Security issues in the communication channel with examples are given in Table 9.2. There are a number of attacks identified in communication channels such as access control attacks, confidential attacks, integrity attacks, authentication attacks, and availability attacks.

9.2.2.1 Approaches to Mitigate Security Issues Related to Communication Channel

For protecting data from leakage while being transmitted to servers, a number of approaches are available. The mobile users mainly encrypt data while transmitting into cloud so that an adversary cannot understand or even be able to get the data. Secure transmission protocols such as https and SSL can be used to transfer data; even VPN (virtual private network) can

TABLE 9.2

Security Issues in Communication Channel with Examples

Security Levels	Security Issues	Examples
Level 2: Communication channel/mobile network	Access control attacks, confidentiality attacks, integrity attacks, authentication attacks, availability attacks.	War driving, rough APs, MAC spoofing, WEP cracking, Man-In-The-Middle attack, Evil Twin, AP phishing, frame injection, reply attacks, guessing, VPN login cracking, LEAP cracking, DoS, Beacon flood, etc.

be used. Socket programming is also used for secure transmission of sensitive data in a cloud environment. Also, public key encryption is used for protecting Man-In-The-Middle (MITM) attacks. Strong password and biometric authentication should be used to enhance data security during transmission. Even the rough access points at public places should be avoided for security reasons. Switching off the wireless interfaces, such as Wi-Fi and Bluetooth, after using the mobile device will also help.

9.2.3 Level 3: Security Issues in Cloud Computing

Level 3 contains the most important issues in mobile cloud computing and prevents a large number of mobile users from using cloud services [4–8]. The users offload their data to the cloud and lose control over those data. An increasing combination of smartphones with cloud infrastructure raises the possibility of threat attacks in the cloud. Cloud computing is based on virtualization technology, and if there is some vulnerability in the virtualization software, the data of one user on the same physical server can be leaked to that of other users. There is also a need for proper access control and data management according to the needs of the consumer. Security issues of cloud infrastructure with examples are given in Table 9.3. In [4,8], isolation of data from other users where the user data are stored and cloud server flexibility are discussed. In Ko et al. [5], the integrity of offloaded data, authentication, and digital rights management (DRM) are dealt with as the main issues related to the cloud environment. In Hui et al. [6], user data and privacy protection, platform reliability from insider and outsider attacks, and access control are the main issues discussed. A number of cloud security issues, such as attacks on virtual machines, availability and single-point failure, phishing, authorization and authentication, and security management in hybrid cloud, are evaluated by Morshed et al. [7].

9.2.3.1 Approaches to Mitigate Security Issues in Cloud Infrastructure

In [4–8], the authors provide different techniques and mechanisms for the protection of data in the cloud. To increase the trust of customers for storing data in the cloud server, it should provide privacy, authentication, confidentiality, and availability of services. These security mechanisms must be strong enough to handle attacks by adversaries and hackers. If a user subscribes to a cloud service, then the CSPs should provide the location of the stored data. The location should be free from geopolitical issues. Thus, the privacy and security of data should be maintained. In the cloud, security is promised using current security technologies such as VPN, access control, encryption, and other such means. There must be a mechanism to recover the user's data if the data is lost or erased by an attacker. There should also be a secure and efficient key management mechanism for the cloud environment. Cloud should use an implicit authentication technique to reduce the risk of fraud in a mobile cloud.

TABLE 9.3

Security Issues in Cloud Infrastructure with Examples

Security Levels	Security Issues	Examples
Level 3: Cloud environment	Integrity, digital rights management, virtual machine attacks, phishing, authentication and authorization attacks, platform level attacks	Data and application integrity, pirating and illegal distribution of digital contents, side channel attacks, SQL injection, etc.

9.3 Security Issues in Mobile Cloud Environment

The integration of mobile devices with cloud infrastructure has given rise to a number of security issues. This includes authentication, authorization, data security, application security and integrity, privacy, digital right management, and so on. We will discuss all these issues with their existing schemes.

9.3.1 Application Security

A number of applications are run on mobile devices. These applications are mainly used for managing personal information and for business needs. These applications include Internet chatting, electronic mails, games, schedulers, and so on. Mobile applications now use cloud services hosted on cloud servers. Providing security of such mobile cloud application is of great concern. Most of these applications are downloaded from Google store, Apple store, Nokia store, or a third-party application store. Such stores do not have any scheme to remove malware from applications. According to Dinh et al. [1], 10 billion applications were downloaded from the Android market in 2010, and 250,000 applications contained malware. Good applications are modified using malicious codes and are spread through unofficial repositories. Such applications leak private data, dial premium numbers, and are backdoor-triggered via SMS, for example. So, there must be some schemes that should take care of the security of such applications and the threat related to mobile cloud applications. In the next section, we discuss some schemes used for applications security and the related threats.

9.3.1.1 Existing Schemes for Application Security

An application-specific firewall was proposed, called WallDroid, in Kilinc et al. [9]. WallDroid is mainly a firewall for Android applications with extra functions. The key component used for providing security in this architecture is VPN technology and the cloud-to-device messaging (C2DM) framework for Android. In this framework, cloud is used to keep track of millions of applications with their reputation and to compare the traffic with the list of known malicious IP servers. Every application has its unique ID, which is a combination of a certificate and a hash value. Android applications are classified into three categories according to their reputation: The Good, The Bad, and The Unknown. Well-known applications are The Good applications, while known to be malicious applications are those that are not known to be The Good and The Bad.

If the application is good, it is directly connected, while if it is a known bad application, the Internet connection is blocked. For unknown applications, VPN service is used for Internet connection via a VPN server, as shown in Figure 9.2. The VPN then monitors the data traffic to see whether it is malicious or not and whether it is sending personal data in clear text or not. If the VPN server determines it as malicious, then it blocks its traffic. The earlier framework is proposed to determine malicious and unknown Android applications [9]. A secure web referral service is proposed by Xu et al. [10], which uses a secure search engine (SSE) for mobile devices to protect the mobile website against phishing and SSL strip-based MITM attack. In this, a cloud-based virtual computing is used for providing each user a VM as personal sec-proxy to analyze the web traffic. In VM, the SSE uses web crawling to check a valid IP address and certification chain. A phishing filter is also used for checking URLs with optimized execution time.

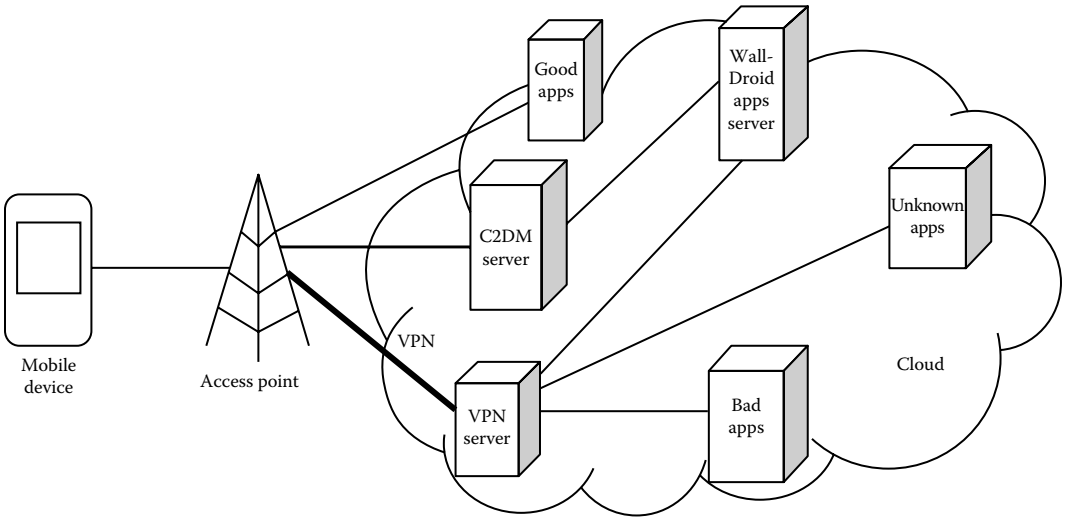


FIGURE 9.2
Architecture of WallDroid.

The components of the SSE service model include URL services, SSL verifier, phishing filter, SSE crawler, SSE services, DNS services, and storage services. Mainly SSL verifiers and phishing filters are used to provide secure web browsing.

