J Supercomput (2017) 73:2558–2631
DOI 10.1007/s11227-016-1945-y

Data security in mobile cloud computing paradigm: a survey, taxonomy and open research issues

Tarunpreet Bhatia[1] · **A. K. Verma**[1]

Published online: 9 January 2017

Abstract The incessant spurt of research activities to augment capabilities of resource-constrained mobile devices by leveraging heterogeneous cloud resources has created a new research impetus called mobile cloud computing. However, this rapid relocation to the cloud has fueled security and privacy concerns as users' data leave owner's protection sphere and enter the cloud. Significant efforts have been devoted by academia and research community to study and build secure frameworks in cloud environment, but there exists a research gap for comprehensive study of security frameworks in mobile cloud computing environment. Therefore, we aim to conduct a comprehensive survey to analyze various cryptographic, biometric and multifactor lightweight solutions for data security in mobile cloud. This survey highlights the current security issues in mobile cloud environment and infrastructure, investigates various data security frameworks and provides a taxonomy of the state-of-the-art data security frameworks and deep insight into open research issues for ensuring security and privacy of data in mobile cloud computing platform.

Keywords Cloud computing · Mobile cloud computing · Data security · Access control · Authentication

1 Introduction

Mobile cloud computing (MCC) incorporates mobile computing, cloud computing and wireless networks where data processing and storage occur outside the mobile

✉ Tarunpreet Bhatia
tarunpreetbhatia@gmail.com; tarunpreet@thapar.edu

A. K. Verma
akverma@thapar.edu

[1] Department of Computer Science and Engineering, Thapar University, Patiala, India

device [1]. In MCC, the computation functionality and data storage for user applications are moved away from mobile devices to more powerful cloud computing platforms. MCC is one of the promising technologies for changing the world that provides pooled cloud resources toward unrestricted storage, utility and mobility to serve multiple mobile devices through Ethernet or Internet at any time and place in heterogeneous environments [2,3]. In order to have in-depth understanding of MCC, one should have clear understanding of cloud computing and its services. Cloud computing has redefined the meaning of computing by replacing a client-server which is an organizational centric model with more scalable, efficient and flexible data centric model [4]. The cloud renders its services to consumers by providing on-demand access to a shared pool of several computing resources such as server, storage area, applications in pay-as-you-use manner. There is no need to have powerful mobile device configuration such as processor and memory, and all the resource-intensive processing is performed on cloud. Since mobile devices are resource-constrained in terms of battery life, memory, bandwidth, etc., cloud service providers allow them to use infrastructure-as-a-service or IaaS (compute, network and storage), platform-as-a-service or PaaS (object storage, identity, queue, etc.) and software-as-a-service or SaaS (applications such as monitoring, finance, collaborative, ERP) at low cost and on-demand basis [5]. In SaaS, the consumers can only use service provider's applications that are running on a cloud. They are not responsible for managing underlying cloud infrastructure and computing resources (network, servers, operating system, and memory) except little user-specific configuration settings of the running application, e.g., salesforce.com. In PaaS, the consumer can deploy onto cloud consumer-created applications using tools and languages as provided by the service providers. The consumer is not responsible for managing underlying cloud infrastructure but has some level of control over applications deployed by consumers and their hosting environment configurations, e.g., Google applications. In IaaS, the consumer can deploy and run software (operating system and applications) on cloud. The consumer is not at all responsible for managing underlying cloud infrastructure but has full level of control over operating system, memory storage and deployed applications, e.g., Amazon's EC2. MCC still in its infancy is different from cloud computing with regard to mobility, bandwidth utilization, fault tolerance, security, etc. Table 1 lists different parameters that differentiate mobile cloud with cloud computing. The thorough knowledge of these parameters helps in dealing with limitations associated with MCC to provide reliable, fault-tolerant and secure mobile app experiences.

Mobile service is among the fastest-spreading technologies in the history. There are several applications that are benefitted by MCC and made a huge impact on the global market. Mobile commerce (m-commerce) has changed the lives of people by providing various applications such as finance, online ticketing, shopping on the mobile devices. These applications were facing challenges such as low bandwidth, low battery power, complex mobile architecture and security risks. MCC integrates m-commerce applications with cloud to overcome above challenges. Mobile learning (m-learning) that combines e-learning with mobility also faces several challenges such as low transmission rate and high cost of mobile devices. These can be overcome by utilizing cloud for large storage and high processing capabilities. Mobile health care allows mobile users to access medical resources in an efficient way as opposed to traditional medi-

Table 1 Comparison of cloud computing and mobile cloud computing paradigm

Parameters	Cloud computing	Mobile cloud computing
Device energy availability	Not required	Limited energy of mobile devices so resource-intensive operations are performed on cloud side instead on the mobile device
Bandwidth utilization	Not required	It should distribute the available limited bandwidth fairly among mobile nodes
Disconnected operation	Not required	Applications served from mobile cloud must support disconnections when mobile users/devices go out of range
Fault tolerance	The cloud needs to be fault-tolerant in events of hardware or software failures	The mobile cloud needs to monitor the connection strength continuously and needs to be fault-tolerant in events of frequent disconnections of mobile device or weaker connection signals
Network latency	Not a major issue	Mobile apps using mobile cloud are more sensitive to high network latencies and disconnected operations than cloud computing
Mobility	Not required	It must support high mobility of devices
Context and location awareness	Not required	Mobile apps must adapt to the frequent changes in the location of devices and collect context information from nearby devices. The applications must monitor the location and context to decide to offload or not to offload on cloud
Security	It must provide secure cloud services with resourceful devices	It must provide both mobile device security and secure cloud services in presence of resource-constrained mobile devices and insecure wireless communication medium

cal applications. The availability of on-demand services on cloud overcomes mobile medical application’s limitations such as low storage capacity, security and reliability of data. Mobile gaming (m-gaming) offloads game engine module to the resourceful cloud servers as it requires high computation and graphic rendering. Mobile apps can use the cloud for app development and app hosting. Mobile apps running on multiple mobile platforms such as Apple iOS, Android, Windows at the same time may rely on the cloud for storage, performing heavy computations, and fault tolerance to evade the repeated development and maintenance efforts.

The detailed MCC architecture is depicted in Fig. 1. The mobile devices such as laptops, PDA’s, handheld devices can access cloud services either through mobile net-

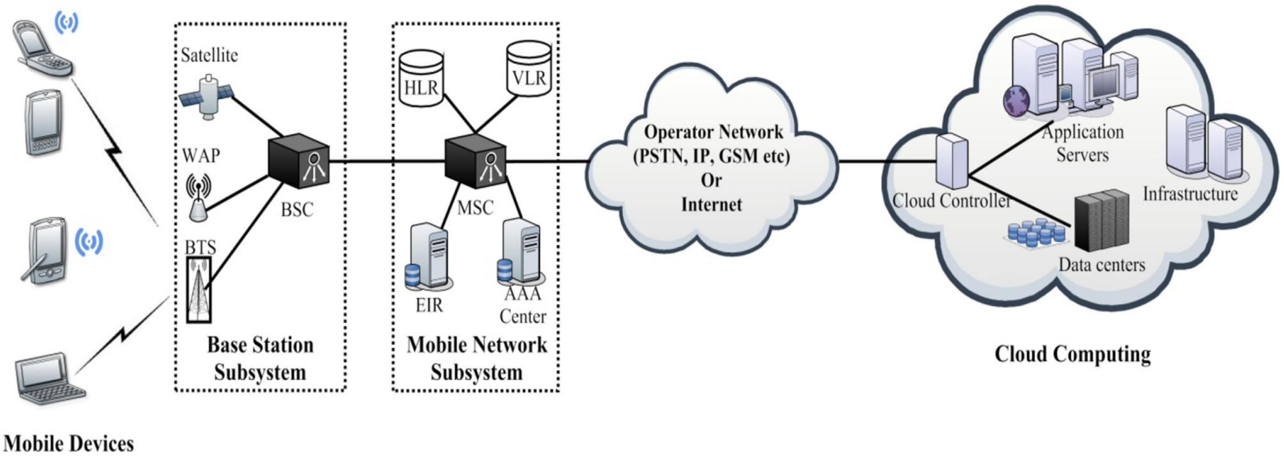

Fig. 1 MCC architecture

work or through wireless access points (WAP). Mobile devices are connected to the mobile networks via base transceiver stations (BTS) or satellites which are responsible for controlling the connections and functional interfaces between the mobile networks and mobile devices. They transmit the mobile users' requests and data to the base station controllers (BSC) which are further connected to mobile switching center (MSC) providing a wide range of mobile network services such as AAA (authentication, authorization and accounting) based on home location register (HLR), visitor location register (VLR), AAA center, equipment identity register (EIR) and subscribers' data stored in databases. The subscribers' requests are then delivered to a cloud through the Internet. In WAP case, the mobile devices connect to the access points through Wi-Fi which further connects to the Internet service provider (ISPs) to provide Internet connectivity. Wi-Fi-based connectivity is more efficient than mobile network GSM, GPRS, 3G, LTE, 4G connections as it provides low latency and consumes less energy. Inside the cloud, cloud controllers connect to data centers and application servers to process the requests and provide mobile users with the corresponding cloud services relying on ubiquitous computing, virtualization and service-oriented architecture.

One of the major issues, still unresolved, for cloud service providers and their clients is the difficulty in figuring out who is responsible for what security measures and controls. The service providers are responsible for creating services and features compliant with data protection and privacy standards on one side, and customer can configure and use those services in a way compliant with its industry and location on the other side. The service providers can create operational controls to protect customer's data on cloud, and customers have to use those controls to prevent unintended data sharing. The service providers are responsible for obtaining certifications and signing service-level agreements (SLA), whereas customers are responsible for verifying service provider's audit reports and certificates according to their organizational data privacy requirements. The boundary between these responsibilities is not clear and thus depends on the agreement signed between the customer and service providers and on the cloud service and deployment model used. For example, PaaS and IaaS providers share security responsibility equally for everything above the virtual machine layer with the clients. But with SaaS and cloud apps, the customer becomes more responsible for things such as access control and monitoring. In private cloud, customer is responsible for security at all the levels, whereas in public cloud, security responsibilities are shared between cloud service provider and customers. Customer is responsible for securing applications and data deployed on cloud platform and cloud service.

