Chapter 1
Security Frameworks in Mobile Cloud Computing

Chaitanya Vemulapalli, Sanjay Kumar Madria, and Mark Linderman

Abstract The concept of mobile cloud computing (MCC) combines mobile computing with cloud resources, and therefore, has opened up new directions in the field of mobile computing. Cloud resources can help in overcoming the memory, energy, and other computing resource limitations of mobile devices. Thus, the mobile cloud computing applications can address some of the resource constraint issues by offloading tasks to cloud servers. Despite these advantages, mobile cloud computing is still not widely adopted due to various challenges associated with security in mobile cloud computing framework including issues of privacy, access control, service level agreements, interoperability, charging model, etc. In this chapter, we focus on the challenges associated with security in mobile cloud computing, and key features required in a security framework for MCC. Initially, we describe key architectures pertaining to various applications of mobile cloud computing, and later, we discuss few security frameworks proposed for MCC in terms of handling privacy, security, and attacks.

Keywords Mobile cloud computing · Security · Mobile computing · Location Based services

1 Introduction

Mobile computing is becoming part of everyday life with wireless communication becoming ubiquitous. The technological advancement in mobile devices and invention of smart phones has taken the usage of mobile phones from the conventional use of voice communication to more now as the computing device. Incorporation

C. Vemulapalli · S. K. Madria (✉)
Missouri University of Science and Technology, Rolla, MO, USA
e-mail: sv2v7@mst.edu; madrias@mst.edu

M. Linderman
AFRL, Information Directorate, Rome, NY, USA
e-mail: mark.linderman@us.af.mil

B. B. Gupta et al. (eds.), *Handbook of Computer Networks and Cyber Security*,
https://doi.org/10.1007/978-3-030-22277-2_1

of sophisticated features like in-built camera, GPS, multimedia capabilities, etc. gave additional functionalities to the mobile devices. The range of functions that can be performed by mobile devices is the main driving force behind the growth of mobile computing. All these advanced features increase the software and processing overhead in mobile devices. Moreover, the advancement of software in mobile devices is happening at a more rapid pace compared to advancement in mobile hardware. Users are not able to fully exploit these advanced features due to hardware limitations of the mobile devices such as limited processing capabilities, insufficient storage, limited battery backup, etc.

In the last decade, with access to Internet becoming more and more ubiquitous, connecting to a cloud server via Internet from a mobile device is no longer a difficult proposition. This stimulated a new idea of using cloud resources for the processing and storage requirements of mobile device and gave rise to the new computing paradigm of mobile cloud computing. In this paradigm, to overcome the above said hardware limitations of mobile devices, storage tasks, communication, and computation intensive tasks are offloaded to cloud servers instead of performing them in mobile devices itself. The mobile devices will retain only thin client for user interface or display of results. Examples of such thin clients include mobile apps like YouTube, Facebook, etc.

Usage of smart phones and mobile cloud computing is also increasing at a rapid pace. According to ABI Research (Allied Business Intelligence, Inc.), a market intelligence company, the number of mobile cloud computing subscribers worldwide grew from 42.8 million subscribers in 2008 to over 100 million in 2014 [13]. Another study by Juniper Research said that the market of cloud-based mobile applications grew by about 88% from $400 million in 2009 to $9.5 billion in 2014 [23]. It was reported that more than 240 million of mobile cloud computing (MCC) customers will use cloud services with an earning revenue of 5.2 billion dollars in 2015. Gartner forecasts that global mobile phone shipments will increase 1.6% in 2018, with total mobile phone sales amounting to almost 1.9 billion units. In 2019, it will grow by 5% year over year. This growth in mobile cloud computing has opened up the possibility of enhancing applications like location-based services, information sharing, etc.

Use of mobile cloud computing in disaster recovery and emergency service has also been described in [26]. Though the concept of using cloud resources has made the mobile computing more useful and empowered it to perform any task without limitations, the security and privacy issues associated with cloud computing are deterring the large scale adoption of mobile cloud computing applications. In despite of efforts devoted in research both in industry and in academia, there are a number of loopholes in the security policies of mobile cloud computing. According to surveys [2, 27], 74% of IT executives are not interested to adopt cloud services due to security issues and risks associated with it. Some secondary limitations like limited processing power, low storage are mentioned as obstacles for computationally intensive and storage demanding applications on a mobile platform. The major data security risks such as data loss, data breach, and data privacy result from the fact that mobile users' data is stored and processed in clouds that are located at the service providers' end.

Rest of this chapter is organized as follows: In Sect. 2, we initially describe some of the architectures of mobile cloud computing proposed by researchers. In Sect. 3, we discuss the importance of security in mobile cloud computing and the security aspects that are necessary in MCC. In Sect. 4, we provide a review of the security frameworks proposed in the literature for authentication, privacy, secure storage, and secure computing. In Sect. 5, we discuss attacks, risk assessment, and vulnerability in mobile clouds. Section 6 is the discussion section comparing different techniques. And, finally, Sect. 7 concludes this chapter.

2 Architecture of Mobile Cloud Computing

Since the demand for smartphones and tablets is constantly increasing, manufacturers of these devices are improving the technology and usability of devices. It is because of handy shape and size, mobile devices are being used to perform most tasks that a desktop or laptop computer is currently used for. These devices can also connect to the resources of cloud computing called mobile cloud computing (MCC) which has increased the challenges due to the security loopholes. Mobile cloud computing is relatively a new computing paradigm and the basic general idea behind an architecture of mobile cloud computing involves mobile devices, mobile network, and cloud servers. Mobile devices access the Internet using wireless network, and through the Internet communicate with the cloud servers.

Figure 1.1 depicts a general architecture of mobile cloud computing. Though this is a basic architecture of mobile cloud computing, various other versions of mobile cloud computing architectures have been proposed based on the applications. Many of the services available under conventional cloud computing are also available for mobile computing. These include Data storage as a Service (DaaS), Communication as a Service (CaaS), Security as a Service (SecaaS), Software as a Service (SaaS), etc.

One of the main applications of mobile cloud computing already being widely used is data storage in cloud. Here, authorized users are allocated storage space in the cloud servers. In this architecture, data files such as images, videos, and other personal files are uploaded to the cloud server to overcome the storage limitations in mobile devices and give the flexibility of accessing the files anytime and from anywhere. Mobile apps like Dropbox, iCloud, SkyDrive, etc. are few such examples.

Another important type of usage or application of mobile cloud computing is where communication and computation are offloaded to the cloud. Figure 1.2 gives a pictorial representation of the architecture associated with this type of service/application of MCC. In this architecture, virtual smart phone devices are setup in the cloud to which the physical devices can connect and offload their tasks. These are called by different names such as virtual images [3], extended semi shadow images (ESSI) [11], etc., by different authors but the underlying idea is the same, i.e., having virtual machines in the cloud server. In this chapter, we use the term virtual image to describe these virtual machines in the cloud. These virtual images can be full or partial images. Virtual images are free of any physical

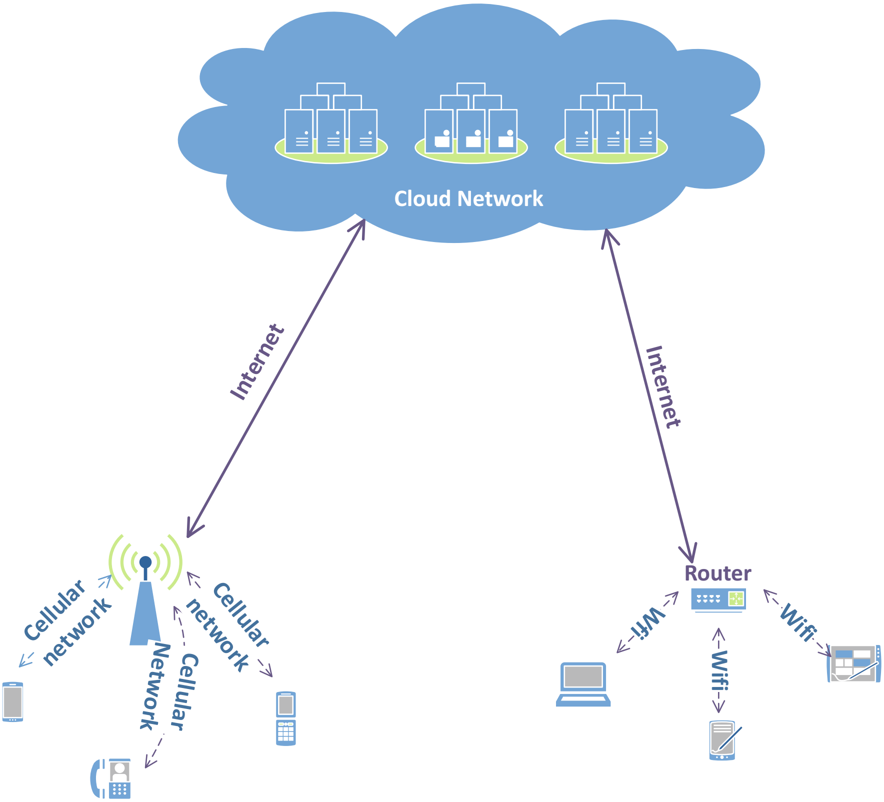

Fig. 1.1 General architecture of mobile cloud computing

limitations that are synonymous with physical mobile devices such as limited battery power and limited processing capabilities. Moreover, users can allocate/configure these virtual images as per requirement. Mobile devices connect to their respective virtual images in the cloud through the Internet from available wireless networks. Mobile devices connect to the nearby access points through wireless communication and access points are connected to cloud servers via the Internet in various ways with fixed network used at some point in the network. The mobile devices are connected to virtual images in the cloud using a secure communication channel through the Internet. The two main operations that result in high battery consumption in a mobile device are computing and communication tasks. The mobile devices can offload high CPU consumption tasks to the virtual image in cloud since it possesses more powerful computing resources and no battery limitation. Similarly, communication among physical mobile devices is affected by many factors such as mobility, range, battery power, and other environmental factors. Offloading the communication tasks to their virtual counter parts in the cloud can help in overcoming these factors since virtual images are fully connected and do not possess any battery limitations.

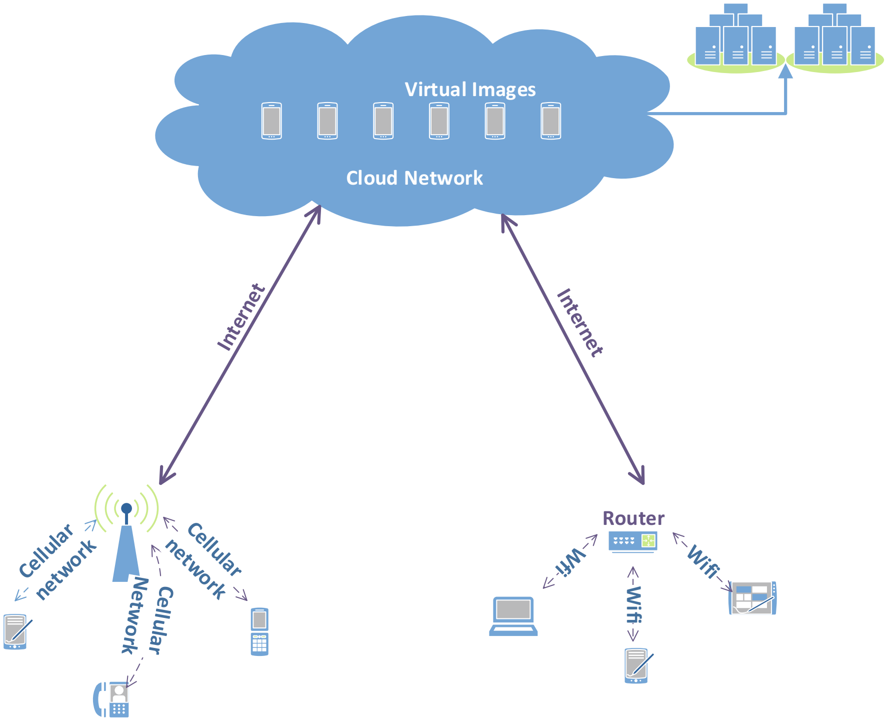

Fig. 1.2 Architecture of mobile cloud computing with virtual images

Olafare et al. [20] performed a research focused on the security challenges and possible solutions in MCC. They proposed the adoption of applications on the mobile device which keep a check on amount of information third-party applications can have access to. In addition, the validity and authenticity of third-party application needs to be checked before installing. Also, the third-party application signature or certificate needs to be checked in order to ensure that the updated version signature matches the original signature of the third-party application. The authors further discussed MCC models/architectures with security components to counter attacks. It is also being suggested to use SSL certificate for the security of communication channel. Without using SSL, it is easy for an attacker to bridge the data transmission and act as a cloud server to tamper the data. Using SSL, when the user starts using the cloud services, data sent to the user is an SSL encrypted data. The key for decryption of the data is sent to the user over a personal email account. The authors then classified the security threats into three major categories: mobile device threats, threats to the cloud (cloud computing), and network threats. For each category of security issues that are related to MCC, the author has designed a framework/architecture with security components.

3 Security Aspects of Mobile Cloud Computing

Most of the applications of mobile cloud computing involve exchange of data with cloud servers which are beyond the control of mobile users. This information may also include private data of users such as his location, usage details, etc. So it is very important to protect this user information from adversary. Since cloud provider is also a third party, it can also be considered as a potential adversary. The security requirements in MCC may slightly vary with the application but the basic and mandatory aspects of security in mobile cloud computing would be (1) authentication, (2) data integrity and confidentiality, and (3) privacy.