The processing of SSE is shown in Figure 9.3. SSL verification is used to counter MITM attack, while a phishing attack is countered by the phishing filter. The performance of SSE

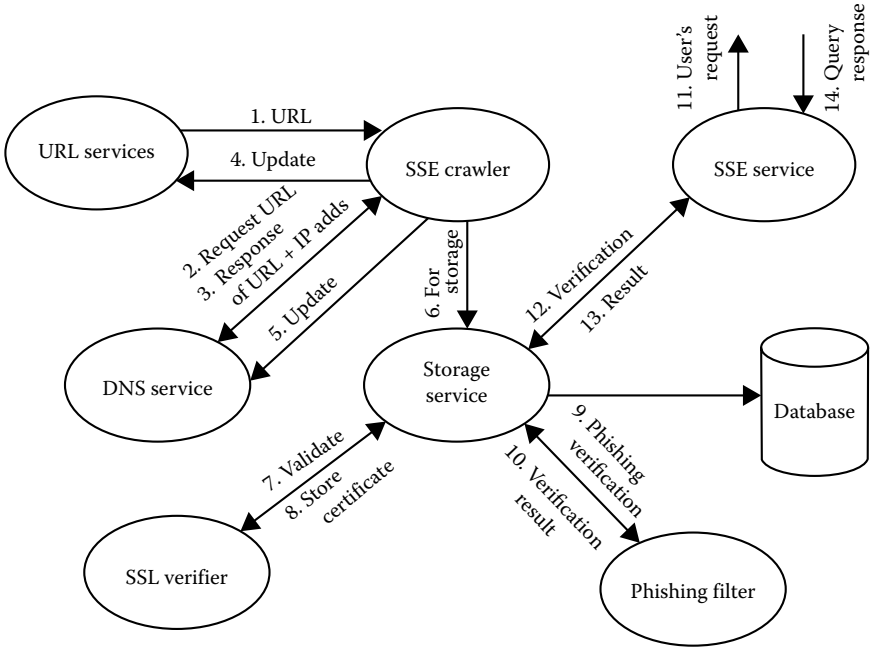


FIGURE 9.3
Working principle of secure search engine.

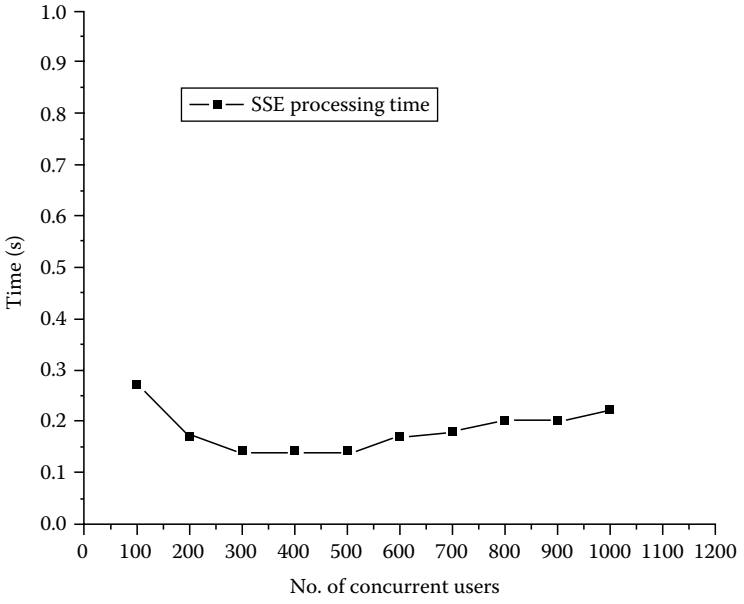


FIGURE 9.4
Processing time taken by SSE services.

is calculated for responding to a valid request. In this experiment, the *x*-axis represents number of concurrent users, and *y*-axis represents time taken to process (in seconds). The experiment is carried out with an empty cache and 100 users. The cache will be filled with inspected websites. Figure 9.4 shows that, at the beginning, the processing time is high but it drops as the time increases due to caching.

A framework is proposed that ensures the security of component-based applications. This model secures the data transmission between the component of the same application at the installation on the mobile device and when being updated [11]. Different security schemes are used for different types of data employed by the applications according to mobile device's energy consumption. It also ensures confidentiality and integrity of the application's components. This framework is also called secure mobile cloud (SMC) and includes five managers for securing mobile cloud applications. The managers are the mobile manager, the mobile and cloud security manager, the optimization manager, the application manager, and the policy manager. The mobile manager collects data and events on mobile devices and sends them to the appropriate manager. Security managers take care of composite security of mobile devices and the cloud. The optimizer manager collects and sends the sensor's data. The application manager checks the integrity of application at setup, while the policy manager determines which security component is required for different security levels. In this framework, application integrity is verified at installation and at updating. Integrity is ensured by checking the existence of applications in stores such as Amazon, Apple, Google store, and so on. Low-energy consuming and component based security architecture for mobiles or LECCSAM [8] is a flexible security scheme that allows terminal users to specify the extension of security properties that they will prefer to integrate with the data using HTTPS.

9.3.2 Authentication Issues

The migration of private and enterprise data to the cloud raises security and privacy issues. To access these sensitive data only by the legitimate users, an authentication protocol is used. Traditionally, a user provides his or her password to the requested server for authentication, which may be attacked. In mobile cloud, legal user authentication becomes an important issue. In Figure 9.5, a simple example of the authentication process is shown.

In the following section, different authentication schemes are proposed to authenticate users in cloud using their mobile devices.

9.3.2.1 Existing Authentication Schemes

An authentication scheme that does not need to enter password, user name, biometric data, and so on, for authentication is proposed in Chow et al. [12]. This framework simply utilizes TrustCube for authentication and generates a score according to user's behavior. The generated probabilistic authentication score is then compared with a threshold value to determine whether the client is authentic or not. The authentication score is not fixed, and can be varied for different applications. This scheme consists of four modules: client devices, authentication consumer, authentication engine, and data aggregator, as shown in Figure 9.6. The client device generates the noticeable context and action such as Internet browsing history, call records, location history, MMS, SMS, phone information, and so on.

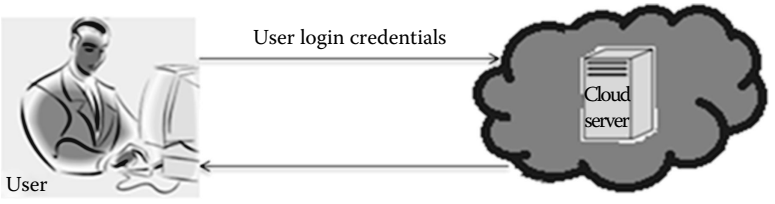


FIGURE 9.5
Simple example of authentication process.

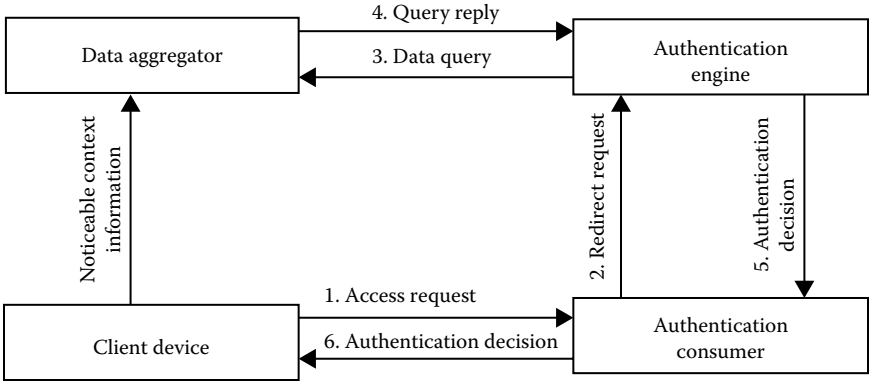


FIGURE 9.6
Authentication architecture.

The data generated by the client device are stored in the local cache until collected by the data aggregator. The authentication engine extracts noticeable context information from the data aggregator and the authentication policies from the authentication consumer to authenticate the mobile device. The authentication policies depend on the client's request. Finally, the authentication engine responds to the client according to the data provided to it through the authentication consumer.

A next-generation authentication scheme for mobile and CE devices, which uses a zero knowledge proof (ZKP) technique for authentication ID, is proposed in Grzonkowski et al. [13]. This scheme is anti-phishing and does not reveal the user's password to the visiting website. The user is not redirected to other web pages after login. This scheme is called SeDiCi 2.0, which consists of three entities: client (C), services (S), and authentication services (AS). Figure 9.7 shows the data flow between the three entities. Client creates an account in AS using his or her password in client application to generate the public key.

The client registers to the service. Then the service verifies the client and records its login detail. The client login and public key are send to AS. To authenticate to service, the client again visits to service and gains Auth_ID from service as response. Auth_ID and URI are then sent to AS, and AS verifies the URI corresponding to Auth_ID. Now the URI is exposed with the given Auth_ID, then the client sends login to service, and service verifies client using Auth_ID. If the verification is successful, the client is authenticated.

An advance protocol for authentication based on biometric encryption is proposed in Zhao et al. [14]. This can be used for future authentication when the mobile devices are equipped with biometric sensors, as shown in Figure 9.8.