1.1 Motivation

Even though migrating to the mobile cloud is a tempting trend from a financial perspective, there are several other aspects that must be taken into account by cloud service providers for providing cloud services and organizations before outsourcing their sensitive data to CSPs. According to Gartner Inc. [6], "The ability to integrate business applications on smartphones, tablets and other wireless devices is predicted to accelerate Mobile SaaS adoption in the corporate business environment that will grow to $3.7 billion by 2016. By 2016, security will be a top 3 business priority for 70% of CEOs of global enterprises". The cloud-based security market remains a viable one, offering providers many opportunities for expansion. The encryption is a new area of growth, and it remains a complex and cumbersome job. According to Cisco VNI Mobile Data Traffic Forecast [7], "Global mobile data traffic will grow 8-fold nearly from 2015 to 2020 with compound annual growth rate (CAGR) of 53%, reaching 30.6 exabytes per month by 2020. 67% of mobile devices will be smart devices by 2020 as compared to 36% in 2015. People all over the world are consuming more and more wireless bandwidth to manage various tasks. The vast majority of mobile data traffic (98%) will originate from these smart devices by 2020, up from 89% in 2015". According to a report from Markets and Markets [8], the mobile market is expected to grow from USD 1.18 billion in 2015 to USD 3.26 billion by 2020, at a CAGR of 22.5% from 2015 to 2020. The cloud security market is expected to be worth $8.7 billion by 2019. The tremendous increase in demand for smartphones and tablets has a parallel demand for IT solutions to speed up applications development for mobile computing while ensuring security is in place. Protecting user privacy and data secrecy from an adversary is an essential factor for the success of MCC paradigm. Since mobile devices are resource-constrained, protecting them from numerous security threats is more difficult than that for resourceful devices. The scarcity of efficient security solutions necessitates the utmost need to conduct a comprehensive survey to acquire deep insight into this field. Therefore, we aim to conduct a comprehensive survey to assess and analyze various cryptographic, biometric and multifactor solutions for data security in MCC and presents state-of-the-art taxonomy. This survey paves way for future research and technological improvements for wide deployment MCC.

1.2 Related surveys

Researchers have studied varied aspects of MCC such as architecture [3,9], challenges [10–12], application models [13–15], energy saving [16], computation overloading [17–20], heterogeneity [2], virtualization [2], resource management [21], security and privacy issues [22,23]. Several studies have been conducted for data security in cloud computing [24–29] to build a level of trust between cloud service providers and consumers, but security issues for MCC environment have not been explored much, and there exist many challenges in the security policies as MCC is still in the preliminary stages of research. Kumar and Rajalaxmi [30] analyzed the mobile cloud security issues and vulnerabilities of mobile cloud devices and highlighted the usage of SCWS (Smart Card Web Services) as a security solution for mobile cloud computing. Khan et al. [31] investigated lightweight data security and application security frameworks in MCC. However, their study was limited to cryptographic techniques, whereas present work highlights cryptographic, biometric and multifactor solutions. Alizadeh et al. [32] surveyed the state-of-the-art user-side and cloud-side authentication methods only in MCC and evaluated the solutions based on usability, efficiency, security and privacy as evaluation metrics. The comprehensive study of data security schemes in MCC ensuring confidentiality, integrity, authentication, non-repudiation and access control, which is crucial for developing future secure solutions, is lacking and demands further efforts. To the best of our knowledge, there exist two surveys [31,32] exploring various possible security solutions in MCC environment. The present work encompasses various security goals such as confidentiality, integrity, authentication, access control and highlights merits and demerits of various conventional schemes. The shortcomings in the conventional schemes could open new roads for researchers to design an impenetrable panacea. This survey also presents state-of-the-art classification of data security schemes and cryptographic techniques in an innovative demarcation on chronological order. Our main contributions are as: possible threats and attacks associated with mobile cloud, classification and critical investigation of various data security schemes, open research issues in MCC environment that ground future researches.

1.3 Paper organization

The remainder of this paper is organized as follows: Section 2 discusses security issues, vulnerabilities, associated threats and attacks in MCC. Section 3 provides classification taxonomy and surveys the state-of-art data security schemes in MCC encompassing cryptographic, biometric and multifactor solutions in detail. It also includes comparative description of security frameworks, their strengths and weaknesses. The open research issues are discussed in Section 5. Finally, Section 6 concludes our survey.

2 Security issues in mobile cloud computing

The widespread evolution of MCC needs secure, reliable and non-repudiated identification and authentication mechanisms. Security has emerged as an obstacle in adoption of cloud and mobile cloud computing [6,19] even though it delivers a wide range of resources. The basic security principles that any data security framework must comply are confidentiality, integrity, authentication and non-repudiation. Apart from these, there are two more principles, availability and access control which are linked to entire system as a whole and not to particular data or message.

- Confidentiality: This principle ensures that only sender and recipient(s) should access the message. It prevents unauthorized access to data, and loss of confidentiality leads to interception.
- Integrity: It ensures correct delivery of message to the intended recipient(s) without any alteration. Loss of integrity leads to modification attack.
- Authentication: It helps in establishing proof of identities that the communicating node is what it claims to be. Absence of authentication measures leads to fabrication attack.
- Non-repudiation: This principle makes sure that sender cannot deny later on for not sending the message.
- Access control: It ensures the use of network's resources and services by only the authorized users. It acts like a bridge between confidentiality, integrity and authenticity. It begins with authentication and then identifies who can "access" what, where access comprises reading of data (confidentiality) and writing (integrity).

- Availability: It ensures that authorized parties are able to access the information when desired. Denying access to information leads to denial-of-service attack in which legitimate users are denied access to resources.

In this section, we have addressed several security issues, threats and possible attacks. These issues have been classified in Fig. 2.

2.1 Infrastructural and architectural issues

This dimension includes virtualization security, network security, data segregation issues, insider attacks and administrative interface issues.

- **Virtualization security:** Virtualization, one of the major components of cloud computing, allows multiple users to store data using applications provided by SaaS providers and share on-demand services. This poses high potential of intrusion into tenants' data if their data are not segregated properly at physical as well as application level. It leads to many risks such as isolation between multiple virtual machines (VMs) running on same physical machine, Cross-VM attacks, injecting malicious code into the application, virtualization software known as hypervisor vulnerabilities that can be exploited to bypass authentication and escalate privileges [31]. For example, vulnerable software Microsoft Virtual PC and Microsoft Virtual Server allow guest machines to run malicious code on host machine or other guest machines. VM Escape attack allows an attacker to break an isolation layer to run an application with hypervisor's root privileges and get access to host OS and other virtual machines running on the host machine. Rootkit is a set of programs or tools used by an attacker to gain administrator-level privileges on a machine either by cracking password or by exploiting hypervisor's vulnerability. VM-based rootkits allow an attacker to execute malicious code to destroy anti-malware programs to escape detection and control over any VM running on physical compromised machine. To defend against such attacks, comprehensive monitoring of hypervisors and strong isolation between VMs are required which curbs an attacker to inject malicious code into neighbor's VM.
- **Network security:** It deals with network communications and configurations. The inherent vulnerabilities of Internet protocols such as ARP, HTTP, TCP allow an attacker to exploit cloud system and its resources through man-in-the-middle (MITM) attack, ARP poisoning, DNS poisoning, session hijacking, etc. Data are obtained from an enterprise and stored on cloud so the strong network encryption techniques such as SSL and TLS are required to protect against such attacks. Improper configuration of SSL leads to MITM attack in which an attacker can access the data communication between two parties such as data centers, VMs. The malicious attacker can redirect all inbound and outbound traffic from other VMs to its own machine through ARP poisoning attack since ARP does not require proof-of-origin. DNS poisoning attack is tricking the domain name server (DNS) to send traffic in the wrong direction by modifying DNS cache content maliciously. The cloud customers must ensure that cloud service providers are taking proper steps to secure their DNS infrastructure. In session-hijacking attack, attacker exploits improper implementation of session ID's in HTTP and assumes

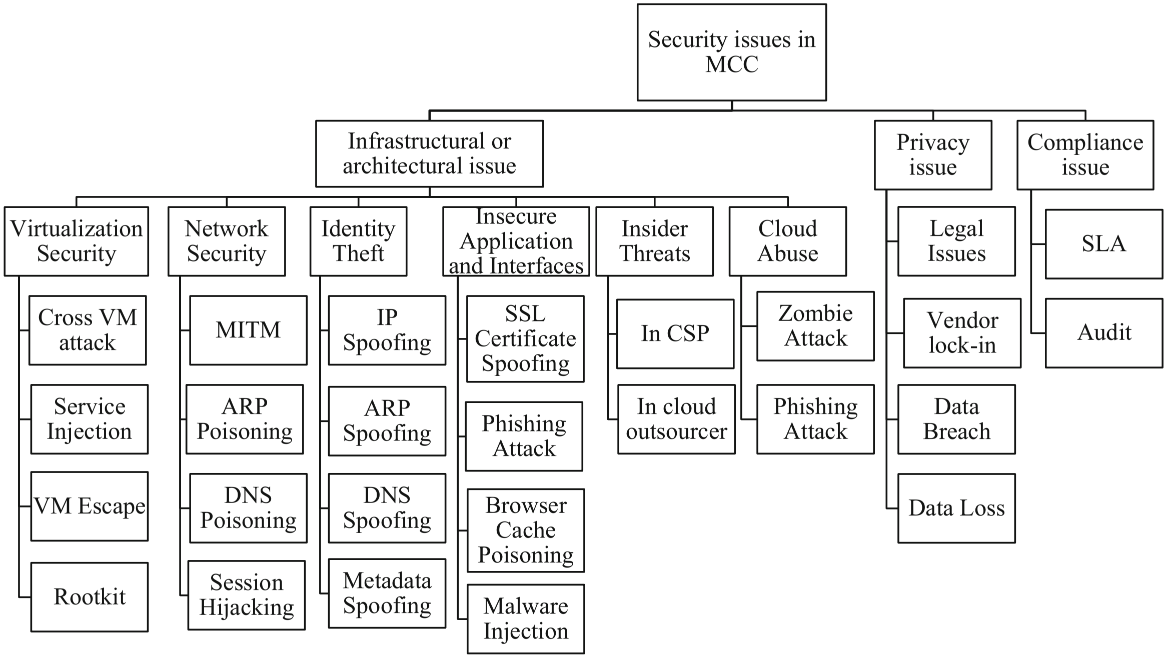

Fig. 2 Security issues in mobile cloud computing

user identity for further communication. Anonymous authentication and encrypting traffic can thwart such attacks. Firewalls and protocols must be configured to provide required level of security for cloud environment.