Authentication In mobile cloud computing, mobile users utilize the cloud resources for their storage needs, offloading computation and communication tasks. Since cloud servers will be used by number of users, there should be an authentication mechanism between mobile users and the cloud. In another architecture of MCC mentioned earlier, virtualization is used and virtual images are maintained in the cloud. This architecture requires added authentication mechanism between virtual images.

Data Integrity and Confidentiality One of the main applications of cloud computing is to use cloud resources for storing users' data. This is one of the major advantages of mobile cloud computing. Usually, mobile devices have limited storage capacity. In order to overcome this limitation, files are offloaded to the cloud servers so that they can be accessed from anywhere and at any time. But the cloud servers are not in the control of mobile users, and hence, cloud service providers could also be potential adversary. Therefore, efficient encryption mechanisms must be in place to preserve the confidentiality and integrity of the files stored in the cloud servers. Moreover, there should be provision for users to verify the integrity of files at any instant of time.

Privacy In mobile cloud computing, mobile users constantly communicate with cloud servers to access their resources. In this process, privacy of the mobile user needs to be protected from the cloud service provider as well. In some applications like location-based services using mobile cloud computing, this is more important as the user location information should be protected from other entities.

4 Security Frameworks for Mobile Cloud Computing

4.1 Authentication Frameworks for Mobile Cloud Computing

A Framework of Authentication in the Cloud for Mobile Users In the paper [7], the authors address the issue of device authentication in mobile cloud computing using policy based authentication. The proposed scheme uses the implicit authentication and trustcube. Unlike traditional authentication mechanisms which are based on aspects like what you have, what you know, and what you are, implicit authentication is based on what you do. By this users are identified by their habits, as opposed to their belongings, memorized data, and biometrics. Implicit authentication can be implemented in various ways like IP address, device profiles, etc. However, in this scheme, they use implicit authentication based on mobile data such as calling patterns, short messages (SMS) activity, website accesses, and location information which is automatically available with the network operators/carriers. This kind of implicit authentication gives an added security by protecting against unwanted access from stolen handsets. Implicit authentication is a statistical test and works based on comparison with threshold values. Based on the observed behavior of the users with mobile data, probabilistic authentication scores are calculated and assigned to client devices. The proposed authentication framework compares the calculated authentic score with the threshold values to verify whether the device is with legitimate user or not. The threshold value and amount of uncertainty allowed is dependent on the type of the application.

Figure 1.3 depicts the block diagram of the proposed framework. It has four main components: (a) client device, (b) data aggregator, (c) authentication engine, (d) authentication consumer. Client devices are the mobile devices on which the user performs his daily actions. The data aggregator constantly collects data on

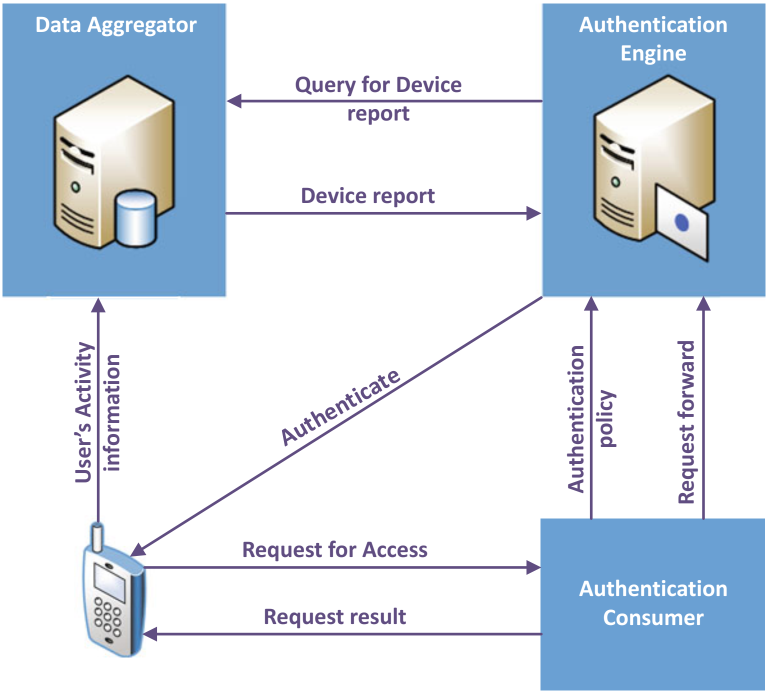

Fig. 1.3 Main components of the MCC framework and their interactions

context and action from the client devices. The authentication engine will obtain this information from data aggregator or from client device directly and the corresponding authentication policies from authentication consumer. Based on the results from the authentication engine, the authentication consumer responds to the clients' request.

Prior to the authentication process, authentication consumer prepares the list of access requests that require authentication. A policy is determined for each of the request and registered with the authentication engine. Each policy consists of at least three parts: the access request, the information to be collected from the client devices or data aggregator for this access request, and a policy rule. The policy rules consist of integrity check rules on the platform and environment, a threshold value for the authentication score, and the alternate authentication method if the authentication score is less than a threshold value. After the policy is registered with the authentication engine, when the authentication consumer receives an access request, it redirects the request to the authentication engine. Authentication engine obtains the required client info from the data aggregator or the client itself and then applies the authentication rule in the policy and determines the authentication result and sends this back to the authentication consumer. If the authentication result is successful, the authentication consumer will service the request. The proposed framework can also be scaled to large number of users by using multiple instances of authentication service within the cloud on demand.

Feasibility of Deploying Biometric Encryption in MCC In the work [31], Zhao et al. proposed an authentication framework for mobile cloud environment using biometric encryption (BE). Biometric encryption is more reliable compared to conventional security systems based on secret key due to its features that are difficult to forget, lose, share, and forge. The science of using physiological or behavioral features of human such as fingerprint, iris, face, signature, voice, etc. to identify him or her is called biometric identification. Combination of this biometric identification and cryptography is called biometric encryption. It combines biometrics and secret key, and they cannot be achieved in the templates stored in the system. Only when a living biometric feature was proposed to the system, the secret key would be generated. There are three encryption system models based on biometric encryption. First is the key release model in which the biometric feature and secret key are superposed to be the biometric feature template. Secret key is released only when the biometric feature matches. Second is the key binding model in which biometric feature and key materials are combined to be the biometric feature templates in encryption scheme. Third model is the key generation model in which secret key is extracted directly from the signal instead of from the external input.

The architecture of the proposed framework is depicted in Fig. 1.4. In the proposed framework, a separate cloud authentication center (CAC) is established to relieve the application server from the burden of analyzing and verifying requests from users. CAC is assumed to be a trusted party. Initially, BE application developers register their products in the platform when they are released. This informs the required parameters, including the category of the application, biometric

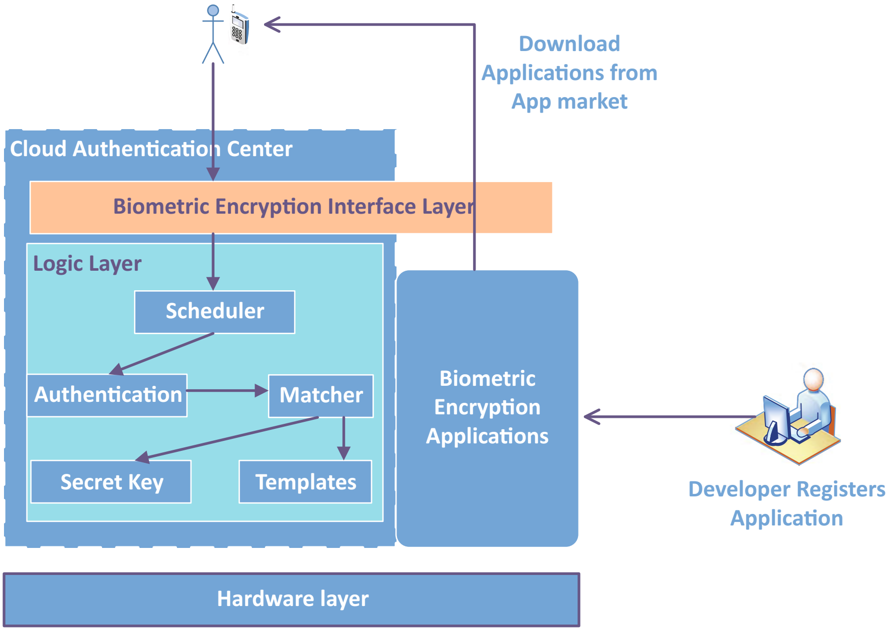

Fig. 1.4 Mobile cloud platform architecture for authentication using biometric encryption

features requested, security level, etc. These applications are downloaded by the users from app repository in the platform. Before a user can begin using the applications, a record containing his biometric features must be created on the platform and this is done through a specific interface. Application accomplishes this by calling BE module on the mobile device. CAC is the core component for the architecture. It schedules requests from clients, matches the submitted biometric data with the original ones, and also manages the biometric feature templates and secret keys. It makes the authorization for all the applications and users. Overall, the CAC analyzes the biometric data sent by applications and sends the result to application servers.

A Framework for Secure Mobile Cloud Computing The authors of this paper [25] discuss the use of biometric authentication framework to access the cloud. Biometric authentication supplies a bigger measure of protection and accuracy compared to other authentication methods with low hardware costs and secure entry. Biometric is the most effective method to authenticate the users and to protect them from illegal and unauthorized customers. The preprocessing steps and algorithms for extracting the features, and matching of the biometrics traits are discussed in detail. The authentication of fingerprint password is done over web-based services within cloud computing. The two phases discussed are biometric authentication framework enrollment and verification. The matching algorithm steps include comparing the input images with the template images. Template images are collected during the enrollment which are then compared with input images during the recognition phase. This phase decides if the input image and template image match or not. The authors proposed a novel matching score algorithm for considering features of biometrics. It is a combination of strong and weak classifiers which combines the matching scores of each subsystem to find multiple matching scores which are then sent to the decision phase. In this algorithm, the weak classifier is called for each iteration in order to generate a weak ranking. The matching algorithm decides to underline diverse parts of the training data. Hence, it was concluded that biometric authentication is the most effective authentication method as the fingerprints are unique.

Middleware Layer for Authenticating Mobile Consumers of Amazon S3 Data In [18], Lomotey and Deters proposed an authentication framework for mobile consumers of Amazon Simple Storage Service (Amazon S3) based on middleware oriented framework called MiLAMob and OAuth 2.0. Usually, to access Amazon S3, users have to provide credentials such as access key, secret access key, and a signature which is not very efficient for mobile environment as it contributes to HTTP traffic in request response architecture. Generating the hash message authentication code (HMAC) signature in mobile device also contributes to the computation overhead. Moreover, storing an access key Id, secret access key, and HMAC signature in mobile device is another security issue since the device can fall into wrong hands at any moment. The proposed framework overcomes these issues by introducing a middleware which handles the security and data request issues with Amazon S3 on behalf of the user. Architecture of the proposed framework shown in [18] is illustrated in Fig. 1.5.

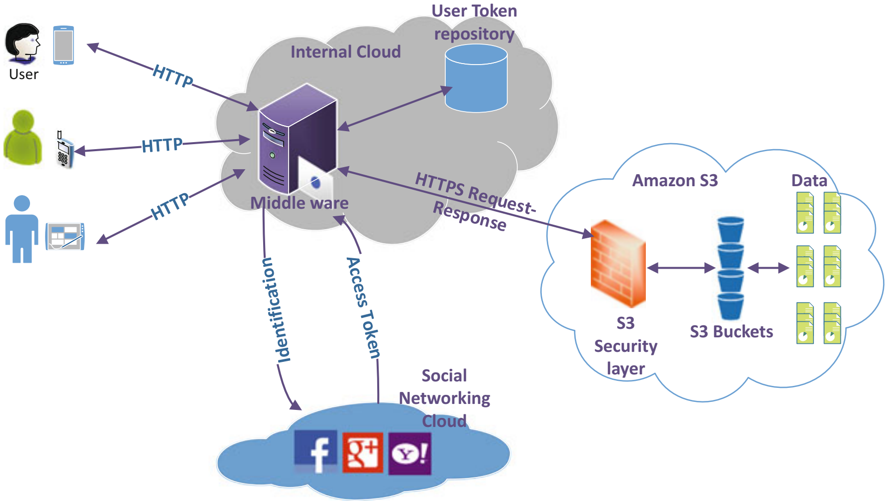

Fig. 1.5 Framework for authentication using middleware-layer for MCC

The proposed MiLAMob framework contains four major components. They are (1) mobile platform, (2) middleware, (3) the social networking platform, and (4) Amazon S3. For mobile platform, the framework advocates usage of mobile web frameworks approach rather than native approach mainly for the reason that it allows users to use heterogeneous mobile devices rather than being confined to a single mobile provider. The middleware is core of MiLAMob framework with three interfaces connected to mobile participants, social networking cloud, and Amazon S3. When a user wants to access Amazon S3, he/she first connects to middleware through publicly available URI. Middleware redirects the user to an authentication page where the user can chose a preferred authentication method. It could be either a personal login or through available social media like Facebook login, Google login, etc. If the user chooses to authenticate using Google credentials, then he is redirected to Google login page where he enters his id and password. After successful authentication, the middleware receives the users' security tokens and based on that it retrieves the user's Amazon S3 security credentials from its repository. The middleware then sends the request over HTTPS to Amazon S3 authentication system. If the request passes the authentication test, middleware retrieves the requested object and sends it to the mobile user. In this mechanism, user only interacts with middleware or social network media and Amazon S3 component is hidden from the user. Mobile users have no knowledge about Amazon S3 security tokens. Due to this, unauthorized use of system is prevented to some extent. Though this middleware component can be hosted on any public domain cloud, this paper advocates to host it on a private cloud to have full control of security issues. Incorporating authentication using social network media is the distinguishing feature of MiLAMob framework and it facilitates business-to-business (B2B) and business-to-consumer (B2C) support. Thus, by allowing user to authenticate using personal login or social network media, MiLAMob framework facilitates what is referred to as hybrid authentication mechanism.