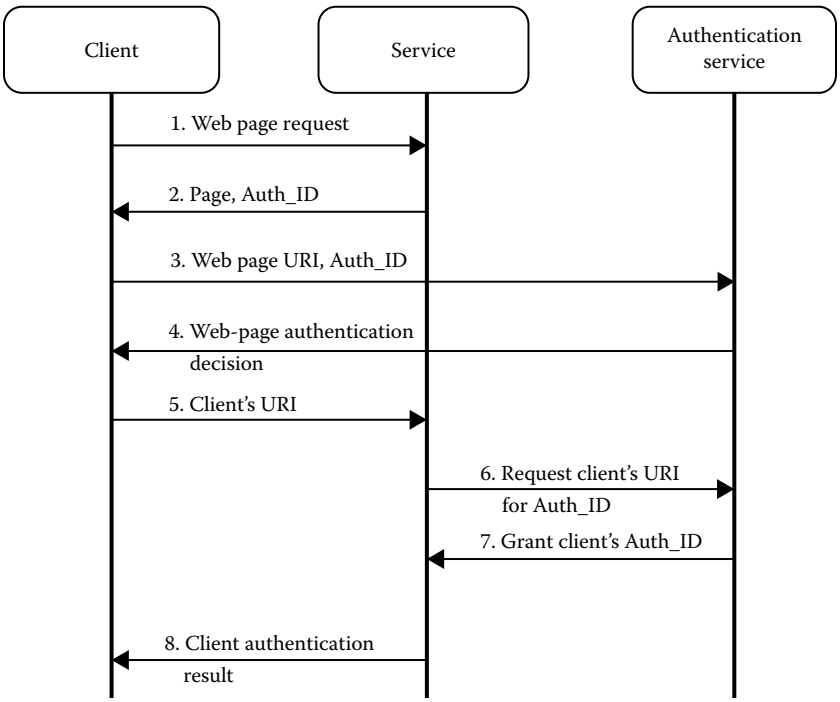


FIGURE 9.7
Dataflow of SeDiCi 2.0 for authentication.

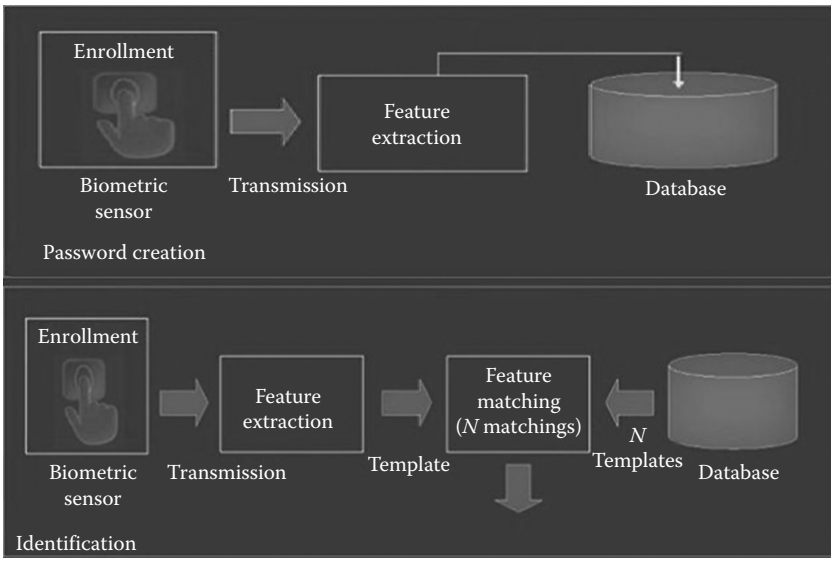


FIGURE 9.8
Biometric encryption-based authentication scheme.

This scheme is more reliable than the traditional password-based schemes because biometric data is difficult to forget, forge, share, or lose. User has his record of biometric features stored in the cloud database.

9.3.3 Data Security

In this section, we will mostly focus on the security of mobile device data that is offloaded to the cloud storage. Mobile devices contain private, commercial, financial, and enterprise data. Leakage of such sensitive data to others may lead to personal and economic loss. More threats are involved when such important data are offloaded to the cloud. The user loses his control over off-premises data. So, such offloaded data must be kept safe and confidential. There must be some mechanism to know data integrity and data to be available when needed. The user also should be aware of where his data are located. Some schemes are discussed in the next section, which focuses on data security in mobile cloud computing.

9.3.3.1 Existing Schemes for Data Security

Mobile device data security is discussed in [19–27], with focus on integrity and confidentiality. Different cryptographic algorithms are used to protect the data from adversaries. These algorithms include incremental cryptography, attribute-based encryption, digital signature, identity-based encryption, message authentication code, and hashing functions. Some authors have also considered the mobile's resource limitation while implementing their schemes.

An incremental cryptography-based trusted computing is used to provide integrity to mobile device files (F). This system consists of three elements: mobile client, cloud service provider (CSP), and trusted third party (TTP) [15]. CSP provides the management, operation, and allotment of cloud resources and services. TTP manages the configuration and installation of a secure coprocessor on the remote cloud. It distributes the secret key K_S to

the associated mobile client and generates a message authentication code (MAC) on behalf of the mobile clients. It has three phases: the initialization phase, the data updating phase, and the integrity verification phase. In the initialization phase, the mobile device data is prepared with incremental authentication. For every file block, an incremental MAC_{FX} is created using K_S, as given in Equation 9.1. The file block is then transferred to the cloud.

$$MAC_{FX} = \sum_{i=1}^{k} HMAC(F_i, K_S) \tag{9.1}$$

The data updating phase consists of three main operations: creation of file blocks, insertion of file blocks, and deletion of file blocks. If the file is available in the cloud, then it sends a copy to the mobile device as well as to the trusted crypto coprocessor (TCC). TCC generates the MAC'_{FX} using K_S and sends it to the mobile client. The integrity is checked by comparing MAC_{FX} with MAC'_{FX}. If both are equal, the file is inserted at B block at the ith position of the file, and again incremental *MAC* is generated using old *MAC* and K_S. The block deletion operation is same as block insertion at the ith position of the file. The difference in deletion is that *MAC* updating depends on the deleted block and old *MAC*. The main integrity mechanism is offloaded to *TCP*, which saves the processing power of mobile devices. The coprocessor generates the *MAC* according to the request and sends it back to the mobile device. The *MAC* of the coprocessor and *MAC* of mobile device are compared and, if they are equal, the integrity is successfully verified, as given in Equation 9.2:

$$MAC(Mob) = MAC(TCP) \tag{9.2}$$

A scheme for authentication of mobile user and the integrity of mobile device data is discussed in Hsueh et al. [16]. In this scheme, standard encryption algorithm, hash function, digital signature, random number, and secret value are used to provide overall mobile device data security when offloaded to the cloud. SSL is used for secure access, and access lists (ACLs) are also available for individuals and the group. This framework consists of mainly four modules: mobile device (*MD*), cloud service provider (CSP), certification authority (CA), and telecommunication module (TM), as shown in Figure 9.9.

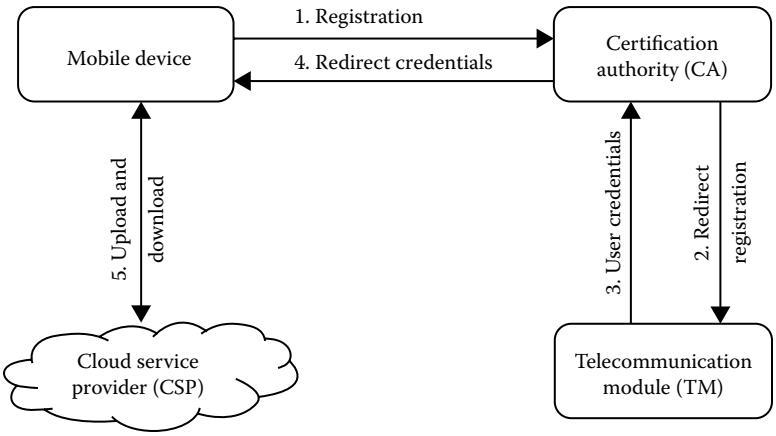


FIGURE 9.9
Architecture of secure data storage in cloud.

In this model, the certifying authority is responsible for the authentication of mobile devices. The telecommunication module generates and keeps records of the mobile device's passwords and other information used to access cloud services.