- **Identity thefts:** It is a form of fraudulent act in which an entity pretends to be someone else to access resources or obtain banking and other critical information. DNS spoofing, IP spoofing, ARP spoofing, metadata spoofing, phishing attacks are all forms of identity thefts. An intruder obtains the IP address of a legitimate user and alters TCP/IP packet headers to masquerade as a trusted host and hides its identity in order to launch IP spoofing attack. This attack can be used to hijack browser, overload targets with traffic and steal information. In an ARP spoofing attack, an intruder binds its MAC address to IP address of a legitimate VM in a network and sends spoofed ARP messages that resulted in data intended for the legitimate host's IP address sent to the intruder VM attached to same virtual switch instead. Malicious attackers use ARP spoofing to gain access to sensitive information or network resources, modify data-in-transit or block traffic on a LAN. In DNS spoofing attack, an attacker machine supplies false DNS information to a VM so that when it browses a particular site, browse request is redirected to a fake IP address created by an attacker to steal banking credentials. Web services description language (WSDL) file contains metadata or descriptions about services offered so that an individual can access those services electronically. In metadata spoofing attack, intruders can steal information or inject malicious code by modifying a service's WSDL file at service delivery time. In phishing attack, an adversary sets up a fake URL identical to real Web application fooling the users to enter a valid credentials and certificates. According to Financial Fraud Action UK, financial fraud largely driven by identity thefts corresponds to £755m in 2015, 26% rise from 2014. Recently in 2016, Collins of Lancaster, Pennsylvania, has been accused of gaining access to more than 100 Apple iCloud and Gmail accounts illegally by sending phishing mails to clients. After this incident, Apple recommended to enable two-factor authentication as passwords can be easily compromised by hackers. Strong remote authentication and authorization mechanisms should be enforced for accessing sensitive information stored on cloud.
- **Insecure application and interfaces:** Cloud service providers rely on user interfaces for exploring resources and tools available by CSPs; administrative interfaces for VM management, deployment, coding, testing, monitoring, user access control and configuration and programming interfaces for accessing virtualized resources and service provisioning. Weak user interfaces and API can expose an organization to several security risks. Somorovsky et al. [33] had provided security analysis of control interfaces of Amazon and Eucalyptus that they can be easily compromised through signature wrapping and XSS techniques. The defects in design and architecture of applications lead to malware injection attacks such as SQL injection, OS injection, XSS (cross-site scripting) injection and LDAP (lightweight directory access protocol [34]) injection attacks. An adversary compromises cloud system by injecting malicious code in a service or malicious virtual machine instance which may modify or block service functionalities. The legitimate requests are thus redirected to malicious services. For example, in Amazon EC2 API, an adversary can introduce malicious command instead of security group name [35]. SSL protocol

ensures secure communication between user's browser and server through the use of SSL certificates which can be verified by user's browser. If server's SSL certificate is vouched by any of the trusted CA's pre-listed in browser, then secure communication is established otherwise warning to user is issued. SSL secure connection can be broken by impersonating legitimate server through SSL certificate poisoning attack to intercept sensitive information. In browser cache poisoning attack, an attacker performs MITM attack on HTTPS session of a user and replaces cached content with malicious data. A reliable end-to-end encryption, data validation checks, secure user interfaces, strong authentication and access control policies can avoid such attacks.

- **Insider attack:** Mostly, organizations and cloud service providers focus more on protection against external attackers and less on insider intruders. An inside attacker can easily harms the network or system as attacker has an authorized access to system and is much familiar with network architecture and system security procedures. The insider employee can be either a malicious employee working for the cloud provider or an employee of an organization which is utilizing cloud services. Coherent demarcation of duties and transparent employee management policies can reduce the extent of damage caused by insider attacks.
- **Cloud abuse:** It means using cloud services for malicious intent such as for breaking encryption keys, sharing pirated software or propagating malware launching DDoS attacks and phishing attacks as it was difficult to perform this using standard computer. Proper validation/verification checks during initial registration phase and constant monitoring of network traffic can prevent such attacks.

2.2 Privacy issues

This dimension includes data security, data loss and legal issues. The data of customers and services by providers are hosted at different geographic locations so it is affected directly or indirectly by multiple jurisdictions or subpoena law enforcement measures. Governance issues such as vendor/data lock-in, data and security control arise due to dissimilarities in underlying cloud architectures and loss of administrative controls of a customer in cloud environment. The availability of large variety of cloud servers and mobile devices on the one hand and lack of well-established uniform standards and APIs on the other hand aggravate vendor lock-in issue in MCC. Law enforcement also results in disclosure of hardware device as demanded by court of law for a particular customer, but this will in turn affect all the customers whose data were stored on that particular device [36]. Breaching into the cloud platform that stores data of various users and organizations can attack the data of all users compromising confidentiality, integrity and authenticity (CIA). The major data security challenges for achieving CIA are key management, identity management, access control and remote integrity check. Data confidentiality ensures that information is available only to authorized users. Data integrity means that data should not be tampered or altered accidently or intentionally. Data authenticity ensures that both parties involved in communication are who they claim to be. Data loss means access to data in the events of CSP's carelessness, malicious attack, power outages, natural disaster such as earthquake, fires or floods.

It can turn into problematic situation for CSPs when data storage complies by certain laws such as HIPAA (health insurance, portability and accountability act). The efforts put to mitigate data loss can exacerbate data breach and vice versa. We can encrypt data to reduce impact of data breach, but if encryption keys are lost, data are gone. On the other hand, if offline backup is kept for valuable data to reduce data loss, there is a high potential of exposure to data breaches. Solutions include strong encryption policies, data integrity checks, secure storage of encryption keys and efficient data backup strategies.

2.3 Compliance issues

This dimension deals with service availability and audit capabilities. Service-level agreements (SLA) ensure required service availability and procedures to be adopted to guarantee certain level of security [37]. Auditing is done by customers, service providers and third parties to continuously assess security and availability services. SaaS providers should consider certain regulatory and legislative frameworks while storing privacy-related information in cloud such as HIPAA, Gramm-Leach-Bliley (GLBA), the Fair and Accurate Credit Transaction Act (FACTA), Privacy Act of 1974, Computer Security Act of 1987.

Table 2 summarizes above-mentioned potential security issues or threats to MCC, their effects on various cloud services and relevant mitigation techniques.

3 Data security in MCC

While securing data in the cloud, we need to figure out possible states in which the data may occur and available controls for that state. With the proliferation of Internet and cloud computing in recent years, protecting data-at-rest is considered as important as protecting data-in-transit [38]. The encryption protects data in the cloud from security breaches, third-party disclosures and compliance violations. Encryption is considered important for all the three possible states of data: in-transit, at-rest and in-use as shown in Fig. 3. The choice of strong encryption algorithms and key management policy is critical for success of any encryption policy. For better security, customers can adopt their own key management infrastructure instead of adopting CSP's default key management infrastructure. Skyhigh Networks Inc., recently analyzed encryption controls of 12000 cloud providers and made a statement that 81.8% of CSP encrypt data-in-transit protecting data from MITM attacks as it moves through Internet, but only 9.4% of CSP encrypt data-at-rest, making it vulnerable to unauthorized access, data breaches and blind government subpoenas. Among them, only 1.1% of CSP use customer-managed encryption keys. The most popular apps such as Facebook, Twitter, PayPal, Gmail, LinkedIn, eBay store data (user credentials, payment card numbers, bank account numbers) in an unencrypted form. Ebay suffered biggest data breach in 2014 when 145 million account credentials were stolen. The possible states of data are as follows:

- **Data-in-transit:** Data including voice, video, text, metadata are thought to be in motion once it leaves an enterprise control and moves across the network or

Table 2 Summary of threats, associated attacks to MCC and mitigation techniques

Threats	Effects	Cloud service affected	Possible attacks	Attack surface	Mitigation techniques
Virtualization security	Allow an attacker to gain access over another user's VM, provide malicious service to user	PaaS, IaaS	Cross-VM, VM Escape, Rootkit, Service Injection	Compromising hypervisor and service identification files	Securing hypervisor, comprehensive monitoring at hypervisor level, VM isolation, IDS/IPS (Intrusion Detection/Prevention System), Data validation checks
Network security	Affects the data security and privacy	SaaS, PaaS, IaaS	MITM, ARP Poisoning, DNS Poisoning, Session Hijacking	Accessing data communication between two parties	Anonymous authentication, Encryption, SSL, TLS
Identity theft	Allow an attacker to be someone else or access their resources	SaaS, PaaS, IaaS	IP Spoofing, DNS Spoofing, ARP Spoofing, Metadata Spoofing	Weak password recovery methods, key loggers	Strong authentication mechanisms
Insecure applications and interfaces	Clear text authentication and improper authorization	SaaS, PaaS, IaaS	SSL Certificate Spoofing, Phishing, Attack on browser cache, Malware Injection	Unauthorized access to services and management interface	Strong authentication and access control policies, Hash-based service integrity check, Secure browsers and API
Insider threats	Affect confidentiality, integrity and availability of resources	SaaS, PaaS, IaaS	In CSP or cloud outsourcer	Gaining access over confidential data and cloud services	Transparent employee management process including compliance reporting and breach notification, segregation of duties

Table 2 continued

Threats	Effects	Cloud service affected	Possible attacks	Attack surface	Mitigation techniques
Cloud abuse	Affect service availability and privacy of user's sensitive information	SaaS, PaaS, IaaS	Zombie, Phishing	Allow intruder to compromise valid user's VM, allow user to access fake link, lack of validation	Strong authentication and authorization, constant monitoring of network traffic against intruders
Data breach and loss	Unauthorized disclosure of private data behind applications	SaaS, PaaS, IaaS	MITM	Weak encryption algorithms, lack of authentication, authorization, weak keys and their storage, unreliable data center	Strong encryption policies, secure storage of keys and API's, efficient data backup and retention policies

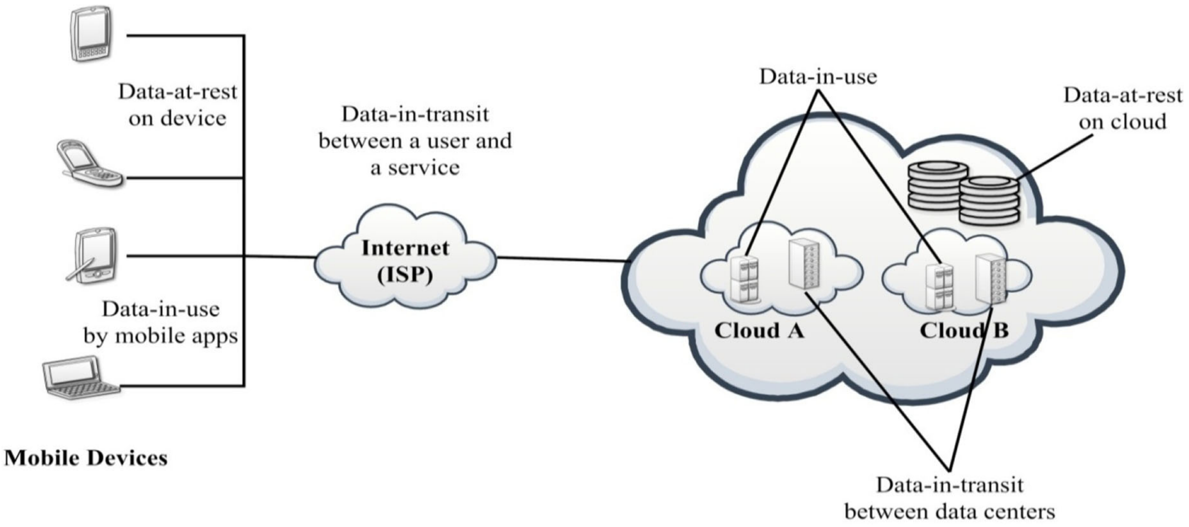

Fig. 3 Possible states of data

to cloud and vice-versa, and thus, its encryption is vital. It involves not only communication with a component outside the cloud service, but also communication between virtual networks. It has to be protected against eavesdropping attack through cryptographic protocols such as SSL or TLS by establishing an encrypted and authenticated channel.