Context Awareness Architecture in MCC Most of the authentication frameworks try to authenticate the device rather than the actual user and device may be lost or go into wrong hands very easily. This is an important issue when it comes to mobile devices. In order to overcome this issue, in [32], Zhou et al. proposed an authentication framework based on context aware data. Context aware data includes phone records, calendar, GPS applications, and battery data. Most of the other implicit authentication frameworks previously proposed take only time factor into consideration and does not take the periodic activities into consideration. The proposed context awareness architecture (CAA) in mobile cloud computing proposed in [32] is illustrated in Fig. 1.6.

The proposed CAA architecture primarily consists of three entities, namely the mobile client, cloud services, and CAA protocol. The mobile client/device has context aware data for mobile devices and corresponding protocol as the two major components. Decision-making device calculates the similarity of users recent behavior and activities with respect to the context awareness algorithm and then compares with the data in users characteristics database. It then passes the calculated

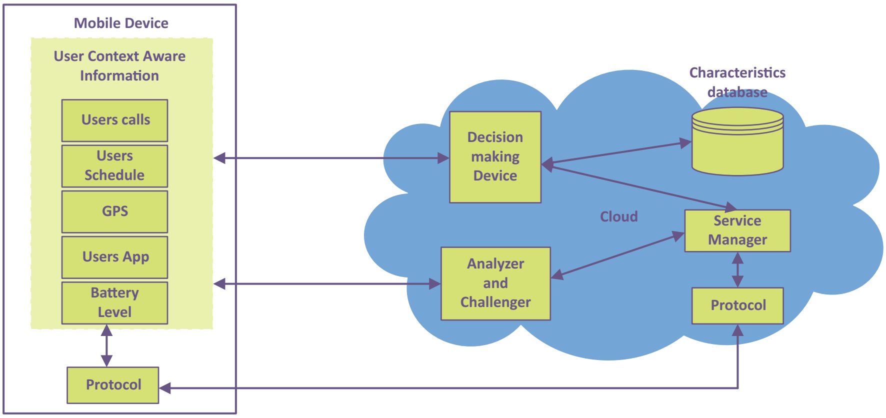

Fig. 1.6 Context aware architecture for authentication in MCC

similarity to the service manager. Service manager is the main component of this architecture. Based on the results provided by decision-making device, the service manager decides whether to allow the users to use the resources or to throw challenges through the Analyzer and Challenger. It also performs the task for formulating and implementing the new protocol and deciding whether to take the users frequent activities into users characteristics database. Data received by Analyzer and Challenger is divided into three kinds: high risk, medium danger, and low risk. If the received data is completely different from the one in characteristics database, then it is considered as high risk and the user is asked to enter a PIN code. If the user fails to enter the correct PIN code, he is denied access to the resources. In the medium danger condition, the user is asked to enter date of birth or a special phone number. In the low risk case, the user need not enter any further information for authentication and this reduces the explicit input of data. The user context data accepted by service manager as that of correct users is stored in users characteristic database for future authentication.

Consolidated Identity Management System for Secure MCC Security is the major obstacle while using the cloud server. In the survey conducted by the authors [14], it was noticed that more than 66% of the users tend to store personal identifiable information (PII) in unprotected text files, cookies, or applications. Mobile devices could be lost or stolen and compromised. These facts related to mobile devices make them attractive targets to obtain unauthorized access by intruders. In order to support the legitimate access process over the clouds, third-party identity management systems (IDMs) have been proposed. The access management systems depend on IDMs for identity generation, authentication, and authorization. However, IDMs are vulnerable to attacks which lead authors to introduce new IDM architecture dubbed consolidated IDM (CIDM) which countermeasures these vulnerabilities. It includes separating the credentials and distributing them over all the IDMs, adding second layer of authentication by allowing user to respond to human-based challenge–response and securing the communication link among cloud service provider and CIDM. A set of experiments were conducted over the IDMs and CIDMs and it was observed that the security provided by CIDM outperforms compared to the security provided by the current IDM systems. Also, it has less energy and communication overhead compared to the current IDM systems.

Identity Management Protocol for Secure MCC With increase in the use of mobile cloud computing, there is an increase in number of applications provided by the SP (service providers) which is causing traffic overload problems. This needs excessive network maintenance, creating an imbalance between profit and investment. The increasing number of mobile users has also caused identity management problems, which according to authors can be solved by using improved IDM3G protocol along with an additional authentication management protocol. The requirements for IDs include not just clarity for users, but also support for multiple IDs and maintaining anonymity and privacy. Interoperability, efficient management, and certification management are discussed in [22] which are considered to be the key network issues. The proposed method maintains the mobile operators (MOs) and constructs a trusted base with cross certification between service providers and MOs. It depends on public key infrastructure (PKI) to enable mutual dependence-based communication and ID management by service providers. It uses IDM which reduces the authentication steps leading to improvement in mobile network bandwidth and availability. The IDM protocol also maximizes the load balancing to cope with social engineering attacks and to reduce network cost. It maintains transparency, confidentiality, and ID management in mobile network. The new method when compared to existing IDM3G has minimum MOs data throughput and overall network cost and improved MOs availability in mobile networks.

4.2 *Privacy Preserving Security Frameworks for MCC*

Security Framework of Group Location-Based MCC Chen et al. [5] proposed a scheme to preserve the identity of user accessing location-based services. They proposed a security scheme that uses location-based group scheduling service called *JOIN* [16] to address this security problem. The architecture of the proposed framework [5] is illustrated in Fig. 1.7.

The *JOIN* system has three main components: (a) mobile devices/mobile users, (b) JOIN server, and (c) cloud database. *JOIN* server stores user data, friends around mobile user, and also handles the authentication of users. On the other hand, location information, services, and information about devices are stored in cloud database. Initially, the user gets registered with the *JOIN* system to start using its services. The mobile device transmits user identification, password, group name, and a key (K_A) to $JOIN$ server for registration. The key (K_A) is generated by applying a hash function on the international mobile subscriber identity $(IMSI)$. $(K_A) = H(IMSI)$. The *JOIN* server stores this information and generates a

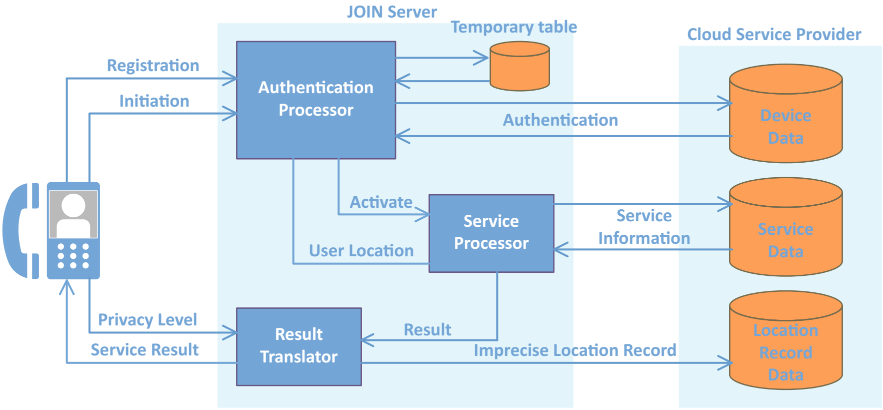

Fig. 1.7 Components of the proposed framework and their interactions

key (K_B) by using hash function on the concatenated string of ID and (K_A). $K_B = H(ID \| K_A)$. K_B and group name are then stored on cloud database for user authentication. When a user wants to use the *JOIN* services, he logs in using the ID and password. Then, in order to start an activity, the user transmits K_A, group name, and location information to *JOIN* server. Upon receiving this information, the *JOIN* server regenerates $K_B^{'}$ by hashing the ID and K_A used by the user. $K_B^{'} = H(ID \| K_A)$. This newly generated key is compared with the K_B stored in cloud database. Upon successful verification, a request is sent to all other members of the same group. All the group members respond with their respective K_A, group name, and location information. *JOIN* server authenticates each of these users as mentioned above and generates a list of friends using temporary table and list of points of interests using the cloud database near the mobile users location. This information is then encrypted using advanced standard algorithm (E) with initiator key (K_B). $C = E_{K_B}(Data)$ and transmitted to the initiator. The initiator then computes the key K_B using self ID and K_A and uses that to decrypt (D) the encrypted data (C), $Data = D_{K_B}(C)$. Thus, in this scheme the identity of user accessing the location-based system (LBS) is protected by applying hash function on the $IMSI$ to generate K_A.

In-Device Spatial Cloaking for Mobile User Privacy Assisted by the Cloud In the paper [28], Wang et al. proposed a framework to protect the privacy of mobile user in location-based services. The overall architecture of the proposed scheme is illustrated in Fig. 1.8.

In this scheme, the spatial space is hierarchically decomposed into h levels with each level having 4^h grid cells [1]. The entire system area is represented by the root at level zero. At each subsequent level, based on each grid cell in the upper level is subdivided into four child cells. In this scheme, two cells having same parent and residing in same row are termed as horizontal neighbors (C_H) and two cells having same parent and present in same column are termed as vertical neighbors (C_V).

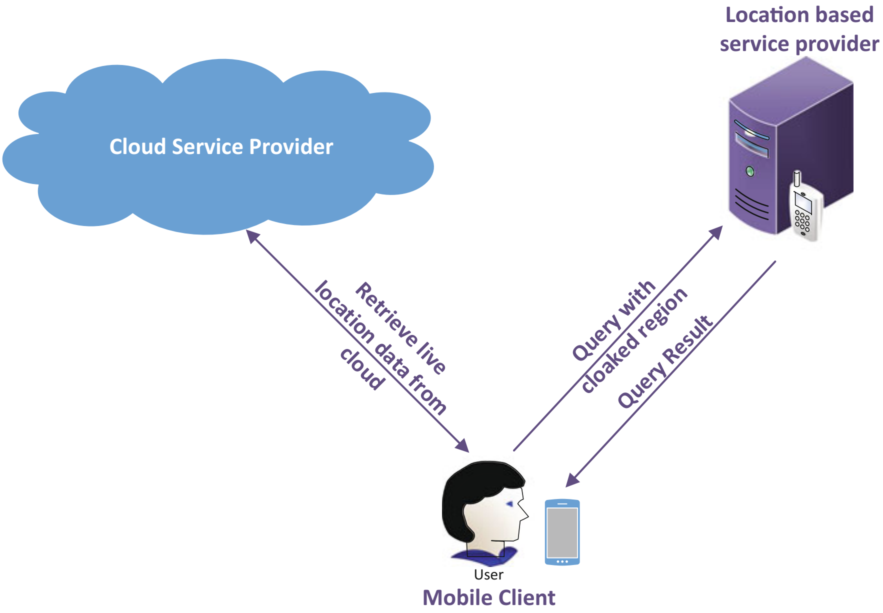
Location based service provider
Cloud Service Provider
Retrieve live location data from cloud
Query with cloaked region
Query Result
User
Mobile Client

Fig. 1.8 Architecture of privacy scheme for MCC

Historical data of each grid cell is maintained by the cloud service provider. When the mobile client wants to access LBS, it requests for the information on live users in four child grid cells corresponding to its current grid cell position (C). If it is less than the threshold (k), the number of live users is calculated according to the equation:

$$Sum(S) = Live\ No\ of\ Users\ in\ C_C + Live\ No\ of\ Users(C_H \| C_V)$$

where C_C is the number of live users in child grid cell containing the request issuer.

If this sum value is greater than threshold k, the generalized spatial region is generated based on C_C, and (C_H or C_V) anyone having a live number of user less than k. If S is less than k, the current grid cell is considered as a generalized spatial region. If the C_C containing the request issuer has more than k live users, C_C becomes the new current cell. The same process is repeated on the new current cell until the bottom level of grid is reached or the child grid cell is found having a live number of users less than k. The process of obtaining live users information from cloud service provider increases the latency and communication overhead. To reduce these affects, an optimizing cloaked algorithm is also proposed in this paper. This algorithm uses the historical information of live users in each grid cell stored in the cloud service provider. User location privacy is protected as the condition imposed on C_H or C_V provides anonymity to the request issuer's cell.

Protecting User Identity with Dynamic Credential and Mobility User identity verification is a crucial part of overall security of the system. Usually, in cloud computing identity verification is done using password or digital certificates which may be hacked by an adversary using some sophisticated techniques. In mobile environment this is a more serious threat as the mobile devices do not possess enough resources to run sophisticated security algorithms. In this paper [29], the authors proposed a dynamic credential generation mechanism for user identity verification. They propose to create a new kind of credential called dynamic credential as the identity proof by using the randomness in communication between cloud and user, like the mobility of the user. This dynamic user credentials change frequently based on the communication between cloud and the user. According to the authors, this dynamic credential provides more security than conventional credential management methods against attackers that can fake or steal the credential. In the first type of conventional credential management system such as using passwords to access the cloud, the credential does not change for a long time. So, if the attacker manages to steal the credential once, he will have access to the data for a very long time until the credential is changed again by the user. In another conventional credential management method, user is forced to change the credential periodically, for example, digital certificates. But even in this, once the attacker manages to fake or steal the credential, the time span would be long enough to launch an attack before user is forced to change the credential. On the other hand, in the proposed scheme, dynamic credential changes constantly based on user-cloud communications.