In this framework, it is assumed that the secure key (*SK*), public key (PK), and session key (*SEK*) are distributed securely among the mobile devices, telecommunication model, and CA. To access the services of the cloud, the mobile user has to register onto the cloud through CA. After successful registration, TM generates a password (*PWD*) for mobile devices to use the cloud resources. This is given in Equation 9.3.

$$MD \rightarrow CA: E_{PKTM}E(MU, Num, TK), U_N, S_{SKMU}(MU, Num), H(MU, Num) \tag{9.3}$$

where

MU represents mobile user's name
Num represents mobile user's number
TK is combination of Num and cloud PWD random
U_N is the random number generated for identity proof
H is the standard hash function, EPKTM represents encryption with the PK of TM
S_{SKMU} generates a signature for mobile user using a cryptographic function on the passed value and SK of the mobile device

When the message is received at CA, it authenticates the user with the received signature. If the user is a valid one, the following message is sent to TM, as given in Equation 9.4.

$$CA \rightarrow TM: E_{PKTM}(MU, Num, TK), U_N, S_{SKCA}(H(MU, Num)) \tag{9.4}$$

The TM authenticates the CA using the S_{SKCA} key. If the CA is authenticated, the TM registers the mobile user and saves the mobile user's information in the local database. The data is used for future verification. The TM generates PWD for the mobile device and encrypts it with the mobile device's PK for secure transmission. PWD is again encrypted with TK to ensure that only an authorized user can decrypt it on receiving. TM forwards the secure information to the mobile device through CA, as given in Equations 9.5 and 9.6.

$$TM \rightarrow CA: E_{PKMU}(MU, Num, U_N, E_{TK}(PWD)) \tag{9.5}$$

$$CA \rightarrow MD: E_{PKMU}(MU, Num, U_N, E_{TK}(PWD)) \tag{9.6}$$

Now, the mobile device encrypts the file with SEK and uploads the file along with PWD, MU, and S_{SKMU} on the cloud as given in Equation 9.7.

$$MD \rightarrow C: PWD, MU, E_{SEK}(Data), S_{SKMU}(H(MU \,||\, SV \,||\, E_{SEK}(Data))) \tag{9.7}$$

where SV represents the secret value generated by the mobile device and is known to MD, CA, and TM. To upload a file to the cloud, MD has to send PWD, MU, and $H(MU||SV)$. Cloud regenerates the hash value using MU and SV, and then compares the result with the received signature for authentication. Then, the cloud sends the encrypted file to MD along with a signature as given in Equation 9.8.

$$C \rightarrow MD: E_{SEK}(Data), H(E_{SEK}(Data) \,||\, SV) \tag{9.8}$$

The mobile device receives the signature and decrypts the file using SEK.

Secure data processing is achieved through trust management and private data isolation. In this model, identity-based cryptography and attribute-based data access control are used for trust management [17]. It ensures security and privacy for mobile devices with the help of multi-tenant secure data management and trust management, and extended semi-shadow image (ESSI)-based data processing model. This architecture consists of three domains: cloud public service and storage domain, cloud trusted domain, cloud mobile and sensing domain. Cloud service and storage domain provides SaaS, while the cloud trusted domain manages the certificate distribution, key distribution, and identity management. In attribute-based identity management, publicly known attributes are used for private key generation for secure communication. This identity can be used as a signature to authenticate the user. ESSI is also called clone of mobile devices running in the cloud trusted domain. ESSI increases the storage and processing power of the mobile devices. It also provides security and privacy to the data and information of the device. Secure policies and rules are used in the cloud trusted domain with the help of a distributed firewall, which is used to check the incoming and outgoing packet for the malware.

The data management system is divided into two groups: critical and normal data. Critical data are encrypted with the user-generated key, while the normal data are encrypted using the cloud-generated key. The incoming data received by ESSI are classified as normal or critical. If the data are identified as critical data, then they are passed through the encryption, decryption, and verification (EDV) module and stored in the secure storage of ESSI. The masking procedure preserves the privacy of the data depending on user preference. This gives scalability, protection to critical data, computation distribution, and resistance to single-point failure.

In Table 9.4, three data security schemes are compared according to their encryption methods and security features, and also their drawbacks.

In [18–24], secure storage of mobile device data to cloud is discussed with consideration of mobile device resource constraints. The mobile device creates a file and processes it and finally uploads it to single cloud or multiple clouds. Three schemes are proposed to achieve security of mobile device data: encryption-based scheme (EnS), code-based scheme (CoS), and sharing-based scheme (ShS) [18]. The link between the mobile device and cloud is secured using media access protocols such as SSL, IPSec, and so on. This

TABLE 9.4

Comparison of Different Data Security Schemes

Encryption Scheme/Method/ Principle	Security Features			Other Features	Drawbacks
	Integrity	Confidentiality	Authentication		
Incremental cryptography (MAC) [15]	Yes	No	No	Energy efficient	Single-point failure, less scalable
Standard hash function, digital signature [16]	Yes	Yes	Yes	Simple and easy to implement	Less energy efficient and scalable
Attribute-based encryption and identity based cryptography [17]	No	Yes	Yes	Resistance to single-point failure, scalable, energy efficient	Mobile device is compromised if somehow ESSI is attacked and manipulated

framework provides data confidentiality and data integrity. Each of the three schemes deals with secure data uploading to the cloud and secure data downloading from the cloud to mobile devices.

A user must have the password, the encryption key (*EK*), and the integration key (*IK*) to upload and download files, as given by Equations 9.9 and 9.10:

$$EK = H(PWD \,||\, FN \,||\, FS) \tag{9.9}$$

$$IK = H(FN \,||\, PWD \,||\, FS) \tag{9.10}$$

In EnS, file *F* is encrypted using *EK* and produces *F′*, and message authentication code (MAC) is generated by using the hash function to *IK* and file for integrity check, as given in Equations 9.11 and 9.12.

$$F' = E(F, EK) \tag{9.11}$$

$$MAC = H(F, IK) \tag{9.12}$$

The mobile device (*MD*) sends concatenation of the encrypted file *F′* and hashing of File name (FN), and sends it to cloud storage (CS), as given in Equation 9.13:

$$F' \,||\, H(FN) \,||\, MAC \tag{9.13}$$

MD stores only FN and deletes *EK* and *IK*.

To download the file, *MD* sends hash FN, and CS searches the file using FN and sends the corresponding file to *MD*. PWD is needed to get *EK*, *IK*, and decrypt *F′* to get *F*, as shown in Equation 9.14:

$$F = D(F', EK) \tag{9.14}$$

MD generates MAC of decrypted file F using *IK* to check the integrity.

A scheme incremental cryptography refers to the improvement of existing schemes [18], i.e. encryption-based scheme, coding-based scheme, and sharing-based scheme. This scheme improves block insertion, deletion, and modification operation in terms of resource utilization of a mobile device [19]. This scheme also requires considerable energy at the initial stage but improves file modification in terms of turnaround time.

In this scheme, the file is divided into d blocks of n bits; hence, FS satisfies Equation 9.15:

$$FS \% d = 0 \tag{9.15}$$

Each file is encoded separately, and finally the encrypted file (EF) is generated by performing the concatenation operation, as given in Equations 9.16 and 9.17:

$$C_j = E_{EK}(F_j) \tag{9.16}$$

where $1 \leq j \leq d$

$$EF = C_1 \| C_2 \| C_3 \| ..... \| C_d \tag{9.17}$$

MAC is generated for each block, and the final MAC is generated by concatenation of all MACs, given in Equations 9.18 and 9.19:

$$MAC_{Fj} = HMAC_{IK}(F_j) \tag{9.18}$$

$$MAC = HMAC_{IK}(MAC_{F1} \| MAC_{F2} \| ..... \| MAC_{Fd}) \tag{9.19}$$

MD only keeps FN and d, while uploads H (FN), blocks MAC, and final MAC to the cloud. *MD* deletes original the file, *IK*, *EK*.

For downloading the file *F*, *MD* sends H (FN) to the cloud, and the cloud searches for the FN and sends the file *F* to *MD*. *MD* divides *F* into d blocks, and each block is decrypted using *EK*. Finally, each decrypted block is concatenated to produce *F*, as given in Equations 9.20 and 9.21:

$$F_j = DEK_j(C_j), \quad 1 \leq j \leq d \tag{9.20}$$

$$F = F_1 \| F_2 \| \ldots \| F_d \tag{9.21}$$

Integrity verification is done by comparing the new MAC calculated using decrypted *F* and the old MAC sent by cloud.

The incremental version of the other two (i.e., coding-based and sharing-based) schemes works in the same way to produce the block MAC and final MAC. The incremental version of EnS, CoS, and ShS consumes more resources because of the extra MAC and the final MAC generation. But the overall turnaround time balances this extra computation while block insertion, deletion, and modification are done.

In Ren et al. [18], some modifications are suggested when data is transmitted between *MD* and cloud storage. Attackers may also monitor the link and recover the file. It is also impossible for the *MD* to guarantee secure storage of data on the cloud. Hence, schemes suffer from MITM attacks.

All three schemes, namely EnS, CoS, and ShS, are improved with respect to security and are called the secure encryption-based protocol (SEnP), secure coding-based protocol (SCoP), and secure sharing-based protocol (SShP) [20]. In SShP, file *A* equal to the size of file *F* is generated by *MD* to participate in XOR for privacy of data. Digital signature and public key encryption are used for data integrity. Acknowledgment is sent back to *MD* from cloud to confirm secure storage. A random number *N* is used for acknowledgement.