- **Data-at-rest:** It refers to inactive data which is stored physically on NAS, SAN, file servers in the form of databases, data warehouses, off-site backups, etc. In addition to encryption, strong access control policies and data federation should be used to thwart attacks.
- **Data-in-use:** It refers to the dynamic data which are stored in non-persistent state, e.g., data or encryption keys in cache, main memory, transactions in message queue, data currently processed by an application. These data are generally in clear text form for performing value-added functions such as searching, retrieval on the data, but cloud security alliance now recommends encryption of data-in-use for better security. Fully homomorphic encryption allows computations on ciphertext, and results thus obtained when decrypted match the computations performed on plain text. Enclaves are used to secure data-in-use in which data are in encrypted form in RAM but available as clear text inside CPU [39].

There is a need of securing data of an organization or enterprise on cloud to ensure confidentiality and privacy that facilitates the need of efficient key management schemes while using cloud SaaS. With the widespread use of sophisticated hacking tools, the data stored in the cloud is at increasing risk of malicious attack [40]. So, there should also be robust authentication schemes for cloud computing to ensure access to data by legitimate and authorized users only. SaaS providers must verify the identity of every single user attempting to access the cloud system. A systematic review is presented in Fig. 4 to complement the work of various researchers with reference to data security and authentication mechanisms in MCC environment. Data security schemes in MCC are classified as:

- **Password-based schemes:** Most of the current systems are based on password-based authentication in which server maintains database for passwords or their hash values after salting. Lamport [41] proposed first one-time password scheme in 1981 for remote authentication. They are easily deployable, but attacker can easily steal, guess or modify the passwords stored in the server. It also suffers from scalability issue. Recently, Ren et al. [42] proposed a lightweight, fine-grained and flexible scheme for access control (F2AC) in multiparty file sharing in mobile cloud computing environment. It supported dynamic operations such as adding or deleting users within an ad hoc group, authorizing or revoking privileges for members transitively and segregating access authentication from system authentication. It consisted of two lists: access control and user authentication list, both maintained at cloud. The access control list is comprised of filename, users, privileges (read, update, modify, create) and conditions (location, MAC addresses, etc.), whereas user authentication list contained username and corresponding token or password. The creator of file assigned and sent token to other users with whom it wants to share file. It is vulnerable to impersonation, password guessing and

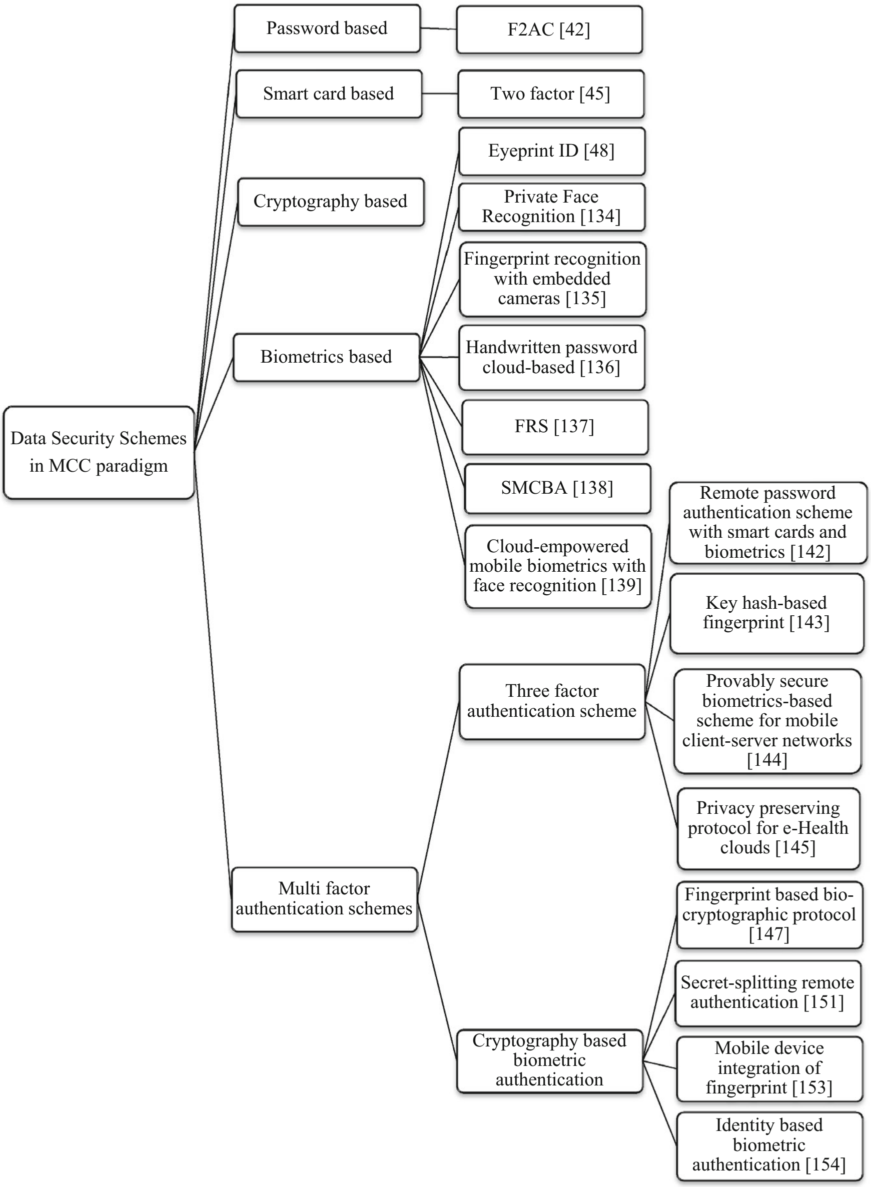
Data Security Schemes in MCC paradigm
Password based
F2AC [42]
Smart card based
Two factor [45]
Cryptography based
Biometrics based
Eyeprint ID [48]
Private Face Recognition [134]
Fingerprint recognition with embedded cameras [135]
Handwritten password cloud-based [136]
FRS [137]
SMCBA [138]
Cloud-empowered mobile biometrics with face recognition [139]
Multi factor authentication schemes
Three factor authentication scheme
Remote password authentication scheme with smart cards and biometrics [142]
Key hash-based fingerprint [143]
Provably secure biometrics-based scheme for mobile client-server networks [144]
Privacy preserving protocol for e-Health clouds [145]
Cryptography based biometric authentication
Fingerprint based bio-cryptographic protocol [147]
Secret-splitting remote authentication [151]
Mobile device integration of fingerprint [153]
Identity based biometric authentication [154]

Fig. 4 Taxonomy of data security schemes in MCC

replay attack. Problems with passwords highlight the need for another system of user identification.

- **Smart cards based:** Smart cards possess an individual's identity-related information in encrypted form on an embedded chip. On-chip defense measures minimize identity thefts. Individual smart cards can be programmed for multiple uses, e.g.,

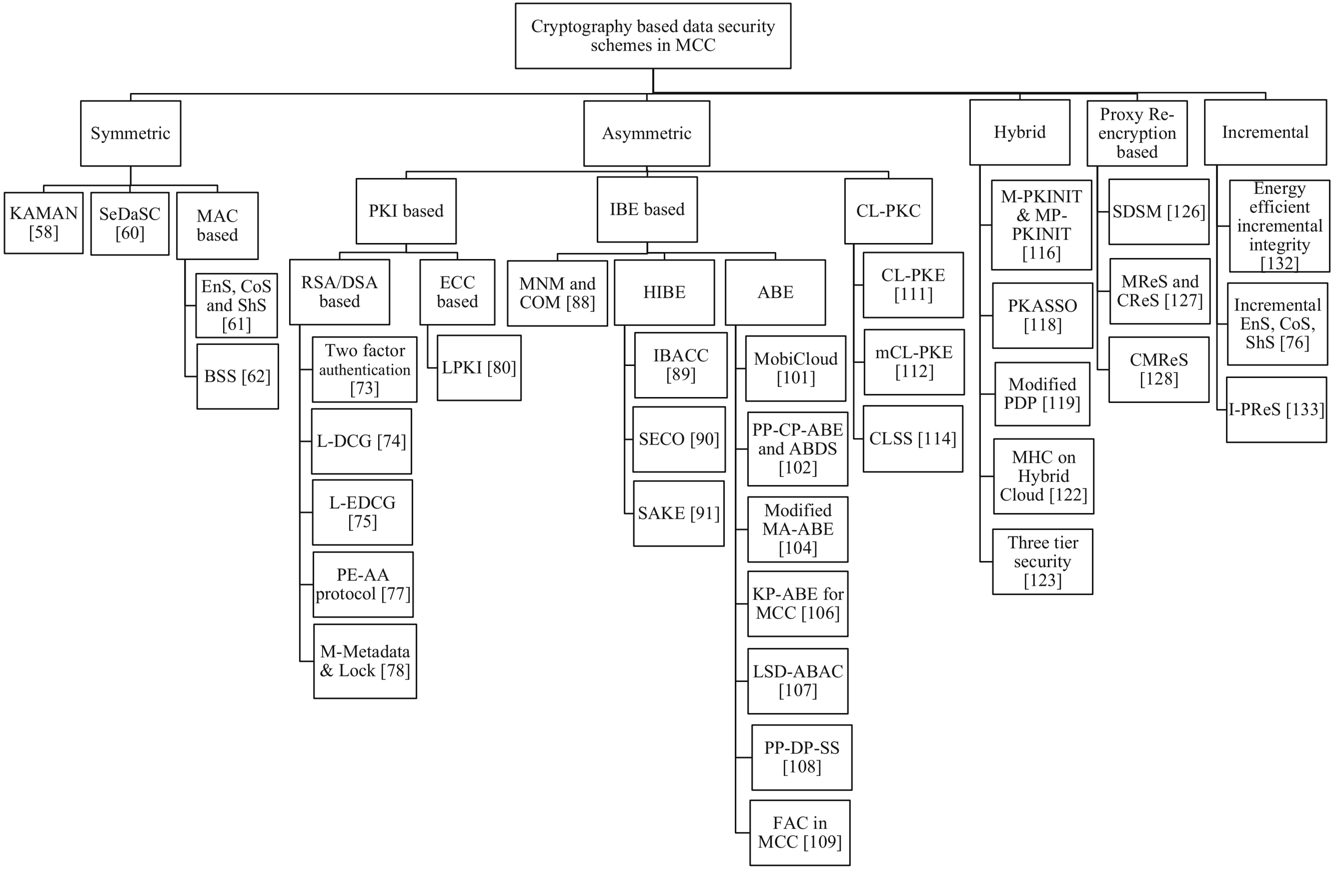

Fig. 5 Classification of cryptography-based data security schemes in MCC

banking transactions, medical entitlement. But the information stored on smart cards can be revealed [43].