According to the proposed scheme, the messages exchanged between user and cloud are transformed into dynamic secrets. Users dynamic secret (S_U) is constantly updated by XOR operation of existing secret and the message (M_i) transmitted $(S_U = S_U \oplus M_i)$. Similarly, dynamic secret (S_C) of the cloud is also constantly updated by XOR operation with the message (M_i) transmitted. A packet counter N is updated with update of dynamic secrets $(N = N+1)$. A mobile user requests for a new data channel when he wants to start a new communication or when he changes the base stations. A threshold value denoted as $N_{threshold}$ is determined based on how frequently the user wants to change the credential. Each time the mobile user requests for new data channel from base station or number of packets exchanged reached a threshold, dynamic credentials (S) are updated. The dynamic credentials are updated as $S = S \oplus H(S_U \| S_C)$, where $\|$ is the concatenation operation. The values of S_U, S_C, and N are set to zero. Due to the constant change in credentials dynamically, the possibility of an attacker recovering user credential and using it successfully is very less.

Thus, any information loss will deprive the attacker of a valid credential at the time of launching the attack and this is the main strength of the proposed scheme. Unreliability of wireless communication with implicit information loss provides protection against attacker tapping on wireless signals. Similarly, the mobility of the user provides protection against attacker tapping on base stations as he has to predict user's movements and place tapping in every possible base station on user's path which is not practical. Even if the attacker manages to know the dynamic credential $S(t)$ for all t and performs spoofing or message injection attacks, it would be self-evident as it causes de-synchronization between user and cloud dynamic credentials. This scheme is also light weight and does not cause any overhead to mobile clients as it involves only bit-wise-XOR and hash functions.

Privacy Protection for Mobile Cloud Data Using Network Coding In this paper [6], the mobile data is collected and analyzed using big data analytics in order to understand and predict each individual behaviors. This information provides a great commercial potential for mobile cloud services. However, to keep the collected individual information secured is a new challenge. The huge computing power of intruders and the un-trustedness of cloud servers are pronounced to be the primary reasons of security breach. Using current security techniques has proved loopholes which will lead to a number of new challenges in protecting mobile privacy. In order to defend against malicious attackers with huge computing power in outsourced database (ODB), the authors proposed an unconditionally secure network coding based pseudonym scheme. The authors did a background study of other privacy methods, namely location privacy protection and system security model for group LBS using ODB. Though these methods may seem very effective, the hackers still have succeeded in breaking the security provided by aforementioned methods. The international mobile subscriber identity (IMSI) based group security (IGS) algorithm is further discussed in detail. The privacy analysis gives a solid proof of how the proposed scheme is unconditionally secure and it can simultaneously defend against attackers from both outside and inside. The results discussed show that the proposed network coding not only exhibits better delay performance, but also provides lower energy consumption compared to other methods.

The authors of this paper suggested an enhanced secure pseudonym scheme to protect the privacy of mobile cloud data, unconditionally secure lightweight network coding pseudonym scheme to face the huge computing power challenge, and two-tier network coding to solve privacy issue of untrusted cloud server issue. The international mobile subscriber identity (IMSI) based group security(IGS) algorithm is further discussed in detail. A two-tier coding includes generating Key_A for authenticating a legal customer's identity, and Key_B (pseudonym) for protecting customer's private data. If Key_A is certificated, the server generates Key_B using Key_A as input which then activates the login processor. The general information flows of privacy preserved LBS are described in this paper. The privacy analysis gives a solid proof of how the proposed scheme is unconditionally secure and it can simultaneously defend against attackers from both outside and inside. The results discussed show that the proposed network coding not only exhibits better delay performance, but also provides lower energy consumption compared to other methods.

4.3 Secure Data Storage Frameworks for MCC

Secure Data Service Mechanism in MCC The secure data service scheme proposed by Jia et al. [13] outsources data and security management to cloud. In this scheme, users have the flexibility to move the data and data sharing overhead to cloud without any information disclosure. Their scheme consists of three main entities, namely data owner, data sharer, and cloud service provider (CSP). Secure data service is achieved by using identity based encryption and proxy re-encryption. Identity based encryption is based on bilinear mapping.

$$e : G_1 \times G_1 \rightarrow G_T \tag{1.1}$$

The above equation defines a bilinear equation having bilinearity, computability, and non-degeneracy properties. Here, G_1 and G_T are the multiplicative cyclic groups with prime order q and g is the generator of G_1. This scheme uses two hash functions, which are

$$H_1 : \{0, 1\}^* \rightarrow G_1, \tag{1.2}$$

$$H_2 : G_T \rightarrow G_1 \tag{1.3}$$

In the proxy re-encryption scheme, a semi-trusted proxy transforms ciphertext encrypted with owner public key into another ciphertext encrypted with requester public key.

The proposed scheme consists of six algorithms used in each of the six phases. The six phases are setup phase, key generation phase, encryption phase, re-encryption key generation phase, re-encryption phase, and decryption phase. Each stage is explained below:

Setup Phase—$Setup(1^\lambda)$ This is the first stage of the scheme where master secret key (MSK) and system parameters are generated based on the given security parameter (λ). System parameters (P_{sys}) include G_1, G_T, g, g_s, and $MSK = s$. The system parameters are public and they are distributed among all users, whereas the master secret key is kept private and known only to the authority.

Key Generation Phase—$KeyGen(ID_O, P_{sys}, MSK)$ In this phase, the mobile users register with the system and obtain a secret key SK. This is generated to the users based on their identity (ID_O) using MSK and H_1, where (ID_O) is user identity of the data owner.

$$SK_{ID_O} = H_1(ID_O)^s, where\ ID_O \in \{0, 1\}^* \tag{1.4}$$

Encryption Phase—$Encrypt(P_{sys}, ID_O, m)$ In the encryption phase, the data file F is divided into n chunks as $F = (m_1, m_2, \ldots m_n)$. Encrypt algorithm is run for each chunk m_i and $M_i = (g_r, m_i.e(g^s, H_1((ID_O)^r))$, where $r \in Z_q^*$ is generated. Finally, the data owner uploads the encrypted version of the data file $F^{'} = (M_1, M_2, \ldots M_n)$ to the cloud.

$$IBE_{ID_O}(m_i) = (g_r, m_i.e(g^s, H_1((ID_O)^r)) \tag{1.5}$$

Re-encryption Key Generation Phase—$RKGen(P_{sys}, SK_{ID_O}, ID_O, ID_R)$ In this phase, the RKGen algorithm generates re-encryption key $RK_{ID_O \rightarrow ID_R}$ which is transferred to the cloud. Cloud uses it to permit the authorized user to decrypt the ciphertext using his own secret key. ID_R is identity of the requester.

$$RK_{ID_O \rightarrow ID_R} = (H_1(ID_O)^{-s}, IBE_{ID_R}(X)) where X \in G_T \tag{1.6}$$

Re-encryption Phase—$Reencrypt(P_{sys}, RK_{ID_O \rightarrow ID_R}, C_{ID_O})$. Here, the ciphertext encrypted using owner public key is transformed into ciphertext encrypted with requester public key. Using this algorithm, the re-encryption key generated in the previous stage $(RK_{ID_O \rightarrow ID_R})$ and the ciphertext for ID_O (C_{ID_O}) are used to generate ciphertext for ID_R (C_{ID_R}) as follows:

$$\begin{aligned} & C_{ID_R} = (c_1, c_2, c_3); \\ & where\ c_1 = g^r; \\ & c_2 = m * e(g^r, H_2(X)); \\ & c_3 = IBE_{ID_R}(X) \end{aligned} \tag{1.7}$$

Decryption Phase—$Decrypt(P_{sys}, SK_{ID_R}, C_{ID_R})$ This is the final phase of secure data service scheme, where the cloud server verifies the requester's re-encryption key and sends the re-encrypted file. The decrypt algorithm decrypts the ciphertext C_{ID_R} using SK_{ID_R} and retrieves the original message m_i.

$$m_i = \frac{c_2}{e(c_1, H_2(x))} \tag{1.8}$$

As the transformation of ciphertext has taken place in re-encryption stage, the requester can decrypt the file without the involvement of the data owner. In this way, the requester gets the entire file $F = (m_1, m_2, \ldots .m_n)$.

Efficient and Secure Data Storage Operations for MCC In [34], Zhou and Huang proposed a privacy preserving cipher policy attribute-based encryption $(PP\text{-}CP\text{-}ABE)$ scheme based on bilinear mapping, access tree, and secret sharing scheme. The architecture for the proposed scheme is illustrated in Fig. 1.9. This scheme mainly consists of five entities, namely data owner (DO), data requester/receiver (DR), encryption service provider (ESP), decryption service provider (DSP), and cloud service provider (CSP). The DO and DR store and retrieve data from cloud, respectively. But, computation intensive tasks like

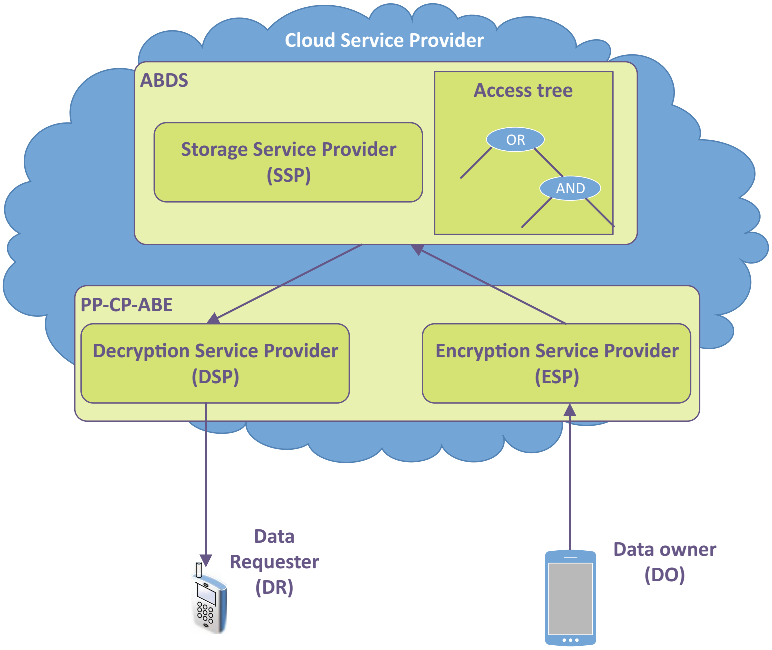
Cloud Service Provider
ABDS
Storage Service Provider (SSP)
Access tree
OR
AND
PP-CP-ABE
Decryption Service Provider (DSP)
Encryption Service Provider (ESP)
Data Requester (DR)
Data owner (DO)

Fig. 1.9 System architecture for the secure data storage framework

encryption and decryption operations are outsourced to CSP. When the data owner wants to upload a file to cloud, ESP encrypts the file without having knowledge about the security keys, and when the data requester/receiver wants to retrieve the file, it is decrypted by the DSP without any data contents being revealed to it. The CSP is used to store encrypted data. This scheme also consists of a trusted authority (TA) which is responsible for generating and distributing keys among data owners.

Access policy tree is constructed with the help of internal nodes and leaf nodes. The leaf nodes represent the attributes associated with DO, while internal node represents the logic gates (e.g., AND or OR). Bilinear mapping function for the scheme is defined as

$$e : G_0 \times G_0 \rightarrow G_1 \tag{1.9}$$

where G_0 and G_1 are the two multiplicative cyclic groups with large prime order p. Pairing also has the bilinearity property:

$$e(P^a, Q^b) = e(P, Q)^{ab} \forall P, Q \in G_0, \forall a, b \in Z_p^* \tag{1.10}$$

This scheme consists of the following phases:

Setup Phase In this phase, the trusted authority (TA) chooses a bilinear map

$$e : G_0 \times G_0 \rightarrow G_1 \tag{1.11}$$

of prime order p with generator g. TA then randomly selects $\alpha, \beta \in Z_p$ and constructs the public parameters PK which is known to everyone and the master key MK known only to itself. The generation of PK and MK is shown below:

$$PK = (G_1, g, h = g^{\beta}, f = g^{1/\beta}, e(g, g)^{\alpha}) \tag{1.12}$$

$$MK = (\beta, g^{\alpha}) \tag{1.13}$$

Registration Phase Users need to register with trusted authority TA to get private keys. Trusted authority authenticates the users based on their attributes (S) and generates a private key to each of the users.

$$SK = \{D = g^{(\alpha+\gamma)/\beta}, \forall j \in S : D_j = g^r . H(j)^{rj}, D_j^{'} = g^{rj}\} \tag{1.14}$$

where $r \in Z_p$, S represents user attributes, $r_j \in Z_p, and\ j \in S.$

Encryption Phase Before the data owner can start outsourcing computation of encryption, the DO needs to specify the data access tree (DAT). The DAT is divided into two sub-trees DAT_{DO} and DAT_{ESP}, where DAT_{DO} is DO controlled data access policy and DAT_{ESP} is cloud controlled data access policy.

$$DAT = DAT_{DO} \wedge DAT_{ESP} \tag{1.15}$$

Here, $\wedge$ represents a logical AND gate and depends on the root node of DAT. The DAT_{DO} normally contains one attribute. The DO randomly creates a one degree polynomial $(q_r(x))$ and generates secrets $s = q_r(0), s_1 = q_r(1), and\ s_2 = q_r(2)$. The DO sends DAT_{ESP} and s_1 to ESP. The ESP runs $Encrypt(s_1, T_{ESP})$ algorithm to generate temporal cipher (CT) on the basis of received information as depicted in the following equations:

$$CT_{ESP} = \{\forall y \in Y_{ESP} : C_y = g^{q_y(0)}, C_y^{'} = H(att(y))^{q_y(0)}\} \tag{1.16}$$

where Y_{ESP} is the set of leaf nodes in T_{ESP}

$$q_y(0) = q_{parent(y)} * (index(y)) \tag{1.17}$$

where $q_{root}(0) = s_1$ in case of DAT_{ESP} and Y_{ESP} represents the set of leaf nodes in DAT_{ESP}, att(y) returns the attributes associated with the leaf node y, and index(y) returns the unique index associated with each node.