MD uploads file *F*, MAC, digital signature, and *N* to the cloud, as given in Equation 9.22:

$$F'[j] \| \{\{H(H(FN + j) \| MAC \| HF'[j] \| N\}_{PR_{MD}}\}_{PU_{CSj}}) \tag{9.22}$$

where

PR is a private key

PU is the public key of *MD* and *CS*

CS computes *N* from the digital signature and sends it back to *MD* using its private key and *MD*'s public key, as given in Equation 9.23:

$$\{\{N\}_{PR_{CSj}}\}_{PU_{MD}} \tag{9.23}$$

MD decrypts using its private key and compares the value of the received *N* with the stored *N*. If the values of *N* are equal, then secure storage of data is verified.

The other two schemes follow the same procedure for upload of data using SCoP scheme; from *MD* to cloud is given by Equation 9.24:

$$F'[j] \,||\, \{\{H(FN + j) \,||\, MAC \,||\, H[F'[j]) \,||\, N\}_{PR_{MD}}\}_{PU_{CSj}} \tag{9.24}$$

Uploading of data through SEnP scheme from *MD* to cloud is given by Equation 9.25:

$$F' \,||\, \{\{H(FN) \,||\, MAC \,||\, H(F') \,||\, N)_{PR_{MD}}\}_{PU_{CSj}}\} \tag{9.25}$$

Downloading is done in same way as discussed in Zhou et al. and Shin et al. [21,22], respectively.

A scheme with the core idea of outsourcing data in the cloud provides trust-based security management in mobile cloud and also provides data confidentiality and fine-grained access control. This framework uses identity-based proxy-re-encryption (PRE) technique to provide security to the mobile cloud [23]. This mechanism protects the data from the untrusted cloud as the data is encrypted and also provides the benefits of cloud storage. The integrity is achieved based on MAC and public signature-based scheme. It consists of three entities: data owner, data server, and data sharer, as shown in Figure 9.10.

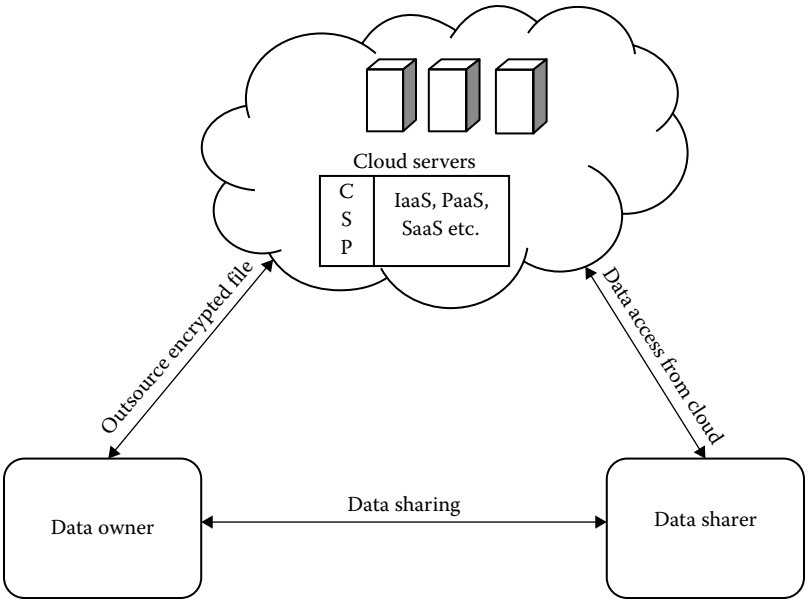


FIGURE 9.10
Data storing and sharing on cloud.

Data owner and data sharer use mobile devices to access cloud services through the Internet. Data owner forwards encrypted data on the cloud server and allows authorized data sharers to access data and decrypt the file. In this protocol, each user has its unique identity and secret key. The cloud stores data, which are delegated to the ciphertext encrypted with data owner's ID to requester ID.

A trusted platform module (TPM) used for all encryption key and key sharing among legal users for data protection is proposed in Shin et al. [22]. TPM function is developed in the secure domain of ARM TrustZone because most of ARM-based mobile devices are not equipped with the TPM chip. This DFCloud framework defines TPM-based secure channel setup, TPM-based key management, remote client attestation, and a secure key sharing protocol across multiple users/devices. In this scheme, client- and server-side encryptions are used. SSL is used for securing data transmission between the client and the server. DFCloud architecture consists of three components: client, DFCloud server, and commodity cloud storage services, as shown in Figure 9.11. TPM emulator is used instead of hardware TPM and provides the same security as hardware components do.

The cloud storage services in the client secure side consist of three subprocesses to maintain security. They are attestation components, key management components, and file handlers. The TrustZone monitor provides context switching between secure and normal worlds. The server acts as a proxy between the client and cloud storage. It consists of a file metadata store, which contains owner information and data block location, and provides services of file uploading and downloading with the use of the metadata store.

There are different protocols used for secure cloud storage services, such as user login, remote attestation, key creation, key sharing, and data uploading and downloading. The user employs his ID and password to read store files by a retrieving a key that is generated only after passing attestation services. Remote attestation protocol (RAP) is used for login and attestation. Platform configuration registers (PCRs) with the PCR number are sent to client TPM as a result of login and attestation. After login, key creation and management are done by use of PCR value, which contains the attestation value H. After passing the attestation test, the sealed EKA is unsealed and used. Key sharing is done between

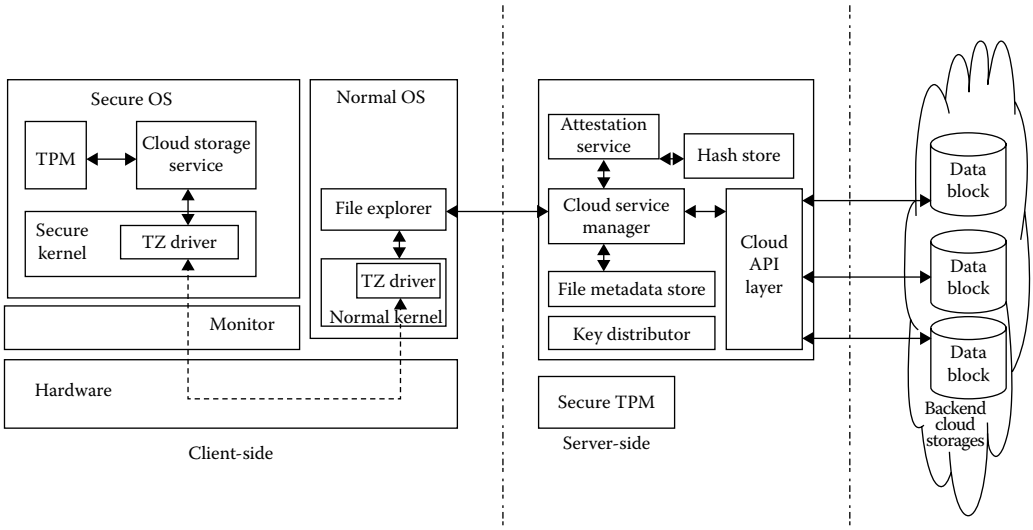


FIGURE 9.11
DFCloud architecture.

device A and device B. In key-sharing, the server distributes the nonce value to A and B. In this, asymmetric encryption is used for sharing the keys between TPM of device A with TPM of device B. A encrypt key EKA uses the public key of B and sends it to B. Then, B decrypts it using its private key. The sealed key is then stored in persistent storage.

Data uploading and downloading are done through file explorer of the client side and cloud service manager of the server side. File handler performs encryption or decryption using keys. Then file handler copies the files to local storage and a download session is established between file explorer and cloud service manager. File explorer sends a download request to server, and cloud service manager searches the metadata of the file stored. After searching in the store, the server sends the location of cloud storage to file explorer. Encrypted data is then decrypted using keys and stored locally. Uploading is the reverse of downloading.

Performance evaluation of files is done on the basis of three categories: without encryption, with encryption in normal world, and with encryption in secure world. Figure 9.12a shows the file uploading time, while Figure 9.12b shows the file downloading time of different sizes. Secure world can only perform cryptographic operation to 512 bytes at a time.