Shoup and Rubin [44] proposed three-party key distribution protocol where smart cards are used to store the long-term keys of users. If a smart card is compromised by an adversary, then user is also compromised. Yang et al. [45] proposed two-factor mutual authentication scheme based on smart cards and passwords. Halevi and Krawezyk's protocol [46] is used for password-based one-way authentication protocol. Each client shared symmetric key and password with server which is saved by the server in two different tables and smart card is used by the client to store symmetric key. The password-based one-way authentication protocol is transformed to password-based mutual authentication and key exchange protocol (PWAKE) which is further upgraded to two-factor smart card-based password mutual authentication and key exchange protocol by using pseudorandom functions and collision-resistant hash functions. Hwang and Li [47] proposed smart card-based authentication scheme relying on ElGamal public cryptosystem. But it provides one-way authentication.

- **Cryptography based:** It is a conventional way of maintaining privacy of data and authenticating users over an insecure networks. The individual or group of users possessing correct cryptographic key have access to encrypted data. A brief classification of various cryptography-based data security schemes in MCC is given in Fig. 5.
- **Biometrics based:** In cryptography-based schemes, an attacker can obtain keys in illegal ways and pretend to be a genuine user. A very reliable and natural solution to prevent such attacks is to use biometric traits, which recognize users through physiological and behavioral characteristic possessed by them. Your fingers, your eyes and voice are always with you, and these cannot be imitated or possessed by others. But it might be possible that biometric templates such as fingerprint are acquired stealthily and used by a malicious user. There are several cloud-based biometric recognition solutions available in the market such as Eyeprint ID [48], BioID [49], Face.com [50], Animetrics FaceR [51], Calsoft Labs [52]. According to recent research by New York-based firm, 240 million users or businesses will continue to use cloud services through mobile devices by 2015, increasing the revenue of mobile cloud computing to $5.2 billion.
- **Multifactor authentication schemes:** They are a combination of two or more schemes in an effort to provide more efficient data security and authentication schemes. There is always a threat of password being cracked even if biometric information is protected using password in some multifactor authentication schemes. It is feasible to integrate biometrics to cryptographic infrastructure. Biometric authentication appearing to be more reliable solution than other traditional authentication measures is also vulnerable to attacks when it comes to remote authentication over open networks.
- **Intrusion detection based:** Cryptographic solutions are proactive and computation-intensive. They perform well for anticipated attacks but may collapse under unknown and unanticipated attacks. The increasing popularity of smartphones by offering advanced computing has given birth to both generic malware (viruses, Trojans, worms) and smartphone-specific malware that may exploit Bluetooth

interface or voice-recognition algorithms. So, we need second wall of defense, i.e., intrusion detection mechanism for detecting and reporting unusual behavior. An effective intrusion detection mechanism should save all malware signatures in phone, but it will consume large computation and memory resources of mobile devices. Zonouz et al. [53] developed Secloud, a cloud-based lightweight security solution for smartphones. It emulates a smartphone in a cloud and keeps it synchronized by sending the device inputs and network connections continuously to the cloud. It consists of 3 main entities: client agent which is a lightweight software running on smartphones for collecting user and sensor inputs from the device and passing to emulated replica on cloud; proxy server for mirroring network traffic between smartphones and Secloud's replica; and emulator in cloud running various host-based and network-based security solutions such as virus scanners, file integrity checker, network-based IDS, snort. It performs security analysis using intrusion detection techniques on emulated replica instead of device itself, thereby reducing the energy and processing power requirement of mobile devices. However, deletion of log values by an attacker is still an issue. It uses run-length encoding as lossless data compression algorithm to prevent network and storage resources. Secloud software agent modifies input subsystem of a device to capture and log physical events after they are processed by drivers and then pass them to handlers. It makes use of cryptographic hash function chains to ensure integrity of logged events and stored pairs of event and hash value. It initiates a kernel-based socket that connects to emulation environment when data plan is active; otherwise, logs are stored on phone's SD card.

3.1 Cryptography-based security schemes in MCC

This section provides an overview of commonly used cryptographic techniques as shown in Fig. 5 in mobile cloud environment and how these techniques deal with different security goals and reasonable network performance.

3.1.1 Symmetric or private key cryptography

It was developed in 1970s in which encryption and decryption are done by common shared key [54]. It is relatively easy and fast to implement, but secret key agreement between parties is a problematic issue. The robustness and efficiency of symmetric key depend on the key length used and secure transmission of keys between the parties. This cryptography fails to provide non-repudiation and authentication mechanisms. If an attacker is successful in compromising symmetric key, then both sides of conversation get exposed and all the messages encrypted with that key are exposed. Kerberos is an authentication protocol based on symmetric key cryptography to allow workstations to share resources in a secure manner [55]. Since symmetric key operations incur less overhead, it provides shorter authentication latency as compared to PKI. In Kerberos, a trusted third party is responsible for certifying communicating parties to one another. The popular symmetric key algorithms [56] are data encryption standard (DES), advanced encryption standard (AES), blowfish, etc. Message authentication code (MAC) relies on symmetric encryption to authenticate originator of a message. MAC-based algorithms takes message and shared symmetric key as input and outputs a MAC value. This allows an individual to verify whether an adversary has tampered with contents of a message or file. MACs can also be built from keyed hash functions (MD, SHA) known as HMAC [57]. Encryption ensures confidentiality, and MAC ensures integrity or authentication so we need to combine both cryptographic primitives to achieve these security principles.

Pirzada and Donald [58] proposed a secure key exchange protocol called KAMAN based on four-pass Kerberos applicable in mobile ad hoc networks [59]. KAMAN employs replication and election mechanism to ensure stable and prolonged connectivity between clients and servers. It consists of multiple servers for authentication and load distribution and eliminates the use of Ticket Granting Server to make it suitable for mobile environment. Mobile clients know the password or secret key and servers store hash value of these passwords. The servers share a secret key among each other to replicate their databases periodically or on demand to withstand network or system failures. The mobile client C1 wants to communicate with C2 in a secure way so it sends authentication request to one of the server S1. S1 will reply back C1 with a ticket containing the session key for requested secure connection. C1 sends this ticket to C2. This ticket acts as a certificate to C2 that a C1 is really who it claims to be. The C2 acknowledges the ticket by sending a timestamp, and a secure communication session is established between C1 and C2 using session key provided by S1.

Ali et al. proposed secure data sharing in clouds (SeDaSC) [60] methodology applicable in both conventional and MCC environment based on symmetric encryption without the use of computation-intensive operations. It provides confidentiality, integrity, forward and backward access control. It involves three entities, namely data owner, trusted third party known as cryptographic server on cloud (CS) and cloud for storage. The data owner sends the data file, the list of the users sharing the data file and the parameters required for generating an access control list to the CS. CS is responsible for encrypting, decrypting, key management and maintaining access control lists. CS generates the symmetric key and encrypts the data before storing on cloud. CS splits the symmetric key into two key shares for each user: One share is transmitted to user, and other key share is stored within the access control list (ACL) related to the data file. Whenever a person leaves a group, CS removes its records from the ACLs of the related files without re-encrypting the files. As the whole key is not possessed by that member, he cannot decrypt any of the data files. The entire key is not shared with the group members by CS to thwart against insider attack so that malicious user within a group cannot decrypt, modify and re-encrypt the data.

Ren et al. [61] proposed three different distributed schemes: encryption-based scheme (EnS), coding-based scheme (CoS) and sharing-based scheme (ShS) for providing confidentiality and integrity of mobile users' files which are stored on single or multiple cloud servers. The three entities involved are mobile device, cloud server and user operating mobile device. These schemes assume that mobile device is semi-trusted, i.e., computations are trusted, but storage is not trusted, and stored data may be lost or stolen. Cloud servers are distrusted, but wired/wireless link security between mobile device and cloud servers is provided by traditional MAC or IP layer protocols such as IEEE 802.11, IPSec, TLS.

- Encryption-based scheme (EnS): In this scheme, mobile device performs file encryption and integrity verification. Before uploading any file, mobile device prompts for a password from user denoted by PWD. Mobile device generates encryption key EK by applying hash function to password, file name FN and file size FS and integrity key IK as follows:

$$EK = H\,(PWD\,||FN||\,FS)$$
$$IK = H\,(FN\,||PWD||\,FS)$$

It then encrypts the file contents F with EK and generates authentication code MAC as:

$$FILE' = E\,(FILE, EK)$$
$$MAC = H\,(FILE, IK)$$

It then sends $(FILE'||H(Fn)||MAC)$ to cloud server for storage. The mobile device stores just filename and deletes EK and IK.
While downloading a particular file FN, mobile device sends $H\,(FN)$ to cloud server. The server searches and sends back $(FILE'||MAC)$ corresponding to matched $H\,(FN)$. The mobile device then prompts user for entering the password corresponding to that file. The device generates EK and IK in a similar manner. The mobile device decrypts as $FILE = D\big(FILE', EK\big)$ and verifies MAC values.
- Coding-based scheme (CoS): CoS uses a lightweight computation operation to preserve privacy of user's data instead of encryption operation. It assumes multiple cloud servers for storing a file in a distributed manner. The user file F is divided into f multiple parts. Each part further has c chunks with b bits each. The mobile device generates coding vector α = [α_1, α_2 ...., α_c] by computing:

$$\underset{1}{\alpha} = H\,(PWD\,||FN||\,FS)$$
$$\underset{i+1}{\alpha} = H^{i}\,(\alpha_i)\,where\ 1 \le i \le c$$
$$IK = H\,(\alpha_1||\alpha_2\,||\ldots||\,\alpha_c)$$

The mobile device codes each part by using vector α as:

$$F'\,[j] = \sum\nolimits_{i=1}^{c} \alpha_i * F\,[i]\,[j]\,,\ 1 \le j \le f,\ each\ F\,[i]\,[j]\,has\ c * b\ bits$$

The mobile device sends $(F'\,[j]\,||H\,(FN + j))$ and $MAC$to f different cloud servers. The downloading phase starts with device sending $H\,(FN + j)$ to j^{th} cloud server which sends back $F'\,[j]$. After decryption, contents of file are restored and integrity is ensured by matching MAC.
- Sharing-based scheme (ShS): It further decreases computation overhead by providing an energy-efficient scheme using exclusive OR-based secret sharing

mechanism. The mobile device randomly generates $f-1$ files $F'[j]$ *where* $1 \leq j \leq f-1$ and last share f is computed as $F'[f] = \oplus_{i=1}^{f-1} F'[i] \oplus F$. The uploading phase and downloading phase are similar to above two schemes. It is more lightweight than CoS as only XOR operations are involved.