In the meantime, DO completes the encryption process using s and s_2.

$$CT_{DO} = \{\forall y \in Y_{DO} : C_y = g^{q_y(0)}, C_y^{'} = H(att(y))^{q_y(0)}\} \tag{1.18}$$

$$C = Me(g, g)^{\alpha s}, C = h^S \tag{1.19}$$

where M is a message and e represents the bilinear mapping. DO sends CT_{DO} Ĉ and C to ESP. The ESP generates the ciphertext on the basis of received information.

$$\begin{aligned} CT = \{DAT = DAT_{ESP} \wedge DAT_{DO}; \hat{C} = Me(g, g)^{\alpha s}; \\ C = h^{s}; \forall y \in Y_{DO} \bigcup DAT_{ESP} C_y = g^{q_y(0)}, \\ C_y^{'} = H(att(y))^{q_y(0)}\} \end{aligned} \tag{1.20}$$

Decryption Phase In this phase, the computation required for decrypting is offloaded to decryption service provider (DSP) by data receiver/requester. Without revealing private key information, DO blinds the private key with the help of random number $t \in Z_p$.

$$D^t = g^{t(\alpha+r))/\beta} \tag{1.21}$$

$$SK_B = \{D_B = g^{t(\alpha+r)/\beta}, \forall j \in S : D_j = g^r.H(j)^{rj}, D_j^{'} = g^{r_j}\} \tag{1.22}$$

DSP uses the blinded key on encrypted file to generate a raw file. DO converts the raw file into the original file with acceptable processing and storage overhead. Detailed decryption process can be found in the paper [34].

Secure Cloud Storage for Data Archive of Smart Phones Secure cloud storage scheme for convenient data archive of smart phones proposed by Hsueh et al. [10] ensures the security and integrity of mobile users' files stored on cloud servers. The architecture of the proposed framework is illustrated in Fig. 1.10. This scheme mainly consists of four entities, namely mobile device which utilizes cloud services, certification authority which is responsible for mobile devices authentication, telecommunication module for generating and tracking mobile device password, and cloud service provider (CSP). In this paper, the authors assume that the secret key SK, public key (PK), and session key (SEK) are securely distributed among all mobile devices, the telecommunication module, and certification authority. The steps involved in the proposed scheme are explained below:

Registration Stage*—Step 1* The mobile user has to register with the telecommunication module via the certification authority to use the services offered by the cloud. The registration request from a mobile device to the certificate authority is represented as:

$$\begin{aligned} MD \rightarrow CA : \ & E_{PK_{TE}}(MU, NO, TK), U_n, S_{SK_{MU}}(H(MU, NO)), \\ & H(MU, NO), Apply \end{aligned} \tag{1.23}$$

where MU represents the mobile user, NO represents the user's phone number, TK is the combination of the phone number (NO) and cloud service password (CPW), U_n is the randomly generated number, H is a standard hash function, $E_{PK_{TE}}$

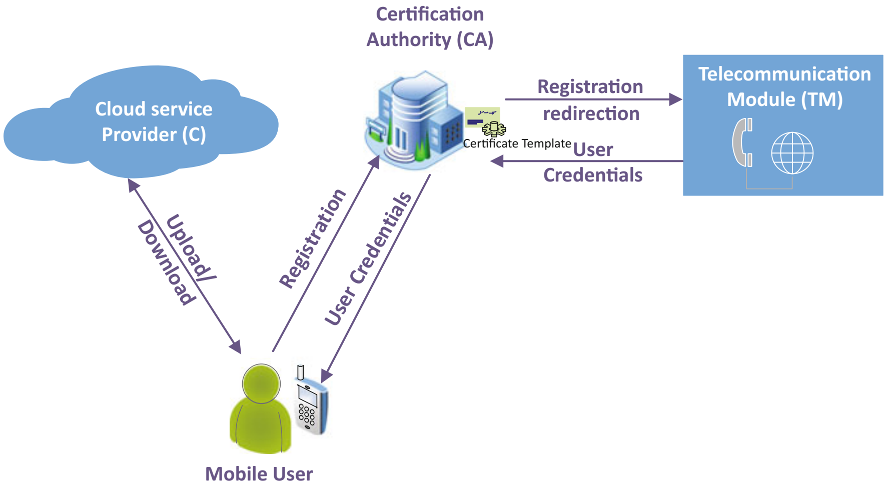

Fig. 1.10 Architecture for the secure data archive in cloud

represents encryption with the public key (PK) of telecommunication module (TE), and $S_{SK_{MU}}$ is a signature for the mobile user using a cryptographic function on the passed value and secret key (SK) of the mobile device. Signature is used to determine the legitimacy of the action and random number is used as proof of identity of the mobile user.

Registration Stage*—Step 2* After receiving the message from the mobile device, the certification authority validates the authenticity of the message with the help of received signature. If the message is from a valid user, certification authority sends the following message to telecommunication module:

$$CA \rightarrow TE : E_{PK_{TE}}(MU, NO, TK), U_n, S_{SK_{CA}}(H(MU, NO)), Apply \quad (1.24)$$

where $S_{SK_{CA}}$ is the signature of certification authority using cryptographic function on the passed value and secret key (SK).

Registration Stage*—Step 3* Upon receiving the above mentioned communication from certification authority, the telecommunication module generates the cloud service password CPW and transmits it to certification authority to be passed on to mobile user.

$$TE \rightarrow CA : E_{PK_{MU}}(MU, NO, U_n, E_{TK}(CPW)) \quad (1.25)$$

Registration Stage*—Step 4* Certification authority switches to pass on the cloud service password from telecommunication module to mobile device.

$$CA \rightarrow MU : E_{PK_{MU}}(MU, NO, U_n, E_{TK}(CPW)) \quad (1.26)$$

Registration Stage*—Step 5* The received cloud service password (CPW) is extracted and stored in phone memory to access cloud services.

Upload Stage Uploading data to cloud is a single step process in which mobile user uploads the data to cloud by encrypting it with session key (SEK).

$$MU \rightarrow Cloud : CPW, MU, E_{SEK}(Data), S_{MU}(H(MU||SV||E_{SEK}(Data)) \tag{1.27}$$

where SEK is the session key, SV is the secret value, and S_{MU} is the signature of the mobile user.

Download Stage*—Step 1* In order to download the file, mobile user sends the required information such as cloud service password, hash function, etc. The cloud can verify that the source of mobile user is correct using the hash function in the request.

$$MU \rightarrow Cloud : CPW, MU, H(MU||SV) \tag{1.28}$$

Download Stage*—Step 2* After verification, the cloud returns the user's personal data to the mobile device.

$$Cloud \rightarrow MU : E_{SEK}(Data), H(E_{SEK}(Data||SV) \tag{1.29}$$

The authors also explained the steps involved in synchronization and sharing of the data using the proposed scheme in [10].

Energy-Efficient Incremental Integrity for Securing Storage in MCC In the paper [12], Itani et al. addressed the issue of verifying the integrity of the data files stored in cloud servers. Their idea is to design secure data structures using the concepts of incremental cryptography and trusted computing which protect user documents with less energy consumption from mobile clients and also support dynamic data operations. As incremental cryptography, they proposed to use set of HMAC functions that supports the incremental update property. This means that if a message having a MAC value is updated by inserting/deleting a block of data, the incremental function can securely generate an updated MAC value using only the inserted/deleted block and old MAC value. According to this scheme, there are mainly three components: (1) mobile client which makes use of MCC services, (2) cloud service provider like Amazon, Google, etc. that provides cloud services, and (3) a trusted third party which sets up a set of secure cryptographic coprocessors in the cloud. Every coprocessor may be associated with more than one mobile clients and distributes a secret key (K_S) to each of the mobile clients. The entire system operation can be described as a three step process. Each of the three steps are described below:

(1) Initialization Step In this step, if a file F_i is being moved to cloud, an incremental MAC, MAC_{F_i} is calculated for it using the shared secret key K_S. This MAC value MAC_{F_i} is stored in mobile client itself and the files are moved to cloud servers.

(2) Data Update Step File update can be done mainly using three dynamic operations: (a) file creation, (b) file block insertion, and (c) file block deletion. Other file update operations such as block replacement and block movement can be performed using earlier mentioned insertion and deletion operations. In file creation operation, to protect the integrity of the newly created file F_{K+1}, MAC value $MAC_{F_{K+1}}$ is computed using the shared secret key K_S. The computed MAC value is stored locally in mobile client, and the file is transferred to the cloud. In file block insertion operation when the mobile client requests for the file F_i to which the block update is to be performed, the cloud sends a copy of the file to mobile client and another one to crypto coprocessor. Upon receiving the file F_i, crypto coprocessor computes the incremental MAC, MAC'_{F_i} for the file and transmits to the mobile client. Upon receiving the file to be updated F_i and MAC'_{F_i} from crypto coprocessor, mobile client verifies the integrity by comparing the received MAC value with the stored MAC value MAC_{F_i}. If the MAC values match, then block is inserted at the required location in the file and MAC_{F_i} is updated by applying incremental MAC operation on the inserted block only using old MAC value and the shared secret key K_S. The file block deletion operation is similar to insertion operation with only difference being MAC updated dependent on deleted block and the old MAC value.

(3) The Data Verification Step The mobile client can verify its files stored in the cloud at any time. The key advantage of the proposed scheme is that this integrity verification can be performed by mobile clients without incurring the overhead of files download or integrity verification. For integrity verification of collection of files or whole file system in the cloud, mobile client first sends a request to crypto coprocessor. Crypto coprocessor then successively retrieves the files from the cloud, generates their incremental MACs using shared secret key K_S, and transmits them to the mobile client. The mobile client then compares the received MACs and stored MACs to verify the integrity of the files.

4.4 Security Frameworks for Computation Using MCC

Securing Authentication and Trusted Migration of Weblets in the Cloud with Reduced Traffic A mechanism for securing communication among the Weblets in elastics applications using mobile cloud computing is proposed in [21]. In the proposed security framework, the authors tried to accomplish three security objectives with respect to elastic applications in MCC. For an elastic application, Weblets can be running either in mobile device or cloud or in both. The location where a Weblet is launched is decided by device elasticity manager (DEM) and cloud elasticity services (CES). Weblets might be migrated from cloud to device or vice versa based on computation they perform and the application. In some applications, Weblets launched in cloud and mobile devices may work independently and in other applications, they might be working in concurrence to accomplish the task based on requirement. First is to provide secure migration of Weblets between cloud and device. Second one is to enable better authentication of the Weblets and the third objective is to manage the traffic in the communication channel between cloud and device. The ideas to accomplish the three objectives are discussed below:

Secure Migration of Weblets Using SSH Protocol To ensure secure migration of Weblets, communication channel between mobile device and cloud is set up using tunneling mechanism with secure shell protocol (SSH) that employs public/private key authentication to verify the end nodes. Initially, the firewall of the mobile device is tunneled by http tunneling at port 80 which is a universally opened port. Next, communication channel is established between mobile device and cloud using SSH protocol on port 22 to form the tunnel. Initially, a request is sent from the mobile device to the cloud to establish the tunnel via Internet. Then the transport layer protocol authenticates. Transport protocol component of SSH acknowledges mobile device with the cloud and user authentication protocol part of SSH acknowledges cloud with device information. After successful verification at both ends, tunnel is established and connection protocol part of SSH multiplexes it into a logical link. Figure 1.11 illustrates the proposed secure tunnel establishment between mobile device and cloud network.

Better Authentication Using SFTP Protocol Weblets are subjected to secure file transfer protocol (SFTP) to have additional layer of security around it. As part of this, Weblets are encrypted after entering the tunnel for transmission. Since, SFTP also works with port 22, another tunneling in the firewall would not be required. Every Weblet transmission is associated with a user authentication key and host authentication key. User authentication key in the Weblet is to prove their genuine mobile location and host authentication key is transmitted to the cloud before Weblet transmission is started. It ensures that each Weblets reaches their correct cloud virtual network and it is changed frequently by SSH. SFTP together with SSH ensures secure migration and authentication of the Weblets.

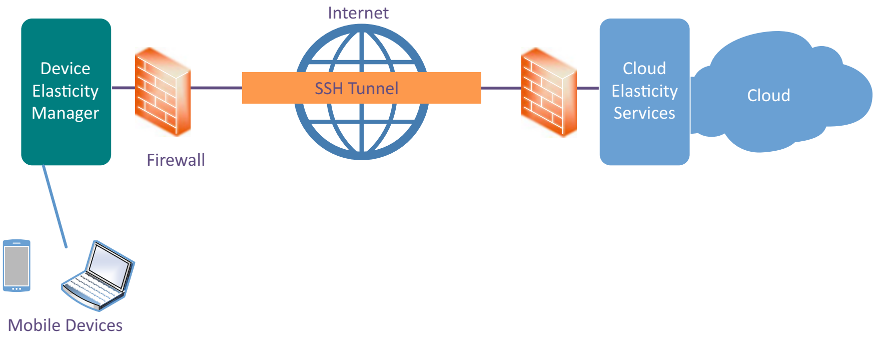

Fig. 1.11 Secure communication channel between mobile device and cloud

Reducing Traffic in Channel with Back Pressure Technique The authors propose to use the backpressure technique of fluids in the communication channel to manage traffic in the communication channel. If the traffic increases, the back pressure technique backs up the Weblets towards the Weblets' origin which could be either mobile device or cloud based on the direction of migration. Once the traffic reduces, the channel releases the back pressure to transmit the Weblets.