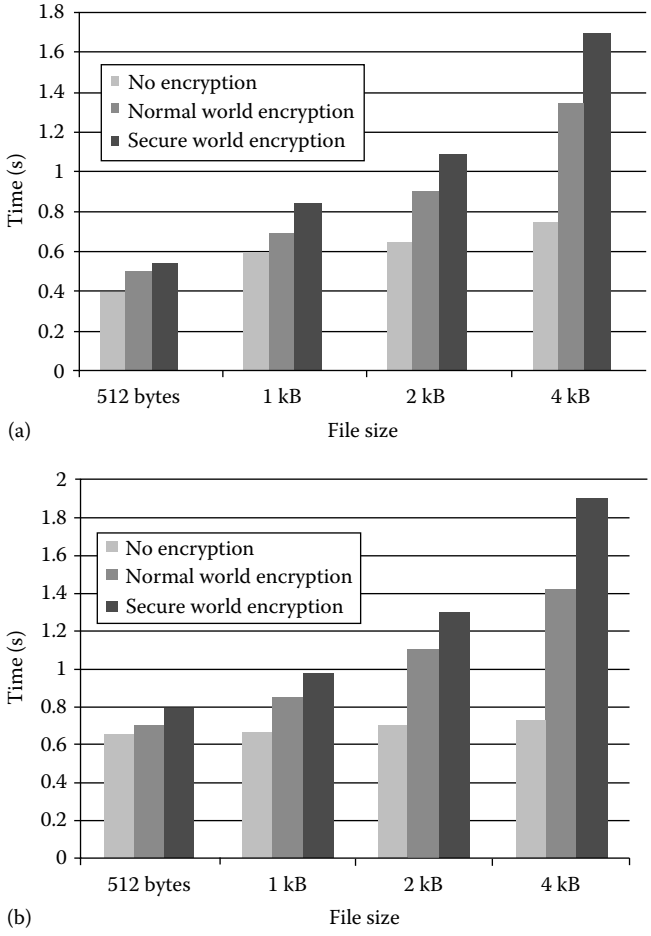


FIGURE 9.12
File upload/download performance. (a) File uploading. (b) File downloading.

Time increases with the increase of file size due to the context switching between the secure world and normal world. Time also increases with the cryptographic operation with increase of file size.

9.3.4 Digital Rights Management

Digital contents such as e-books, images, audios, videos, and so on, are now stored in cloud. Mobile users can access these contents using the Internet from the cloud servers. These digital contents could be pirated and distributed illegally. Digital rights management checks this abuse by regulating content usage. The DRM system allows only authorized users who have the license to access such contents.

9.3.4.1 Existing DRM Scheme

A cloud-based SIM DRM (CS-DRM) scheme for the mobile cloud environment has four main entities: a SIM card, a DRM agent, a custom player, and a CS-DRM-compliant browser [24], as shown in Figure 9.13.

The SIM card is used to provide subscriber identity, to authenticate between cloud clients, and also to verify the integrity of the license. The DRM agent is used to communicate between cloud clients and to implement logical rules. Custom player is used to

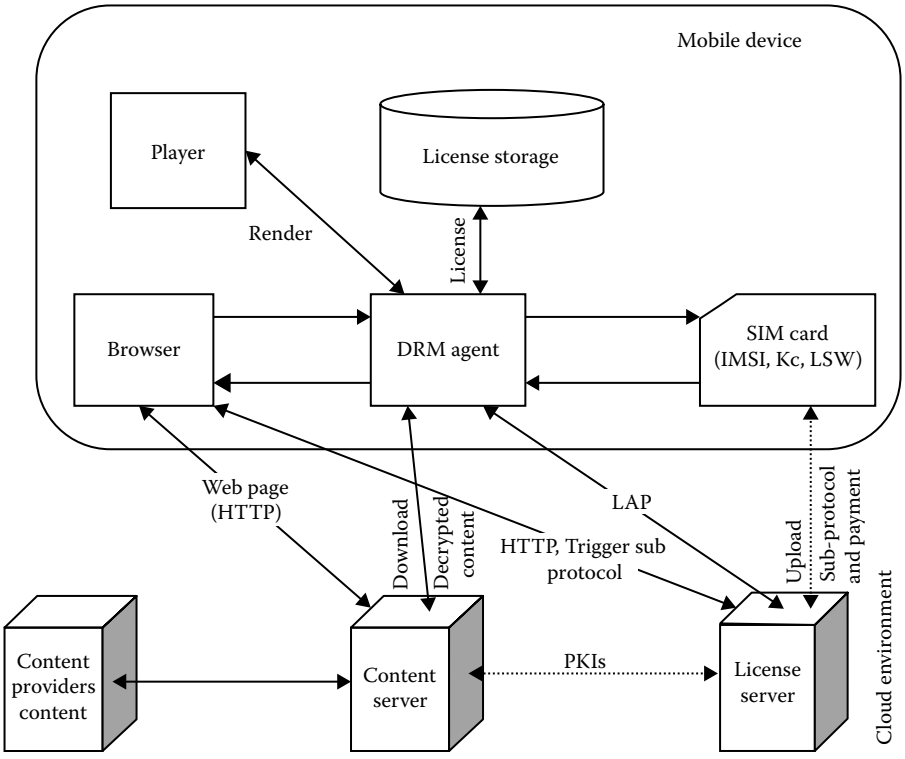


FIGURE 9.13
Architecture of digital right management.

play digital contents that cannot be illegally distributed. The CS-DRM browser is used to browse the website of the backend and also to notify the DRM agent of the next action according to a response or event. This scheme consists of five phases: preparation, rights management, license acquisition, play, and download/upload. The preparation phase initializes the backend, generates the keys for symmetric encryption, transfers content ID to licensed server, and so on. The rights management phase customizes digital contents, while the license acquisition phase acquires the license of that digital content from the license server. After getting the license of the digital content, the user decrypts the content to play. In the download phase, the user can change the device and download the license to his or her new device to enjoy the digital contents. Upload phase is used to guarantee the integrity of the digital content. Implementation of the CS-DRM scheme called phosphor, which shows that this scheme is efficient, secure, and practicable, is also discussed.

9.3.5 Intrusion Detection

Mobile devices are now called smartphones because they are used not only for phone calls but also for browsing, reading news, watching videos, and many more things, as done by PCs. With the increase in software complexities of smartphones, the number of bugs and exploitation of vulnerabilities also have increased. Smartphones use the same software architecture as PCs, and they are attacked by same class of viruses, worms, malware, and Trojan horses. These malware programs get into the smartphones through unofficial repositories and carry out their malicious activities. The compromised device can send the user's private data and call logs, premium messages, financial transactions, location, and so on. To prevent these attacks, the best way is to install an antivirus such as AVG-Mobilation, Kaspersky, or Avastetc on the smartphones for intrusion detection.

9.3.5.1 Drawbacks of Intrusion Detection

Intrusion detection has a number of drawbacks such as storage requirements, the need for larger CPUs, and battery consumption. Since the antivirus software is based on loading signatures that require storage, and as they run on mobile instruments, they draw a large amount of CPU and battery power. Smartphones have limited resources, so such solutions are not efficient for these devices. Hence, some lightweight intrusion detection schemes are proposed by researchers, which are based on the integration of the mobile device and cloud computing.

9.3.5.2 Existing Schemes for Intrusion Detection

There are two approaches used for detecting malware in mobile devices. The first is the static approach, which uses an antivirus installed on the smartphones with the signature of malwares database looking for suspicious patterns. Second is the dynamic approach, in which behavior-based detection is involved and runs in a controlled and isolated environment for detecting any malicious codes. In this chapter, we discuss mostly the dynamic approach for intrusion detection of mobile devices using cloud servers.

9.3.5.2.1 Crowdroid

Behavior-based malware detection for Android phones, called Crowdroid, is a lightweight client application that is downloaded from Google store and installed in the device [25]. Crowdroid is used to send all the preprocessed system calls to a central server. Data acquisition, data manipulation, and malware analysis and detection are the three components of this architecture. The data manipulator parses the received data and creates a system call vector for each interaction; hence, a dataset is prepared according to its behavior. These datasets are then clustered for determining benign application or malware application with the help of a behavior-based malware detection server. Experiments show that clustering the results enables successful detection of self-written and real malware. The full architecture is given in Figure 9.14.