Khan et al. [62] proposed BSS for MCC in which files are logically divided into blocks and lightweight XOR operations are used for achieving confidentiality and extending battery lifetime of mobile node without compromising the security. A mobile user M will choose a secret password consisting of ASCII characters of length $'k'$. Each character of password is encoded into binary string of length 8 bits which is further extended in number of steps say M.

```
$for\ each\ M = 1 to \lceil n/8k \rceil$
{
        // shift password characters by M positions in ASCII table
        $i_j = i_j + M\ (mod 256)$   ∀j = {1, 2, ..., k}
      $p_M = p_M || p_{M-1}$
}
```

Encryption Phase: A separate key Key_j is generated for each block b of file as follows:

$$Key_j = xPWD_{j-1} \oplus C_{j-1}\ where\ C_0 = IV$$
$$xPWD_j = Shift_R\left(xPWD_{j-1}\right)\ where\ xPWD_0 = xPWD\ and\ 1 \leq j \leq b$$
$$C_j = Key_j \oplus B_j\ where\ B_j\ is\ jth\ chunk\ of\ original\ file$$

It also ensures integrity of the uploaded file by using hash-based MAC. The mobile user first calculates integrity key IK by applying hash filenames on filename, extended password and file size. Thereafter, MAC is generated from file contents and IK generated above. Mobile user is finally responsible for uploading file, hash function and generated MAC to the cloud.

$$IK = H\left(FN||xPWD||FS\right)$$
$$MAC = HMAC\left(FILE||IK\right)$$

Decryption phase: When a mobile user wants to download a file from cloud, he will send a request along with $H(FN)$ for unique identification of file to CSP. The mobile user generates an extended password and integrity key again for decryption. The downloaded MAC is matched with newly calculated MAC, and their similarity ensures integrity of file.

3.1.2 Asymmetric or public key cryptography

In 1976, Diffie and Hellman introduced the concept of another cryptography, popularly known as asymmetric or public key cryptography (PKC) [63]. The two different keys are involved, a public key known to everyone used for encrypting and verifying digital signatures and corresponding private key owned by the receiver only used for decrypting and digitally signing the message. PKC can meet the basic security principles mentioned in Section 2. In asymmetric cryptography, if an attacker compromises private key of a user, only messages sent to that user are exposed and messages sent to other party are not exposed as they are encrypted with different key pair.

Public Key Infrastructure (PKI)

PKI is a set of software, hardware, policies, procedures and people that work in collusion to facilitate the secure e-transfer of data over an insecure network. PKI authentication framework consists of public key certificates and digital signatures from a trusted certificate authority (CA) [64,65]. The traditional authentication method based on user ID and password though cost-effective is inadequate for e-commerce transactions, so we need more rigorous proof to confirm the identities involved in communication. In PKI authentication, CA issues a secure digital certificate to the user which binds public keys with user's identity and credentials for a given period of time. In such a way, trust in the user public key relies on one's trust in the CA issuing that digital certificate.

There are several variants of PKI-based cryptosystems such as RSA [66], ElGamal [67] and elliptic curve cryptography (ECC) [68]. Each of these cryptosystem can be used for encryption and digital signatures. There are inherent limitations of MAC or HMAC symmetric key algorithms for verifying authenticity. They fail to provide non-repudiation and involve cumbersome key exchange procedure. Digital signatures employ asymmetric cryptography for providing authenticity of a message. Digital signature binds a sender to digital message using sender's private key which can be verified by the receiver using sender's public key. The signature algorithm takes as input hash value of a message and sender's private key to produce cryptographic value known as digital signature as output. The message along with signatures is sent to receiver end for verification. Diffie and Hellman proposed the concept of digital signatures in 1976 [63]. Thereafter, many digital signature schemes were proposed such as RSA [66], Lamport signatures [69], Merkle signatures [70], digital signature algorithm (DSA) [71], ElGamal signatures [66], elliptic curve DSA (ECDSA) [72].

Lee et al. [73] proposed a two-factor authentication framework for cloud computing based on PKI authentication and mobile out-of-band (OOB) authentication. Two-factor authentication scheme integrates two methods of authentication to enhance the level of assurance that a valid user has been authorized to access a service making it more difficult for unauthorized users to access the service. PKI provides first step of authentication which is a combination of public/private keys, digital certificates and trusted third party. OOB adds another layer of authentication in which user's credential information and one-time random code are transferred to another party through OOB channel and sensitive data on other channel. This solution provides security against phishing and replay attacks preventing fraudulent users to access cloud services. Many online banking services are utilizing OOB authentication for verifying users' identities through a separate channel to prevent real-time threats. User provides PKI certificate and password to prove his/her identity to cloud server (CS). CS sends certificate to corresponding CA for verification. On receiving positive reply from CS, it sends authentication request to authentication server (AS). AS generates one-time authentication code using NLM-128 generator and sends code to the CS, user's mobile phone via a secure OOB channel and workstation. The user accepts the session if the code on phone matches with the session code on workstation. CS also matches the code received from user and AS and grants access if match occurs.

In MCC environment, an adversary can easily hack digital credentials such as password or digital certificates. Xiao and Gong [74] proposed a first lightweight dynamic credential generation (L-DCG) scheme for identifying the mobile users based on randomly generated dynamic credential. Dynamic credentials are updated with mutual coordination between mobile user and cloud when mobile user requested for new communication channel or updates have reached a pre-decided threshold value.

$$DC_{current} = DC_{current} \oplus H(DC_{mobile}||DC_{cloud})$$

DC_{mobile} refers to mobile user's credential which is updated by applying XOR with message sent to cloud by that mobile user and DC_{cloud} refers to cloud's dynamic credential which is updated by applying XOR with message sent to mobile user by cloud. This scheme suffers from certain limitations. It assumes cloud as fully trusted entity, and mobile user has to update dynamic secret frequently resulting in power consumption of mobile device.

Khan et al. [75] enhanced L-DCG [74] and proposed lightweight enhanced dynamic credential generation scheme (L-EDCG) in MCC environment for protecting user's identity. L-DCG treats cloud as fully trusted component and allows mobile users to update dynamic credentials frequently which may result in high processing overhead, communication delay and increased energy utilization on mobile device. The proposed scheme L-EDCG involves 3 main entities, namely mobile user or device, CSP and manager. It performs well in fully distrusted cloud environment and offloads dynamic credential generation operations to fully trusted entity called manager to reduce processing overhead on mobile device. The manager under full control of organization generates public-private key pairs, mobile secret, cloud secret and current credential for each mobile user. It can range from a single system to a private cloud. The current credential is calculated by manager as [76] and the communication between user and cloud occurs through manager. The current credential sharing process when a mobile user wants to access cloud service is as follows:

1. Initially, manager chooses a random nonce values for mobile device and CSP ($Nonce_{mobile} and Nonce_{cloud}$) and encrypts with their respective public keys ($PK_{mobile} and PK_{cloud}$) to verify their authenticity before sending them dynamic credentials.
2. The mobile user and CSP decrypt the received nonce, re-encrypt with the manager public key to verify credibility of manager and forward to the manager.
3. The manager verifies the authenticity of the mobile user and CSP through received nonce, and if verified, follows step 4.
4. Finally, the manager encrypts the current credential information and corresponding nonce with the public key of mobile user and CSP separately to ensure confidentiality. The encrypted information is further encrypted with the private key of manager ($SK_{manager}$) that is used as a signature and delivered to mobile user and

CSP. The mobile user can also verify the credibility of the credential through received signature.

$$DC_{mobile} = Encrypt_{SK_{manager}}(Encrypt_{PK_{mobile}}(Nonce_{mobile}, DC_{current}))$$
$$DC_{cloud} = Encrypt_{SK_{manager}}(Encrypt_{PK_{cloud}}(Nonce_{cloud}, DC_{current}))$$

Khalid et al. [77] proposed a cloud-based secure and privacy enhanced authentication and authorization (PE-AA) protocol that allows anonymous communication. It comprises of 4 different servers, namely strong authentication server for authentication, authorization sever based on XACML policy for authorizing and granting an access token to user for a particular service, identity management server for storing identity information and distributing identity information and keys, local certificate authority server for generating and verifying anonymous certificates. The protocol employed a Web application interface for communication between mobile users and different servers. This protocol is resilient against Web application attacks such as SQL injection, phishing, cross-site scripting, session hijacking, cookie tampering.

Annappaian and Agrawal [78] proposed a customer end multilevel cryptography with metadata and lock algorithm (M-Metadata & Lock) for secure data storage on cloud. The number of levels and ways of encryption are pre-decided by customer based on its data confidentiality requirement and organizational structure. The authors have proposed 3-level and 3-way multilevel cryptography framework in which three levels are chief data officer, cryptography officer and data designers and three ways are data lock, data encryption and metadata, respectively, as shown in Fig. 6.

The responsibilities of three entities/levels involved are as follows:

- Chief data officer: Responsible for locking data using some authentication mechanisms, such as passwords, biometric trait.
- Cryptography officer: Responsible for encrypting data with private and public key cryptographic algorithms.
- Data designer: Responsible for determining and storing metadata, i.e., data pattern or structure of the data to be stored. The sensitive data are converted to metadata before encryption.

Before migrating plain text data to remote cloud, it is pre-processed using jumbled process and then encrypted in chunks. Consider a plain text PT: ABC, DEF to be stored on cloud with jumbled metadata D as even characters appear before odd characters from left to right, i.e., 213. In jumbling process, plaintext gets converted to BAC, EDF. These 2 chunks are then encrypted using algorithms such as AES, RSA or DES. The individual encrypted chunks are grouped together in a folder and locked before sending to cloud for storage. While retrieving the stored data, client has to follow reverse steps, unlocking the folder, decrypting using the same algorithm and converting back to plain text from jumbled data.