Secure Cloud Framework for Mobile Computing and Communication Using MobiCloud Huang et al. [11] proposed a new MCC framework which is a new service oriented model of mobile ad hoc networks (MANET). In this model, each mobile node is termed as service node which may provide or consume a service. Service includes sensing services, storage services, or computation services. This framework is close to the architecture described earlier in the architecture section where there are virtual images of mobile phones stored in the cloud to offload some of the tasks performed by physical devices. In this model, these are termed as extended semi shadow images (ESSI). These could be a partial clone, an exact clone, or an image of device having extended functionality. The communication channels between mobile node and ESSI is through a secure connection like SSL, IPsec, etc. The architecture of MobiCloud as shown in [11] is illustrated in Fig. 1.12.

In this architecture, information flow and data access control are isolated by creating multiple virtual domains using network virtualization service called

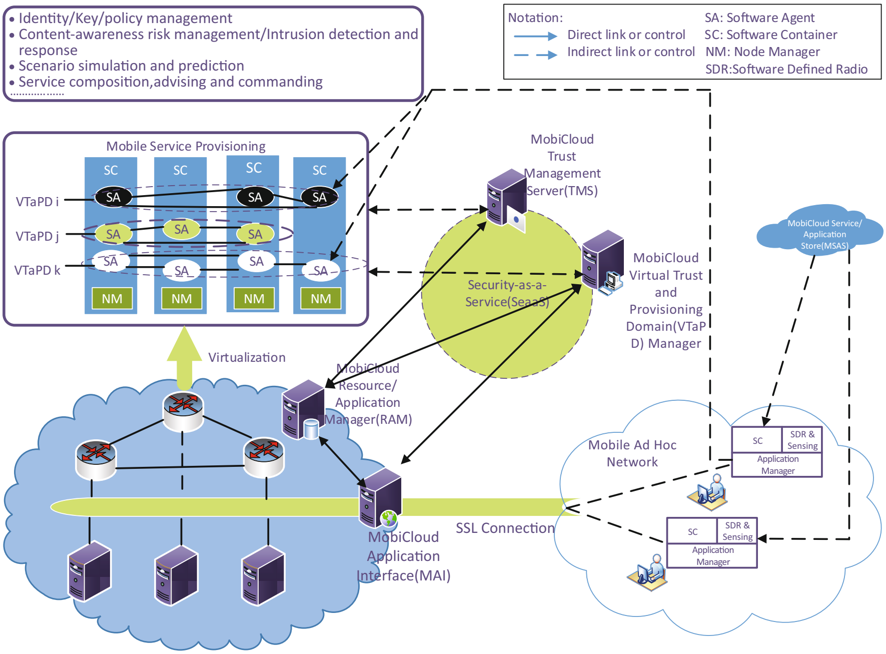
• Identity/Key/policy management
• Content-awareness risk management/Intrusion detection and response
• Scenario simulation and prediction
• Service composition,advising and commanding
Notation:
Direct link or control
Indirect link or control
SA: Software Agent
SC: Software Container
NM: Node Manager
SDR:Software Defined Radio
Mobile Service Provisioning
SC
SC
SC
SC
VTaPD i
VTaPD j
VTaPD k
SA
NM
MobiCloud Trust Management Server(TMS)
MobiCloud Virtual Trust and Provisioning Domain(VTaPD) Manager
Security-as-a-Service(SeaaS)
MobiCloud Service/ Application Store(MSAS)
Virtualization
MobiCloud Resource/ Application Manager(RAM)
MobiCloud Application Interface(MAI)
SSL Connection
Mobile Ad Hoc Network
SC
SDR & Sensing
Application Manager

Fig. 1.12 MobiCloud architecture

VTaPD. This is created using programmable routers [17]. MobiCloud Application Interface (MAI) plays the key role on the server side. It is responsible for the services used by mobile devices and also provides interfaces for: (a) MobiCloud Virtual and Provisioning Domain manager ($VTaPD$) module, (b) resource and application manager module. All the nodes in a particular $VTaPD$ have the complete routing information of that particular $VTaPD$. On the device side, link between mobile device and cloud services is achieved through software agent. A single software agent may be running on a mobile device as well as on a cloud platform also. Under the supervision of application manager, a mobile device can access multiple cloud services or MANETs with the help of multiple software agents. Sensor manager is responsible for collection of sensing information such as battery status, location information, etc. from the mobile nodes and node manager handles the loading and unloading of software agents on ESSI. Each $VTaPD$ is associated with multiple software agents that may belong to different $ESSI$. The $VTaPD$ manager decides on the intrusion detection and risk management by collecting sensing information from mobile device. The trust management server module handles the key management, data access management, and user-centric identity management.

In this scheme, the authors used the attribute based key management which uses multiple attributes to identify an entity. When compared with the asymmetric encryption technique where attributes are considered as public keys and trusted authority generates the corresponding private keys. Only difference being the private keys are not generated from large prime number but instead from descriptive terms. The private keys are securely distributed to mobile devices by the trust management server module. The attribute based identity management scheme defines point of network presence ($PoNP$). The line radiating from the $PoNP$ shows the relationship of mobile users to various counter parties. Each $PoNP$ is a combination of type, value, and attributes. The type consists of: (a) identity issuer, (b) private key issuer, and (c) validation period. The PoNP may have multiple attributes. Each attribute is the combination of type and value. The default $PoNP$ is associated with each individual having a unique value. The uniqueness is achieved by applying a publicly known hash function on some uniquely identifiable attribute of the mobile user (e.g., passport number, email address, or driver license identity). Multiple $PoNPs$ or attributes are used to define the publicly known native identities for the device. These identities are used for authentication, authorization, and access control.

Securing Elastic Applications on Mobile Devices for Cloud Computing As mentioned in the first section, computation tasks of mobile devices can also be offloaded to cloud servers to achieve fast processing or save battery consumption, etc. In this paper, the authors tried to address various security issues associated with elastic mobile application. Zhang et al. [30] proposed a design for elastic devices which are resource constrained devices such as mobile phones augmented with cloud-based functionalities. They proposed framework for elastic applications that can run efficiently on resource constrained devices by transparently making

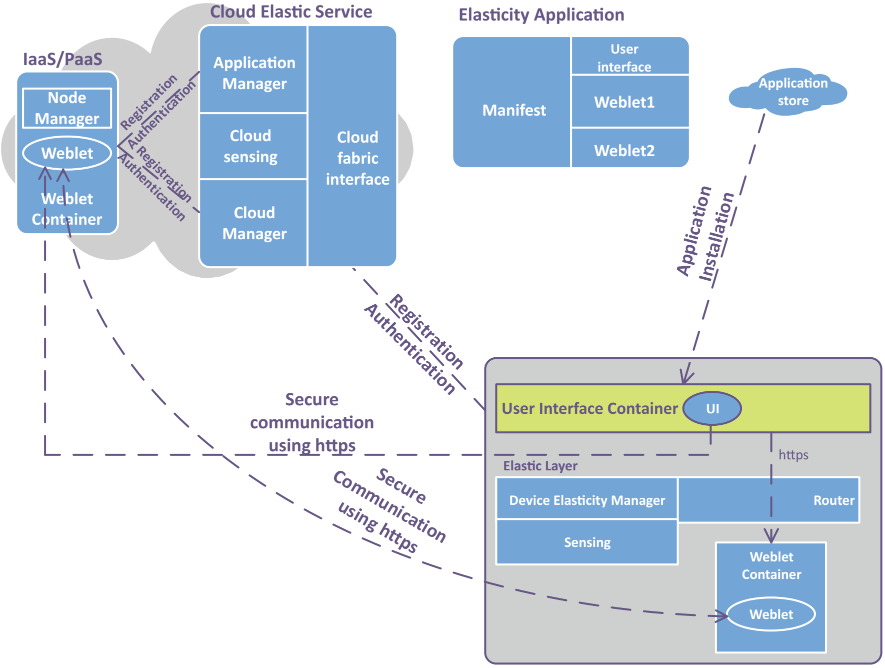

Fig. 1.13 Security framework for elastic computation

use of cloud resources whenever needed. Basically, an elastic application consists of one or more Weblets, each of which can be launched on a device or cloud or migrated to cloud based on requirement and also communicate with each other. They also discussed the security requirements in such a model and proposed security framework for elastic mobile applications in cloud environment.

Figure 1.13 illustrates the architecture of the proposed security framework in elastic applications as shown in [30].

On the Device Side Key components on the device side include (1) device elasticity manager, (2) router, (3) sensing collector. Device elasticity manager (DEM) takes care of the application launch configuration and run time adjustments. Configuration includes Weblet location, selection of communication paths, etc. It also maintains a cost model and optimizer based on which configuration settings of the application are decided. Router module keeps track of the location of Weblets and makes the location of Weblets transparent to users. Sensing module maintains the information regarding device utilization and shares the same with DEM when required.

On Cloud Side In the cloud, we have the cloud elastic service (CES) that consists of: (a) cloud manager that maintains each Weblets resource utilization details like memory used, bandwidth consumed, and computation. (b) Application manager that is responsible for installation and maintenance of elastic application in the cloud.

(c) Sensing module that monitors failures and resource availability. Resource usage within cloud node is monitored by the node manager.

In this paper, the authors tried to address various security issues associated with elastic mobile applications. For each application, the developer calculates the $SHA1$ value of each Weblet and stores it in Java like application package. Only authenticated users can install the applications. While installing the application in device or cloud, the installer re-computes the hash value and compares it with the stored hash value for integrity verification. A Weblet session key and Weblet session secret are generated by DEM. The scheme also described the authentication process among Weblets. During application launch, the keys are distributed to all Weblets that belong to an application. Initially, the Weblet generates a hash-based message authentication code ($HMAC$) using nonce, source Weblet ID, destination Weblet ID, and Weblet session secret. This along with the original message is sent to the destination Weblet. Upon receiving the message, destination Weblet recalculates the $HMAC$ using the received message and own session secret key. This value is compared with the received $HMAC$ in order to authenticate the Weblet. This paper also describes the secure process to migrate a Weblet from device to the cloud. When Weblet migration is required, DEM sends a request to Weblet to stop executing. The Weblet saves the running state and session secret and returns to DEM which in turn sends a request to cloud manager using the cloud fabric interface for migration. Cloud manager allocates resources to migrated Weblet and starts the execution from last saved state.

SMOC: A Secure Mobile Cloud Computing Platform Mobile offloading is a new concept that is been used ubiquitously. Hao et al. [8] suggested to run an operating system of mobile device and arbitrary applications on a cloud-based virtual machine. It leverages the hardware virtualization functionality on the smart mobile device. The authors provided two design fundamentals in detail. First sharing a resource platform so that an application running can freely migrate between the user's mobile device and a backend cloud server. A special file system extension was designed to enable free migration. Second, hardware virtualization technology, which isolates the data from the local mobile device operating system is used to protect user data. The authors introduced two client programs of which one is depicted as input proxy responsible for capturing the user's input before passing them to the guest OS and sending it to the VM so that guest OS cannot learn anything about the user's input. On the other hand, the guest OS is the location where app's output is rendered. Even the compromised VM cannot affect other apps and data. The user can also delete the compromised VM afterwards. The user response time can be improved, and the device energy consumption can be reduced if heavy computational processes can be moved from mobile device to the cloud VM. It also has security advantages. The platform design is discussed in detail in this paper. In conclusion, the authors say that they successfully implemented a prototype of the suggested platform using off-the-shelf hardware. The platform is then observed to be efficient, practical, and secure.

5 Attack, Risk Assessment, and Verifiability in Mobile Clouds

Cyberattack Detection in MCC Using a Deep Learning Approach Mobile cloud computing (MCC) is an emerging architecture which provides a myriads of benefits because of which it has become a soft target for cyber threats in the mobile cloud environment. In this paper [19], the authors proposed a framework with an advanced detection mechanism developed from deep learning technique. It allows to detect various attacks with high accuracy. They also have discussed the limitations of intrusion detection approaches used by other researchers. In the system model proposed by the authors, when the request is sent from a mobile user to the system, it goes through attack detection module which has various functions to detect the attack. Every request is verified carefully by comparing with the current database and/or sending to security service providers for double-checking. If the request identified as harmless, it is treated normally, whereas in other case, the request is treated as malicious and attack defend function works on the request to implement prompt security policies in order to prevent the spread as well as impacts of this attack.

The authors explained deep learning model for cyberattack detection and explained how the learning model detects cyberattacks in the cloud system. The learning model detects cyberattacks in the cloud system. It consists of two phases, namely feature analysis and learning process. Different types of malicious packets may have special features usually discrete from normal requests. These special features will be used to differentiate the malicious requests from the normal requests. The deep training is implemented using the set of simple sub-models which are learned sequentially. The non-linear transformation is used to obtain sensible set of weights. These training weights decide if the request is malicious or harmless. The authors discussed dataset collection and evaluation methods to evaluate the experimental results. It was observed that the proposed method can achieve high accuracy in detecting cyberattacks, and outperform other existing machine learning methods. In addition, they have emphasized the stability, efficiency, flexibility, and robustness of the deep learning model which can be applied to many mobile cloud applications.