A malware detection scheme using a lightweight host agent that runs on mobile devices and an off-device network service that runs on cloud and receives files from the agent for detection have been proposed [26]. Host agents trap the file, divert it to the handler routine, and generate a unique ID by hashing the file. This ID is then compared with the cached ID,

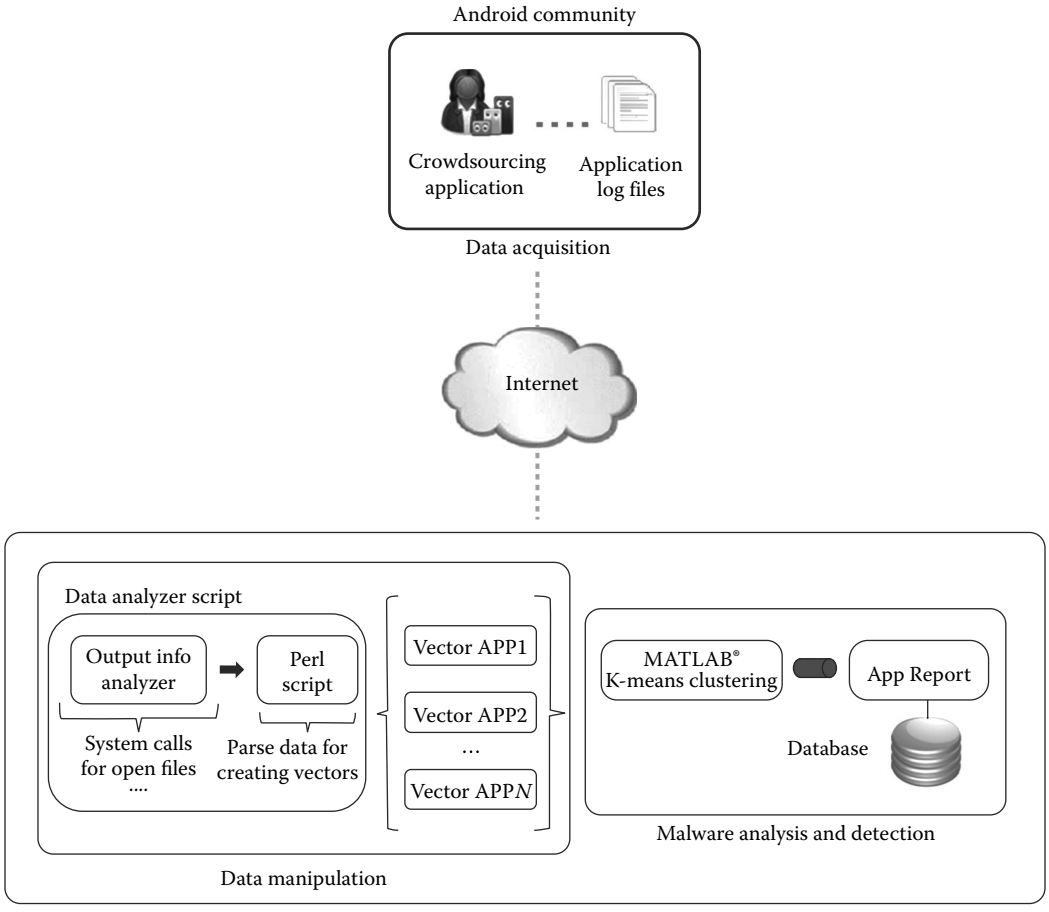


FIGURE 9.14
Crowdroid architecture.

and if the ID is not present, it is sent to the cloud for analysis. Network service receives the file sent by the host agent and determines whether the file is malicious or not. The network service consists of a number of antivirus engines running in parallel on a virtual machine on the cloud server. This scheme provides SMS spam filtering, phishing detection, and a centralized blacklist to provide security to mobile devices.

9.3.5.2.2 Paranoid Android

Paranoid Android (PA) is a versatile security, which offers a wide range of security measures for smartphones [27]. A prototype is also implemented. Called PA for intrusion detection, it is scalable and flexible. In this scheme, a synchronized replica of the phone runs on the cloud security server. The tracer on the phone mainly records the data between the kernel and user space employing system calls and replays on the replica. The full architecture is given in Figure 9.15.

9.3.5.2.2.1 Architecture

1. The tracer records all information needed to accurately replay its execution.
2. The replayer receives the trace and faithfully replays the execution within the emulator.
3. The proxy intercepts and temporarily stores inbound traffic, and the replayer can access the proxy to retrieve the data needed for replaying.

The recorded information is transmitted to the replica using the proxy server. The cloud replays the tracer information on the concurrent replica, and if attack is detected, PA needs to warn the user about the threat. When the user receives the notification of an attack, it starts a recovery process using the data stored in the replica and brings the device back to a safe or clean state.

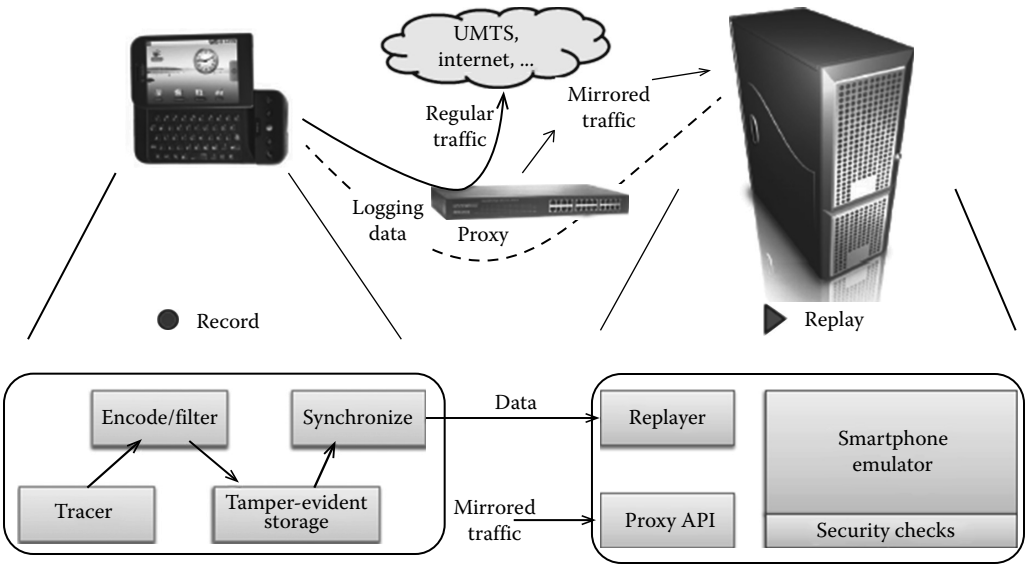


FIGURE 9.15
Paranoid Android architecture.

A cloud-based intrusion detection system for smartphones has been proposed in which a response engine continuously performs in-depth and forensic analysis of smartphones to detect any misbehavior [28]. The response engine will take appropriate action if any misbehavior is detected. In this scheme, the smartphone must be registered with the intrusion detection system. After registration, the smartphone has to provide device information such as the company name, model number, operating system, software, applications, and so on. The emulator, after receiving all the information about the device, creates a replica of the same smartphone on the cloud. The cloud provides a lightweight client agent to be installed on the device for proxy setting. The proxy server duplicates the incoming and outgoing traffic. The data are then provided to the emulated platform for detection. The client agent gathers all input and sensed data of the device, sends them to emulated replica, and waits for the reply. When any misbehavior is detected, the client is notified and appropriate action is taken to bring back the system to the normal condition. In this, a protocol is prepared for the intrusion analysis engine for Linux kernel, which uses two sources of information, namely a set of intrusion detection systems and system call tables.

9.3.5.2.3 Secloud

Zonouz et al. [29] extended the work done in Houmansadr et al. [28], and a prototype called Secloud was proposed and implemented as a comprehensive security solution for Android smartphones. Secloud is a powerful intrusion detection scheme that uses the cloud to protect smartphones. Secloud uses fewer resources of the mobile device and provides real-time security as a service to the smartphone, as shown in Figure 9.16. Secloud is kept synchronized to the actual device, and if any intrusion is detected, the emulated replica sends a notification to take appropriate action to the threat. The practical deployment of Secloud takes care of the file consistency by hashing the folders, and only those hash values, that are not present in the device to save resources, are sent. User privacy, encryption, and alternative ways for notification are used to make the system robust.

The encryption method is used by Secloud to encrypt and send credential data to replica, where these data are decrypted and analyzed. Experiments show that Secloud accurately detects known and unknown threats. It is also efficient in CPU and memory utilization.

In Table 9.5, different existing intrusion detection schemes are compared according to the type of intrusion detection, platforms, prototypes, and features.

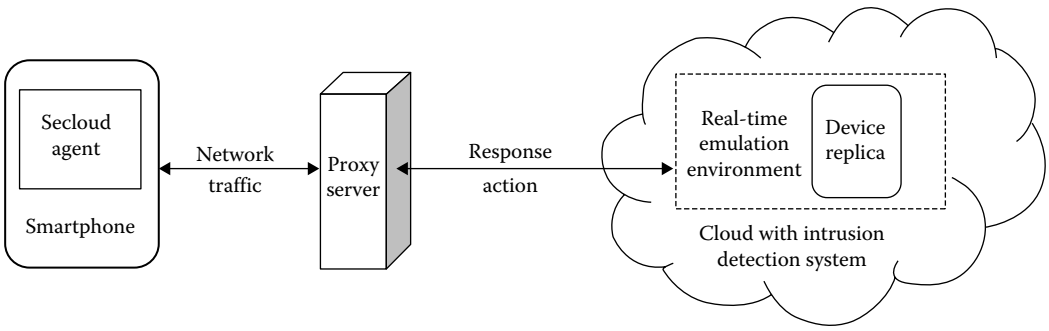


FIGURE 9.16
Secloud architecture for intrusion detection.