In 1985, Miller introduced the use of elliptic curves in public key cryptography [68], but ECC algorithms such as elliptic curve Diffie–Hellman (ECDH) for key exchange, ECDSA were deployed after 2002. Before the introduction of ECC, most of the algorithms using PKC for encryption or digital signatures are based on RSA/DSA which incurs a heavy processing overhead with increasing key length for providing better

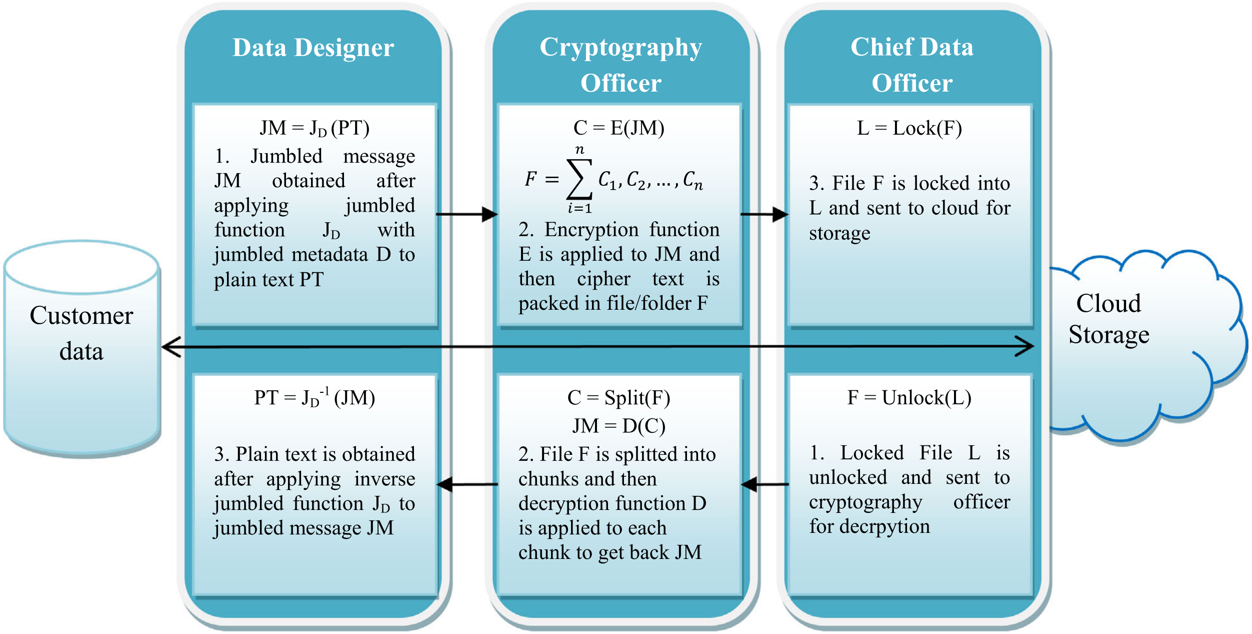
Data Designer
JM = J_D (PT)
1. Jumbled message JM obtained after applying jumbled function J_D with jumbled metadata D to plain text PT
PT = J_D^{-1} (JM)
3. Plain text is obtained after applying inverse jumbled function J_D to jumbled message JM
Cryptography Officer
C = E(JM)
$F = \sum_{i=1}^{n} C_1, C_2, \ldots, C_n$
2. Encryption function E is applied to JM and then cipher text is packed in file/folder F
C = Split(F)
JM = D(C)
2. File F is splitted into chunks and then decryption function D is applied to each chunk to get back JM
Chief Data Officer
L = Lock(F)
3. File F is locked into L and sent to cloud for storage
F = Unlock(L)
1. Locked File L is unlocked and sent to cryptography officer for decrpytion
Customer data
Cloud Storage

Fig. 6 Multilevel cryptography approach

security. This overhead has affected electronic commerce sites to a large extent as they conduct large number of secure transactions. ECC has gained attention these days as it promises to provide equal security with much smaller key length, thereby reducing processing overhead [79].

Toorani and Beheshti proposed LPKI, a lightweight PKI for resource-constrained mobile platforms [80]. LPKI reduces computational overhead and communication cost by employing ECC and signcryption [81,82]. ECC-based approaches are better than modular exponentiation-based systems. Traditional signature-then-encryption scheme digitally signs a message then encrypts the message, whereas signcryption scheme generates digital signatures and encrypts a message in one logical step with one pair of private-public keys assigned to entities. LPKI consists of six components such as registration authority for registering users and issuing them unique identifier, certificate authority for managing and issuing X.509v3 certificates, digital certificates for binding user's identity to his/her public key, certificate repository for storing certificates, Key Generating Server (KGS) for generating public-private keys otherwise end entities are responsible for generating and storing key pairs on their smart cards or SIMs, online certificate status protocol [83] for handling inquiries about certificate revocation status, validation authority for performing validation tasks for end entities using Delegated Path Validation (DPV) protocol [84] and timestamp server for generating valid timing information for other components.

Identity-Based Encryption (IBE)

IBE, proposed by Shamir in 1985, is based on asymmetric cryptography. It allows an individual/sender to compute a public key of a user/receiver by combining public system parameters with unique identity information such as user's email address for encryption, and corresponding private key of receiver is generated by PKG. Boneh and Franklin proposed first practical IBE system in 2001 [85]. It is an application of Weil pairings over elliptic curves and finite fields. Later in 2003, they introduced the concept of bilinear mapping and Weil pairing is an example of such a map [86]. Hierarchical IBE (HIBE) [87] is a generalization of IBE that maps an organization hierarchy. In HIBE, an identity at upper level issues private keys to identities at lower level without compromising the security as it cannot decrypt messages intended for other identities.

Rajni et al. [88] proposed two models, namely mobility node model (MNM) and centralized owner model (COM) based on IBE for privacy and security in mobile cloud environment. In MNM, data owner provides access to mobile client coming from external environment via proxy server that generates key shares without revealing clients' identity to cloud. COM provides a centralized control mechanism which suits well for large networks where same key is used for all mobile clients belonging to same group unlike MNM where a new key is used for all mobile clients increasing computation overhead. In COM, data owner is connected with trusted leader of each group and requests of internal or external mobile client propagates through trusted leader to data owner and cloud. Li et al. [89] proposed a lightweight hierarchical identity-based authentication protocol for cloud computing (IBACC) with corresponding IBE encryption and signatures. In IBACC, communication and computation cost of client is one IBE ciphertext and one IBE signature, while that of server is one IBE decryption and one IBE signature verification.

Dong et al. [90] proposed a secure and scalable data collaboration services (SECO) to allow availability of shared data among multiple users in a consistent manner. It leverages multilevel HIBE for ensuring data confidentiality in an untrusted cloud environment. It allows one-to-many encryption, data writing operation, fine-grained access control, collusion resistance and backward secrecy. The detailed architecture is shown in Fig. 7. SECO consists of 5 different entities: root private key generator (R-PKG) which stands at topmost level possesses a master key and generates private keys and system parameters for low-level PKGs (L-PKG); L-PKGs request private keys from their upper level PKGs and generate corresponding private keys for lower-level PKGs; domain PKG (D-PKG) requests private keys from upper L-PKG and generates private keys for their domain entities or users; users collaborate with each other to share data, receive private keys from D-PKG and interact with cloud server for storing, accessing and updating data dynamically, cloud server for providing storage-as-a-service to all domain entities. Each domain is comprised of D-PKG and multiple users working in collaboration. D-PKG keeps a user list which contains public keys of all authorized users within the domain. Within a domain, a user can encrypt data using multiple recipient public keys and system parameters before storing data in the cloud. The decryption can only be done by intended recipients and corresponding D-PKG using their private keys.

Liu et al. [91] proposed SAKE, a scalable authenticated key exchange for mobile e-health networks (MHN) which refines existing hierarchical identity-based signature and Diffie–Hellman key exchange scheme. In large-scale MHN, there is large number of hierarchies, such as NHIO (National Health Information Organization), RHIO (Regional Health Information Organization) of state, city, street and community and MHN users so there may be long paths between users belonging to different domains. SAKE employs virtual hierarchical network architecture of maximum three levels to reduce heavy communication overhead, mutual authentication costs and authentication path length. The key exchange process consists of 2 phases: Preparation phase involves processing at NHIO and RHIO and generates secret keys for mobile health network units, and handshake phase which is run by individual mobile units generates shared key for communication between individual mobile units.

Attribute-Based Encryption (ABE)

ABE, one of the promising cryptographic techniques, was proposed by Sahai and Waters in 2005 to achieve data confidentiality in a distributed cloud environment [92, 93]. Using ABE, the data owners can enforce fine-grained access policies according to nature of data. In ABE, ciphertext can be decrypted by a particular key if and only if the attribute set satisfies the access policy. ABE has been widely deployed in cloud-based storage scenarios for providing fine-grained access control [94,95]. For example, an employer can upload an encrypted file on the cloud and defines access policy of that file using the following static attributes and functions (NOT, OR, AND): “Manager” OR “Director” AND “Enterprise ABC.” Hence, an employee who is a “Manager” or “Director” employed at “Enterprise ABC” can decrypt the file. It is further classified into two types: key policy ABE (KP-ABE) and ciphertext policy ABE (CP-ABE) [96,97]. In the KP-ABE scheme, private keys of users are associated with an access policy and the ciphertext is associated with a set of attributes [97]. Decryption is