A Stochastic Programming Approach for Risk Management in MCC Mobile cloud computing (MCC) is an emerging platform because using cloud computing technology, mobile applications can be performed more efficiently, thereby generating huge profits for mobile users as well as cloud service providers. Although there are many security solutions implemented to detect and prevent cyberattacks, achieving complete security is still nearly impossible. The cyber insurance is an emerging alternative to address and manage cyber risks. Under cyber insurance coverage, cloud service provider (CSP) losses will be covered partially or fully by the cyber insurance provider (IP). However, cyber insurance is not always the best solution of the variety of attacks and CSP's limited budget. The authors of this paper [9] discuss the advanced risk management strategies to minimize losses caused by cyberattacks, to select appropriate security solutions, software/hardware implementation and insurance policies, to deal with different types of attacks. They developed a dynamic framework based on stochastic programming approach for the risk management problem in mobile cloud environment. The sole purpose of the framework is to find the optimal trade-off among security policies, insurance policies, and countermeasures under uncertainty of cyberattacks and their losses such that the expected total cost of the cloud service provider is minimized. The framework suggested consists of two decision stages for CSP. In the first stage, the CSP has to decide how much it should invest to buy security packages to prevent cyberattacks and how much it should spend to buy insurance packages to cover losses caused by the attacks. To address the multi-stage optimization problem under uncertainty of attacks and a limited budget, the authors adopt the stochastic programming method to find optimal budget allocation policies for the CSP.

Secure and Verifiable Outsourcing of Exponentiation Operations for MCC Cloud computing allows the end users to securely access the shared pool of resources such as computational power and storage. Among all the computation types, modular exponentiation and scalar multiplication on elliptic curves in finite group is a widely used cryptosystem. It includes large integers, making it expensive for mobile phones. Outsourcing the exponentiation operation to cloud server is a cheap option but not secure. Kai et al. have introduced a secure outsourcing scheme (ExpSOS) [33] which requires limited number of modular multiplications at local mobile environment. The procedure depends on secure disguising procedure that maps the integers in the group which maps into another larger group so that cloud can carry out the computation in larger group keeping data secure. The end-user can recover the result back from the result returned by the cloud. The authors assumed that for the end-user, exponentiation operations are operated in the integer ring modulo N, where N is not necessarily a prime number. They multiplied N by a randomly selected large prime p and define L = pN as a part of secure disguising procedure. Then k was selected such that $1 \leq k \leq p-1$. $y = x + kN \pmod{L}$ was computed, where x is input to cloud. It is hard to determine which point x is mapped to, without the knowledge of k. The algorithm for the ExpSOS protocol under honest but-curious single-server (HCS) model is discussed in detail in this paper. They considered secure outsourcing as two building blocks to implement scalar multiplication, point addition, and point doubling. The ExpSOS is then analyzed by the authors for the necessary properties of a result verification scheme through some counterexamples and security complexity analysis is done over it. It was then concluded that ExpSOS enables end users to outsource the computation of exponentiation to a single untrusted server at the cost of only a few multiplications and it can provide different security levels at the cost of different computational overhead. ExpSOS also provides a secure verification scheme with probability approximately 1 to ensure that the mobile end users do always receive a valid result.

A Practical, Secure, and Verifiable Cloud Computing for Mobile Systems It is known that providing data to the cloud service provider in plaintext may lead to loss of data privacy. Premnath et al. [24] combined the secure multiparty computation protocol and the garbled circuit design with the cryptographically secure pseudorandom number generation method, which enables cloud to perform any arbitrary computation on encrypted data. In this method, with private pseudorandom bit sequences and Boolean circuit, the servers create garbled circuit. The servers create the garbled circuit as $GC = GC1\bigoplus GC2\bigoplus GC3\bigoplus.....\bigoplus GCn$ by performing an XOR operation on the shares obtained by local computations over private pseudorandom bit sequences and the Boolean circuit of the servers. With the use of these garbled inputs, another server executes garbled circuit to give garbled outputs.

In this process, the client sets servers and sends desired computation and seed value to each server. Each server creates a Boolean circuit which corresponds to the requested computation. Each server generates a private pseudorandom bit sequence using seed value. Using the semantics for the pair of garbled values, the client translates these garbled values into plaintext bits to recover the result of the requested computation. The client checks garbled output for each output wire matches with garbled values that it computed on its own. The system assures privacy of the mobile clients' data. It can enable oblivious evaluation of any arbitrary function on a third-party cloud server. It was also observed that it requires very little computation and communication participation from the mobile client to achieve secure and verifiable computing capability with this method. This method is also useful to detect a cheating evaluator if it provides output without performing any computation.

Deep Learning for Secure Mobile Edge Computing Mobile edge computing (MEC) is the most efficient approach for enabling cloud-computing capabilities over cellular networks. However, security is increasingly becoming a challenging issue in MEC-based applications. In this paper [4], the authors proposed a deep-learning-based model in order to detect security threats and malicious attacks incorporating the location information into the detection framework. The model uses unsupervised learning to automate the detection process at the edge of a cellular network. It includes feature preprocessing engine and malicious application detection engine. In the feature preprocessing engine, APK files were unpacked and the feature elements that will be used as the input of the malicious application detection engine were extracted. The two-dimensional array of bits was created based on which kind the feature element falls into. The malicious application detection engine includes first unsupervised pre-training with unlabeled samples and second, supervised fine-tuning with labeled samples.

The deep learning architecture has a multi-layer stack of modules to compute the non-linear input–output mapping. The automated learning of the features using a general-purpose learning algorithm is the key advantage of deep learning. In case of non-linear and complicated relations between features and malicious applications, the output layer of the malicious application detection engine was used as the input of the SoftMax function to represent a categorical distribution. They compared the performance of the proposed model with four widely adopted machine learning algorithms, namely Support Vector Machine, Decision Tree, Random Forest, and

SoftMax Regression. The strengths of the proposed model are discussed in detail. It was also observed that the size of the training dataset plays an important role in improving the accuracy of the deep-learning-based detection method. The authors concluded that on average the accuracy of the proposed deep-learning-based model is more compared to other four detection methods.

6 Summary and Discussion

Table 1.1 gives a cumulative picture of various security frameworks discussed in this chapter and the main ideas behind their security mechanism. In addition, we provide an executive summary and discuss some of the possible research ideas for future in reference to the work discussed in this chapter.

In [7] proposed by Chow et al., secure authentication is achieved by using implicit authentication mechanism. Observable user information is collected and stored on data aggregator after hashing at the mobile client to preserve the privacy of the users. The authentication engine makes use of this information for implicit authentication and generates result for authentication consumer. But this requires frequent application of hash function by the mobile client each time the user related information is transferred to the data aggregator. This may result in computation overhead on the mobile device. The proposed idea of using biometric information along with encryption or secret key for authentication by Zhao et al. in [31] could be feasible to implement in future as biometric sensors can be accommodated in the mobile devices. But as of date not many mobile devices are equipped with biometric sensors to implement the proposed framework. On top of it, there are certain pitfalls with biometric features as well. They are prone to problems like false acceptance, nearest impostors attack, change in fingerprint with age, etc. Secure storage of biometric information is also a challenge. Moreover, biometric science is still developing and challenges associated with using biometric encryption still needs an in-depth research. Another authentication framework [32] proposed by Zhou et al. is similar to the scheme proposed in [7]. But this one could be more secure than [7] since this considers more parameters like periodic events, spatial information, and others also while building the context aware data for a particular user. But in this scheme also the process of updating the context data of the user in the cloud could cause communication and computation overhead for the devices. The authors [25] discuss how biometric is the most effective method to authenticate the users and to protect from illegal and unauthorized customers. However, the authors have not implemented or simulated the log files based on their scheme. Also, they need to redesign policies for accessing log record as they will presumably be utilized to discover unapproved endeavors to get to data by outsiders, the cloud supplier, or any gatecrashers. In the other authentication framework [18] proposed by Lomotey el al., the authors propose to use a middleware layer having interface with social media network to handle authentication with Amazon S3 on behalf of mobile device. But this scheme again has the drawback of using id and password

Table 1.1 Cumulative study of the proposed frameworks

Paper title	Security issue addressed	Main idea
Authentication in the clouds: a framework and its application to mobile users [7]	Authentication between mobile client and cloud	Implicit authentication
Feasibility of deploying biometric encryption in mobile cloud computing [31]	Authentication in mobile cloud computing	Combination of biometric identification and secret key called biometric encryption
A framework for secure mobile cloud computing [25]	Biometric authentication	Preprocessing steps and algorithms for extracting the features and matching the biometrics trait
The context awareness architecture in mobile cloud computing [32]	Authentication of user in mobile cloud computing	Using context aware computing
Consolidated identity management system for secure mobile cloud computing [14]	Securing lost, stolen, or compromised personal identifiable information	Architecture dubbed consolidated identity management system (CIDM) which countermeasures the vulnerabilities to personal identifiable information
Improved identity management protocol for secure mobile cloud computing [22]	User ID management and security problems	Using improved IDM3G protocol along with an additional authentication management protocol
Middleware-layer for authenticating mobile consumers of Amazon S3 data [18]	Authentication of mobile consumers for Amazon S3	MiLaMob framework, social network media, and hybrid authentication mechanism
Securing authentication and trusted migration of Weblets in the cloud with reduced traffic [21]	Secure migration of Weblets in elastic application using MCC	Secure shell protocol, secure file transfer protocol, and back pressure technique
MobiCloud: building secure cloud framework for mobile computing and communication [11]	Proposed MobiCloud framework and addressed relevant security issues	Network virtualization service, attribute based key management
Securing elastic applications on mobile devices for cloud computing [30]	Secure installation, migration, authentication, and authorization of Weblets in elastic applications	Hash function for installation, shared session, and secret keys for authentication and migration. Application session keys and application session secrets for authorization
SMOC: a secure mobile cloud computing platform [8]	Security against untrusted applications	Sharing a resource platform and hardware virtualization technology
Mobility can help: protect user identity with dynamic credential [29]	Identity protection/privacy protection	Uses randomness in user-cloud communication to generate dynamic credentials

(continued)

Table 1.1 (continued)

Paper title	Security issue addressed	Main idea
Privacy protection for mobile cloud data: a network coding approach [6]	Huge computing power challenge and privacy issue of untrusted cloud server	Development of unconditionally secure network coding based pseudonym scheme
A security framework of group location-based mobile applications in cloud computing [5]	Identity or privacy protection	Hash function on IMSI number of mobile client
In-device spatial cloaking for mobile user privacy assisted by the cloud [28]	Privacy protection	Spatial cloaking
Efficient and secure data storage operations for mobile cloud computing [34]	Security of data stored in cloud	Privacy preserving CP-ABE based on bilinear mapping access policy tree and secret sharing scheme and attribute based data storage scheme
Secure cloud storage for convenient data archive of smart phones [10]	Security and integrity of data stored in cloud	Uses the standard cryptographic functions
SDSM: a secure data service mechanism in mobile cloud computing [13]	Secure data storage and data sharing in cloud	Bilinear mapping based identity based encryption and proxy re-encryption
Energy-efficient incremental integrity for securing storage in mobile cloud computing [12]	Enables verification of integrity of files stored in the cloud	Incremental cryptography and trusted computing
A deep learning approach for cyberattack detection in mobile cloud computing [19]	Data integrity, users confidentiality, service availability	Detect and isolate cyber threats using advanced detection mechanism based on deep learning approach
A stochastic programming approach for risk management in mobile cloud computing [9]	Software and hardware implementation and insurance policies	Stochastic programming approach to minimize the expected total loss for the cloud service provider
Secure and verifiable outsourcing of exponentiation operations for mobile cloud computing [33]	Secure outsourcing of exponentiation operations to one single untrusted server	Secure outsourcing disguising scheme (ExpSOS) which requires limited number of modular multiplications at local mobile environment
A practical, secure, and verifiable cloud computing for mobile systems [24]	Data privacy	Combination of secure multiparty computation protocol and the garbled circuit design with the cryptographically secure pseudorandom number generation
Deep learning for secure mobile edge computing [4]	Security of mobile edge computing	Deep-learning (unsupervised learning) based model to detect security threats and malicious attacks using location information

to initially identify with the middleware. So if the attacker can get the user id and password through phishing or other social engineering attacks, he can get access to the user's social media (if the user uses social media to identify himself to the middleware) and also the data stored in Amazon S3.

The authors in [14] introduced a new IDM architecture dubbed consolidated IDM (CIDM) which countermeasures possible vulnerabilities. But the authors have not investigated the possibilities, consequences, and countermeasures of cloud provider compromise through, for example, tampered binaries, injected malicious code, or malicious insiders. Also, the authors have failed to investigate the issue of inadequate dynamic federation and agile mechanisms in current IDM systems which is an architectural concern and should be addressed at the design level. The proposed method in [22] maintains the mobile operators (MO) and constructs a trusted base with cross certification between service providers and MO. While it depends on public key infrastructure (PKI) to enable mutual dependence-based communication and ID management by service providers, the authors have not evaluated the context of DoS attack which should be conducted continually and additional studies of the PGP algorithm are needed. In security schemes presented for computation using MCC, the security scheme proposed in [21] by Panneerselvam et al. is based on the idea of establishing a tunnel using secure shell protocol and secure file transfer protocols for secure migration of Weblets from mobile to cloud and vice versa. Though the scheme is straightforward and feasible, it requires an additional task of constant monitoring of the tunnel since attackers can use the tunnel to bypass the firewall on either side. MobiCloud framework proposed by Huang et al. in [11] is showed to enhance the MANET functionality. But in the proposed security mechanism the authors did not consider the trustworthiness of the cloud node. Mobile user information should also be securely stored in the cloud. An elastic mobile cloud application model was proposed by Zhang et al. in [30]. They also proposed security framework for the same which includes secure installation, secure migration of Weblets, authentication between the Weblets and authorization of Weblets. Though the proposed scheme ensures secure installation, it does not mention about the security threat to Weblets after installation of Weblet in the cloud. If an attacker can modify the code of the Weblet in the cloud, then it can result in configuration change of DEM and CES.