TABLE 9.5

Comparison of Different Intrusion Detection Systems

Authors	Intrusion Detection Approach	Platform	Prototype	Other Features
Burguera et al. [25]	Dynamic	Android OS	Crowdroid	Lightweight, scalable
Oberheide et al. [26]	Dynamic	Android OS	CloudAV (Mobile version)	Less complex, energy efficient, scalable
Portokalidis et al. [27]	Dynamic	Android OS	Paranoid Android	Highly scalable and flexible, energy efficient
Houmansadr et al. [28]	Dynamic	Android OS	Secloud with minimum function	Energy efficient, scalable
Zonouz et al. [29]	Dynamic	Android OS	Secloud with full function	Powerful, energy efficient, secure

9.4 Conclusion

Cloud computing is an emerging field of wireless network, and integrating it with handheld devices solves numerous issues such as storage, battery power, and computational processes, but raises a number of security threats both in the cloud and in the devices. These threats are mostly a combination of mobile device threats, communication threats, and cloud environment threats. Different types of framework related to security issues in mobile cloud computing have been explained in this chapter. The issues were related to ensuring privacy, authentication, security, trust, and so on, to data and applications that are offloaded to the cloud from mobile devices. Also discussed were how the mobile user authenticates them to cloud, how location-based service privacy is achieved, and how the real-time intrusion detection is achieved. Different comparison tables were used to analyze the features and drawbacks of the proposed schemes. It was observed that most of the frameworks offload processor-intensive computation jobs to the cloud because of the resource limitation of mobile devices. New security threats, which are due to lack of isolation among various virtual machine instances running on the same physical server, need to be handled. To provide security in the MCC environment, the cloud provider should ensure data security, network security, data locality, data integrity, application security, data segregation, data access, data breach issues, and various other factors.

Questions

1. Comment on security and privacy of mobile cloud computing.
2. Draw and explain the security models of mobile cloud computing.
3. Compare the different intrusion detection systems used in MCC.
4. What are the security issues in the mobile cloud computing environment?
5. What is digital right management in MCC?
6. What are the approaches to mitigate security issues related to mobile devices?

References

1. 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*, 13(18), 1587–1611, 2013.
2. A. N. Khan, M. L. M. Kiah, S. U. Khan, and S. A. Madani, Towards secure mobile cloud computing: A survey, *Future Generation Computer Systems*, 29(5), 1278–1299, 2013.
3. X. Lin, Survey on cloud based mobile security and a new framework for improvement, in *IEEE International Conference on Information and Automation*, Shenzhen, China, pp. 710–715, 2011.
4. F. S. Gharehchopogh, R. Rezaei, and I. Maleki, Mobile cloud computing: Security challenges for threats reduction, *International Journal of Scientific and Engineering Research*, 4(3), 8–14, 2013.
5. S. K. V. Ko, J. H. Lee, and S. W. Kim, Mobile cloud computing security considerations, *Journal of Security Engineering*, 9(2), 143–150, 2012.
6. S. Hui, Z. Liu, J. Wan, and K. Zhou, Security and privacy in mobile cloud computing, in *Ninth International Wireless Communications and Mobile Computing Conference*, Sardinia, Italy, pp. 655–659, 2013.
7. M. S. Morshed, M. M. Islam, M. K. Huq, M. S. Hossain, and M. A. Basher, Integration of wireless hand-held devices with the cloud architecture: Security and privacy issues, in *International Conference on Parallel, Grid, Cloud and Internet Computing*, Barcelona, Spain, pp. 83–88, 2011.
8. S. Resondry, K. Boudaoud, M. Kamel, Y. Bertrand, and M. Riveill. An alternative version of HTTPS to provide nonrepudiation security property: A flexible component-based approach for secured transactions in a mobile environment, in *Wireless Communications and Mobile Computing Conference (IWCMC), 2014 International*, August, Nicosia, Cyprus, pp. 536–541, 2014.
9. C. Kilinc, T. Booth, and K. Andersson, WallDroid: Cloud assisted virtualized application specific firewalls for the Android OS, in *IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications*, Liverpool, England, pp. 877–883, 2012.
10. L. Xu, L. Li, V. Nagarajan, D. Huang, and W. T. Tsai, Secure web referral services for mobile cloud computing, in *IEEE Seventh International Symposium on Service Oriented System Engineering*, Redwood City, CA, pp. 584–593, 2013.
11. D. Popa, M. Cremene, M. Borda, and K. Boudaoud, A security framework for mobile cloud applications, in *11th RoEduNet International Conference*, Sinaia, Romania, pp. 1–4, 2013.
12. R. Chow, M. Jakobsson, R. Masuoka, J. Molina, Y. Niu, E. Shi, and Z. Song, Authentication in the clouds: A framework and its application to mobile users, in *Proceedings of the ACM Workshop on Cloud Computing Security Workshop*, New York, pp. 1–6, 2010.
13. S. Grzonkowski, P. M. Corcoran, and T. Coughlin, Security analysis of authentication protocols for next-generation mobile and CE cloud services, in *IEEE International Conference on Consumer Electronics-Berlin*, Berlin, Germany, pp. 83–87, 2011.
14. K. Zhao, H. Jin, D. Zou, G. Chen, and W. Dai, Feasibility of deploying biometric encryption in mobile cloud computing, in *Eighth China Grid Annual Conference*, Changchun, China, pp. 28–33, 2013.
15. W. Itani, A. Kayssi, and A. Chehab, Energy-efficient incremental integrity for securing storage in mobile cloud computing, in *International Conference on Energy Aware Computing*, Cairo, Egypt, pp. 1–2, 2010.
16. S. C. Hsueh, J. Y. Lin, and M. Y. Lin, Secure cloud storage for convenient data archive of smart phones, in *IEEE 15th International Symposium on Consumer Electronics*, Singapore, pp. 156–161, 2011.
17. D. Huang, Z. Zhou, L. Xu, T. Xing, and Y. Zhong, Secure data processing framework for mobile cloud computing, in *IEEE Conference on Computer Communications Workshops*, Shanghai, China, pp. 614–618, 2011.

18. W. Ren, L. Yu, R. Gao, and F. Xiong, Lightweight and compromise resilient storage outsourcing with distributed secure accessibility in mobile cloud computing, *Tsinghua Science and Technology*, 16(5), 520–528, 2011.
19. A. N. Khan, M. M. Kiah, S. U. Khan, S. A. Madani, and A. R. Khan, A study of incremental cryptography for security schemes in mobile cloud computing environments, in *IEEE Symposium on Wireless Technology and Applications*, Kuching, Malaysia, pp. 62–67, 2013.
20. X. Liu, R. Jiang, and H. Kong, SSOP: Secure storage outsourcing protocols in mobile cloud computing, in *IEEE 14th International Conference on Communication Technology*, Chengdu, China, pp. 678–683, 2012.
21. Z. Zhou and D. Huang, Efficient and secure data storage operations for mobile cloud computing, in *Proceedings of the Eighth International Conference on Network and Service Management*, Laxenburg, Austria, pp. 37–45, 2012.
22. J. Shin, Y. Kim, W. Park, and C. Park, DFCloud: A TPM-based secure data access control method of cloud storage in mobile devices, in *IEEE Fourth International Conference on Cloud Computing Technology and Science*, Taipei, China, pp. 551–556, 2012.
23. W. Jia, H. Zhu, Z. Cao, L. Wei, and X. Lin, SDSM: A secure data service mechanism in mobile cloud computing, in *IEEE Conference on Computer Communications Workshops*, Shanghai, China, pp. 1060–1065, 2011.
24. C. Wang, P. ZouPeng, Z. Liu, and J. Wang, CS-DRM: A cloud-based SIM DRM scheme for mobile internet, *EURASIP Journal on Wireless Communications and Networking*, 14, 1–19, 2011.
25. I. Burguera, U. Zurutuza, and S. Nadjm-Tehrani, Crowdroid: Behavior-based malware detection system for Android, in *Proceedings of the First ACM Workshop on Security and Privacy in Smartphones and Mobile Devices*, New York, pp. 15–26, 2011.
26. 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*, New York, pp. 31–35, 2008.
27. G. Portokalidis, P. Homburg, K. Anagnostakis, and H. Bos, Paranoid Android: Versatile protection for smartphones, in *Proceedings of the 26th Annual Computer Security Applications Conference*, New York, pp. 347–356, 2010.
28. A. Houmansadr, S. A. Zonouz, and R. Berthier, A cloud-based intrusion detection and response system for mobile phones, in *IEEE/IFIP 41st International Conference on Dependable Systems and Networks Workshops*, Hong Kong, China, pp. 31–32, 2011.
29. S. Zonouz, A. Houmansadr, R. Berthier, N. Borisov, and W. Sanders, Secloud: A cloud-based comprehensive and lightweight security solution for smartphones, *Computers and Security*, 37, 215–227, 2013.