In the security schemes presented for privacy preservation, the security scheme proposed by Chen et al. [5] to preserve the privacy of LBS users is based on the idea of using hashed IMSI number. But if the IMSI number is stolen from the legitimate user, the entire system fails. Another privacy preserving scheme discussed in this paper was proposed by Xiao et al. [29]. It is based on the concept of dynamic credentials where the credentials are constantly changed based on the communication between user and the cloud. But in this scheme, the cloud which is also a third party is assumed to be trusted entity which is a very strong assumption. In another privacy preserving scheme, Wang et al. [28] proposed a privacy preserving framework for location-based services using mobile cloud computing. But the accuracy of the proposed mechanism is dependent on historical lower bound of the number of users in each grid cell. This is because it predicts number of users in each grid cell based on the historical data that may be wrong at that instant of time which in turn results in privacy loss. The enhancing secure pseudonym scheme to protect the privacy of mobile cloud data and unconditionally secure lightweight network coding pseudonym scheme [6] will face the huge computing power challenge as well as two-tier network coding challenge to solve privacy issue of untrusted cloud server.

Coming to the secure storage frameworks, Jia et al. [13] proposed a secure data storage scheme which is based on proxy re-encryption and identity based encryption. This scheme is designed to offload most of the security tasks to cloud, the mobile users have to perform cryptographic operations before uploading file to cloud which require considerable amount of energy. Moreover, utilizing cloud resources for all the cryptographic computation may increase the usage charges to the user. Zhou and Huang [34] also proposed a security framework based on privacy preserving CP-ABE and attribute based data storage scheme. The underlying CP-ABE scheme is proven to have linearly increasing ciphertext with increase in attributes. As the proposed scheme also involves a kind of CP-ABE, it also suffers from the same drawback. Another work proposed by Hsueh et al. [10] used standard asymmetric encryption techniques to encrypt the files and then stores them on the cloud servers. But due to this process, the computation overhead in the mobile devices increases. The security framework proposed by Itani et al. [12] provides a way for mobile user to verify the integrity of files stored in the cloud. This scheme is based on the incremental cryptography and trusted computing. The proposed security framework is clearly energy efficient mainly for two reasons. First, due to use of incremental MAC, computation overhead on the mobile client is greatly reduced as we need not compute the hash value for whole file every time it is updated. Second, while verifying the integrity of the file(s), the mobile client just need to compare MAC values as the task of computing MAC value for file(s) is done by crypto coprocessor. But the proposed scheme only provides a way to verify the integrity of the file stored in the cloud. It does not protect the files from being modified or unauthorized access, as files are directly moved to cloud and in cloud computing environment, cloud service provider is also a third party and can be a potential adversary.

The authors in [19] proposed a deep learning model for cyberattack detection but have not implemented on real devices and evaluated the accuracy of the model on the real time basis. Also, they have not evaluated the energy consumption and detection time of the deep learning model and compared with other methods. The authors of this paper [9] discuss the advanced risk management strategies to minimize losses caused by cyberattacks to select appropriate security solutions, software/hardware implementation, and insurance policies to deal with different types of attacks. However, they have not studied the relation between security and insurance providers through bundling strategies and matching theory. In addition, they agree that they have not investigated the relation between a direct loss and its indirect losses. The paper [33] suggested ExpSOS scheme with the security parameter, which is cost-aware in that it can provide different security levels at the cost of different computational overhead. Hence, it is difficult to provide the cost of entire process beforehand. The authors [24] suggested to use private pseudorandom bit sequences and Boolean circuit that the servers use to create garbled circuit. This method preserves the privacy of the client data even if the evaluating server colludes with all but one of the cloud servers that participated in the creation of the garbled circuit. In this paper [4], the authors proposed a deep-learning-based model in order to detect security threats and malicious attacks incorporating the location information into the detection framework. However, the critical challenge is handling streaming and fast-moving input data and to use these data to train the deep-learning-based model.

Mobile cloud computing is inherited from cloud computing and hence many of the security issues in cloud computing also exist in mobile cloud computing. MCC also has an added constraint of limited computational resources at the mobile device end that needs to be considered while designing the security frameworks. Hence, some of the security frameworks that work well with cloud computing may not be applicable to MCC. Lightweight frameworks are needed for mobile cloud computing. All the frameworks we discussed in this chapter perform the CPU intensive tasks in the cloud to avoid overhead in the mobile devices. Cryptographic functions like hashing, other high computation tasks are designed to be offloaded to the cloud. Cloud services are mostly charged based on usage so this concept of offload computation tasks actually becomes a trade-off between energy saved at the device side and expenses paid for the cloud usage.

7 Conclusion

Mobile cloud computing (MCC) provides mobile users with a rich resource functionality despite the restricted resources in their mobile devices. In this chapter, initially, we discussed the importance of different mobile cloud computing frameworks and their implicit advantages. Next, we described the key architectures of the mobile cloud computing, and key aspects of security in mobile cloud computing environment. Next, we reviewed some of the security frameworks proposed for mobile cloud computing. Privacy is a significant challenge in using mobile cloud-based services, particularly when processing mobile users' data or applications and when shifting them from mobile devices to heterogeneous distributed cloud servers located at multiple locations. Thus, next, privacy issues and some solutions in mobile cloud computing domain have been discussed. We also discussed secure storage for mobile cloud as well as secure computing ideas for mobile cloud computing. Later, we provided a discussion section where after providing a summary, we contrasted different schemes and compared them with possible future work. Thus, this chapter will serve as a good review of the security work in MCC for those who are targeting research and building applications in this area.

References

1. Aref, W. G., & Samet, H. (1990). Efficient processing of window queries in the pyramid data structure. In *Proceedings of the Ninth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, PODS '90* (pp. 265–272). New York: ACM. https://doi.org/10.1145/298514.298579
2. Buyya, R., Yeo, C. S., Venugopal, S., Broberg, J., & Brandic, I. (2009). Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation Computer Systems, 25*(6), 599–616. https://doi.org/10.1016/j.future.2008.12.001
3. Chen, E., & Itoh, M. (2010). Virtual smartphone over IP. In *IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks (WoWMoM), 2010* (pp. 1–6). https://doi.org/10.1109/WOWMOM.2010.5534992
4. Chen, Y., Zhang, Y., & Maharjan, S. (2017). Deep learning for secure mobile edge computing. arXiv preprint arXiv:1709.08025.
5. Chen, Y. J., & Wang, L. C. (2011). A security framework of group location-based mobile applications in cloud computing. In *40th International Conference on Parallel Processing Workshops (ICPPW), 2011* (pp. 184–190). https://doi.org/10.1109/ICPPW.2011.6
6. Chen, Y.-J., & Wang, L.-C. (2017). Privacy protection for mobile cloud data: A network coding approach. arXiv preprint arXiv:1701.07075.
7. Chow, R., Jakobsson, M., Masuoka, R., Molina, J., Niu, Y., Shi, E., et al. (2010). Authentication in the clouds: A framework and its application to mobile users. In *Proceedings of the 2010 ACM Workshop on Cloud Computing Security Workshop, CCSW '10* (pp. 1–6). New York: ACM. https://doi.org/10.1145/1866835.1866837
8. Hao, Z., Tang, Y., Zhang, Y., Novak, E., Carter, N., & Li, Q. (2015). SMOC: A secure mobile cloud computing platform. In *IEEE Conference on Computer Communications (INFOCOM)*. Piscataway: IEEE.
9. Hoang, D. T., Niyato, D., Wang, P., Wang, S. S., Nguyen, D., & Dutkiewicz, E. (2018). A stochastic programming approach for risk management in mobile cloud computing. In *Wireless Communications and Networking Conference (WCNC), 2018*. Piscataway: IEEE.
10. Hsueh, S. C., Lin, J. Y., & Lin, M. Y. (2011). Secure cloud storage for convenient data archive of smart phones. In *IEEE 15th International Symposium on Consumer Electronics (ISCE), 2011* (pp. 156–161). https://doi.org/10.1109/ISCE.2011.5973804
11. Huang, D., Zhang, X., Kang, M., & Luo, J. (2010). MobiCloud: Building secure cloud framework for mobile computing and communication. In *Fifth IEEE International Symposium on Service Oriented System Engineering (SOSE), 2010* (pp. 27–34). https://doi.org/10.1109/SOSE.2010.20
12. Itani, W., Kayssi, A., & Chehab, A. (2010). Energy-efficient incremental integrity for securing storage in mobile cloud computing. In *International Conference on Energy Aware Computing (ICEAC), 2010* (pp. 1–2). https://doi.org/10.1109/ICEAC.2010.5702296
13. Jia, W., Zhu, H., Cao, Z., Wei, L., & Lin, X. (2011). SDSM: A secure data service mechanism in mobile cloud computing. In *IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), 2011* (pp. 1060–1065). https//doi.org/10.1109/INFCOMW.2011.5928784
14. Khalil, I., Khreishah, A., & Azeem, M. (2014). Consolidated identity management system for secure mobile cloud computing. *Computer Networks, 65*, 99–110.
15. Khan, A. N., Mat Kiah, M. L., Khan, S. U., & Madani, S. A. (2013). Towards secure mobile cloud computing: A survey. *Future Generation Computer Systems, 29*(5), 1278–1299. https://doi.org/10.1016/j.future.2012.08.003
16. Lee, Y., Wang, L., & Gau, R. (2010). Implementation issues of location-based group scheduling for cloud applications. In *IEEE VTS Asia Pacific Wireless Communications Symposium Conference (APWCS 2010).*

17. Lockwood, J., McKeown, N., Watson, G., Gibb, G., Hartke, P., Naous, J., et al. (2007). NetFPGA–An open platform for gigabit-rate network switching and routing. In *IEEE International Conference on Microelectronic Systems Education, 2007. MSE '07* (pp. 160–161). https://doi.org/10.1109/MSE.2007.69
18. Lomotey, R. K., & Deters, R. (2013). Middleware-layer for authenticating mobile consumers of amazon s3 data*. In *Proceedings of the 2013 IEEE International Conference on Cloud Engineering, IC2E '13* (pp. 108–113). Washington: IEEE Computer Society. https://doi.org/10.1109/IC2E.2013.10
19. Nguyen, K. K., Hoang, D. T., Niyato, D., Wang, P., & Dutkiewicz, E. (2018). Cyberattack detection in mobile cloud computing: A deep learning approach. In *Wireless Communications and Networking Conference (WCNC)*, Piscataway: IEEE.
20. Olafare, O., Parhizkar, H., & Vem, S. (2015). A new secure mobile cloud architecture. arXiv preprint arXiv:1504.07563.
21. Panneerselvam, J., Sotiriadis, S., Bessis, N., & Antonopoulos, N. (2012). Securing authentication and trusted migration of weblets in the cloud with reduced traffic. In *Third International Conference on Emerging Intelligent Data and Web Technologies (EIDWT), 2012* (pp. 316–319). https://doi.org/10.1109/EIDWT.2012.20
22. Park, I.-S., Lee, Y.-D., & Jeong, J. (2013). Improved identity management protocol for secure mobile cloud computing. In *46th Hawaii International Conference on System Sciences (HICSS), 2013*. Piscataway: IEEE.
23. Perez, S. (2010). Mobile cloud computing: $9.5 billion by 2014.
24. Premnath, S. N., & Zygmunt, J. H. (2014). A practical, secure, and verifiable cloud computing for mobile systems. *Procedia Computer Science, 34*, 474–483.
25. Ramavathu, L., Bairam, M., & Manchala, S. (2017). A framework for secure mobile cloud computing. In *Proceedings of the First International Conference on Computational Intelligence and Informatics*. Singapore: Springer.
26. Satyanarayanan, M. (2010). Mobile computing: The next decade. In *Proceedings of the 1st ACM Workshop on Mobile Cloud Computing &#38; Services: Social Networks and Beyond, MCS '10, pp. 5:1–5:6*. New York: ACM. https://doi.org/10.1145/1810931.1810936
27. Subashini, S., & Kavitha, V. (2011). Review: A survey on security issues in service delivery models of cloud computing. *Journal of Network and Computer Applications, 34*(1), 1–11. https://doi.org/10.1016/j.jnca.2010.07.006
28. Wang, S., & Wang, X. (2010). In-device spatial cloaking for mobile user privacy assisted by the cloud. In *Eleventh International Conference on Mobile Data Management (MDM)* (pp. 381–386). https://doi.org/10.1109/MDM.2010.82
29. Xiao, S., & Gong, W. (2010). Mobility can help: Protect user identity with dynamic credential. In *Eleventh International Conference on Mobile Data Management (MDM)* (pp. 378–380). https://doi.org/10.1109/MDM.2010.73
30. Zhang, X., Schiffman, J., Gibbs, S., Kunjithapatham, A., & Jeong, S. (2009). 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). New York: ACM. https://doi.org/10.1145/1655008.1655026
31. Zhao, K., Jin, H., Zou, D., Chen, G., & Dai, W. (2013). Feasibility of deploying biometric encryption in mobile cloud computing. In *8th ChinaGrid Annual Conference (ChinaGrid), 2013* (pp. 28–33). https://doi.org/10.1109/ChinaGrid.2013.10
32. Zhou, J., Chen, J., Li, L., & Zhang, Z. (2012). The context awareness architecture in mobile cloud computing. In *Fifth International Symposium on Computational Intelligence and Design (ISCID)* (Vol. 1, pp. 302–305). https://doi.org/10.1109/ISCID.2012.83
33. Zhou, K., Afifi, M. H., & Ren, J. (2017). ExpSOS: Secure and verifiable outsourcing of exponentiation operations for mobile cloud computing. *IEEE Transactions on Information Forensics and Security, 12*(11), 2518–2531.
34. Zhou, Z., & Huang, D. (2012). Efficient and secure data storage operations for mobile cloud computing. In *8th International Conference on Network and Service Management (CNSM) and 2012 Workshop on Systems Virtualization Management (SVM)* (pp. 37–45).