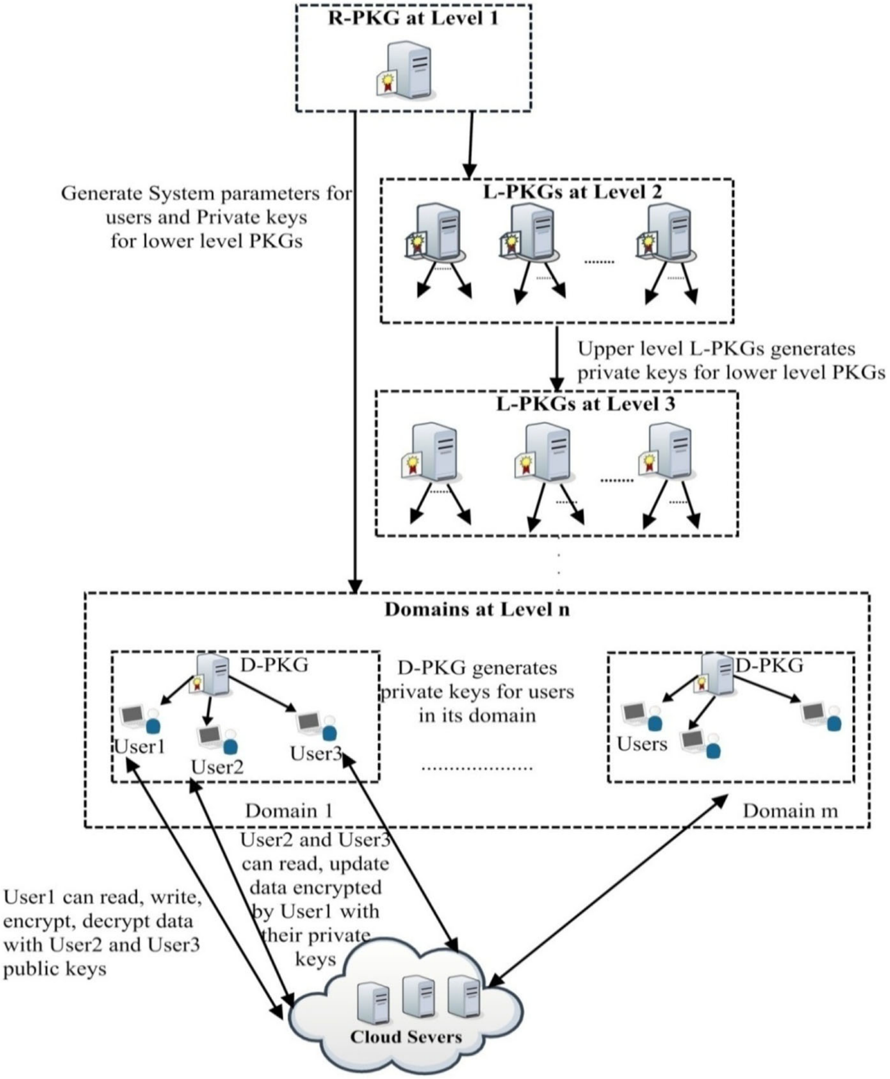
R-PKG at Level 1
Generate System parameters for users and Private keys for lower level PKGs
L-PKGs at Level 2
Upper level L-PKGs generates private keys for lower level PKGs
L-PKGs at Level 3
Domains at Level n
D-PKG
User1
User2
User3
D-PKG generates private keys for users in its domain
D-PKG
Users
Domain 1
Domain m
User2 and User3 can read, update data encrypted by User1 with their private keys
User1 can read, write, encrypt, decrypt data with User2 and User3 public keys
Cloud Severs

Fig. 7 Architecture of SECO

allowed if the access policy defined in the private key matches with the attributes associated with the ciphertext. Yu et al. [98] proposed a fine-grained access control scheme for cloud storage based on KP-ABE. It introduces a minimal computation overhead on the data owner as computation-intensive tasks are delegated to cloud servers. However, this scheme incurs a significant computation cost in key generation and decryption phase. However, in CP-ABE, situation is reversed. The ciphertext is encrypted with access policy, and each user possesses a certain set of attributes which are associated with user’s private key [99]. A user can decrypt the ciphertext if he has corresponding private key. CP-ABE is preferred in MCC environment as it is more close to role-based access control (RBAC) [100] model.

Huang et al. [101] proposed MobiCloud that provides a secure framework with enhanced connectivity and processing power to support MANET functions such as routing, information dissemination and trust management. MobiCloud utilizes cloud computing technology to create a virtualized MANET communication layer by transforming each physical mobile node to a service node on cloud as ESSI (extended semi-shadow images) to address communication and computational deficiencies of mobile nodes. MobiCloud launches a new service, virtual trusted and provisioning domain (VTaPD) for isolating information flow belonging to multiple virtual security domains through programmable routers. Each mobile device is equipped with application manager and sensor manager. Application manager manages software agents (SAs) running on mobile device and cloud platforms, whereas sensor manager manages the sensing data about the device itself such as processor type, battery state, location information and information about neighboring mobile nodes such as neighbor's identity, link quality. Within each VTaPD, there are multiple SAs corresponding to each ESSI which are loaded/unloaded by node manager. The core component for providing security-as-a-service in MobiCloud comprises VTaPD manager and Trust Manager Server (TMS). MobiCloud resource and application manager construct VTaPDs as directed by VTaPD manager and TMS. VTaPD manager collects the context-aware sensed information from mobile device and uses it for risk management and intrusion detection. MobiCloud TMS provides attribute-based key distribution and encryption for secure data access control, identity search and federation services. MobiCloud provides following cloud services for MANET environment.

- Serving as an arbitrator for identity, key, data access management.
- Providing security isolation to information belonging to multiple security domains.
- Monitoring mobile nodes and network against intrusion detection and risk assessment.

Zhou and Huang [102] proposed privacy-preserving cipher policy attribute-based encryption (PP-CP-ABE) and attribute-based data storage (ABDS) for efficient data storage in MCC environment. In PP-CP-ABE, computation-intensive encryption and decryption operations are outsourced to CSPs without compromising the security level as compared to traditional CP-ABE in which operations are performed locally. It consists of data owner (DO), data requester (DR), encryption service provider (ESP), decryption service provider (DSP), storage service provider (SSP) and trusted authority (TA). PP-CP-ABE provides encryption and decryption services to DOs and DRs, respectively, through third party (ESP and DSP) with storage of encrypted data in SSP without revealing the data content and secret keys to them. ABDS allows lightweight and resource-constrained mobile devices to access and manage encrypted data stored on cloud through frequent upload, download and update operations. Yu et al. [103] proposed a security framework for cloud computing based on CP-ABE. But their framework requires users to reveal a part of secret key to the cloud, while PP-CP-ABE forwards blinded private keys to CSPs.

Li et al. [104] proposed modified multiauthority attribute-based encryption (MA-ABE) scheme for resource-constrained mobile devices. They have extended Chase and Chow's scheme [105] for mobile users by introducing a semi-trusted cloud server between the mobile user and attribute authorities for offloading communications and computations from mobile users to this server. The semi-trusted cloud server interacted with the different attribute authorities on behalf of the mobile user and obtained the masked shared decryption keys. It combines all the keys and gets one masked key which can be unmasked by a mobile user only to decrypt the message. The cloud server cannot decrypt the message and determine the attributes of the mobile user, thus preserving the security and privacy of the user.

Lv et al. [106] proposed the first KP-ABE scheme for mobile cloud storage system which outsources encryption and decryption operations to cloud servers. It delegates the tasks of attribute-based access control and updating public keys, ciphertext and private keys to the cloud servers from attribute authority and mobile users without disclosing data contents. This scheme consists of 4 entities: attribute authority (AA), data owners (DO), cloud servers (CS) and mobile users. AA is responsible for generation of public key and master key. Whenever a new user arrives, AA generates a partial component of private key SK_{p1} using master key and transmits to user. DO is responsible for defining attribute set and encrypting data before outsourcing to cloud. CS stores the data and provides access to stored data. It also generates other component of private key SK_{p2}. Cloud severs transmits a partially decrypted ciphertext using SK_{p2} to the mobile user instead of the whole ciphertext when attribute set of user satisfies the access tree. Mobile user further decrypts partially decrypted ciphertext to get plaintext using SK_{p1}.

Fei et al. [107] proposed a lightweight static and dynamic attributes-based access control (LSD-ABAC) scheme for secure data access in mobile computing environment. This scheme includes both static and dynamic attributes such as location of a mobile user, risk level associated in using mobile app currently to enhance security by exploiting features of smart mobile devices. LSD-ABAC consists of 4 phases: setup, key issuing, encryption and decryption. The setup algorithm takes as input security parameters and outputs a bilinear group and a set of public-private key pairs for communication with attribute authorities. The semi-trusted cloud authority communicates with each attribute authority on behalf of user in key issuing phase. The data owner encrypts the file using set of attributes maintained by attribute authorities as well as a set of real-time context attributes obtained from smart mobile device and uploads on cloud. The decryption algorithm takes as input the decryption credentials received from attribute authorities, hash value of context-related attributes, partial decryption key as received from semi-trusted cloud authority, ciphertext from cloud and gives the original data as output.

Dong et al. [108] proposed a privacy-preserving data policy sharing service (PP-DP-SS) based on ciphertext policy attribute and identity-based encryption techniques that ensure collision resistance, data confidentiality and privacy preserving by not disclosing user attributes to the cloud. It consists of data owner, data consumer, cloud server and private key generator (PKG). Data owner refers to an enterprise whose data are stored on cloud, and data consumers are users having an access to stored data as decided by data owner and can decrypt the desired data using secret keys. It is the responsibility of data owner to compute secret key for each user based on all the attributes accessible to that user. PKG computes private keys for users and distributes private keys to users and corresponding public keys (user ID) to data owner. A cloud server provides services to users and consists of authorized user IDs in the cloud. Each file has certain attributes attached to it. The users also possess certain attributes as provided by data owner and user ID. This scheme is an integration of four algorithms namely system initialization that generates parameters for all system entities, encryption phase in which data owner encrypts files and uploads to cloud, key generation in which data owner generates and distributes secret keys to each user through cloud after encrypting it with public key of that user, and decryption phase in which a user can decrypt data in cloud server if and only if he possess matched set of attributes.

Wu et al. [109] proposed secure and cost-effective fine-grained fuzzy access control protocol which allowed small and medium enterprises to offload access control computations to cloud. It relies on attribute-based encryption to avoid communication between cloud and enterprise during authentication stage. The proposed protocol consists of 3 stages: setup, user registration and user authentication stage. During setup phase, enterprise and cloud server agrees upon shared secret key K using Diffie–Hellman key exchange protocol. The enterprise selects a signature key SK and signature verification key PK known to all. The user U with username UN, password pwd and set of attributes $ATTRIB$ chooses a public-private key pair PK_U, SK_U, respectively, during registration stage and sends to an enterprise. Enterprise generates 2 signatures as follows:

$$S_1 = SIGN_{SK}(UN\,||PK_U||\,H(pwd||PSG(K||UN))$$

where PSG is pseudorandom generator and H is cryptographic hash function

$$S_2 = SIGN_{SK}(UN||ATTRIB)$$

During authentication stage, mobile user sends all above parameters to cloud. The cloud server first verifies S_2 and then S_1. Thereafter, cloud server sends a random nonce to user which user signs using his private key and sends back to cloud. After verifying, cloud accepts the user.

Certificateless cryptography

Traditional PKI requires deployment of infrastructure for issuing and managing certificates to guarantee the authenticity of public keys. IBE eliminates need of certificates but suffers from key escrow problem as the key generation server can access the private keys of all users. ABE suffers from both key escrow problem and revocation problem as the private keys of existing users have to be updated whenever a user is revoked. Al-Riyami and Paterson proposed a new cryptosystem called certificateless public key cryptography (CL-PKC) [110] to solve key escrow and key revocation problem. CL-PKC involves key generation center (KGC) for generating partial private keys to users using their identities and master key. KGC securely supplies partial private keys to users and has no access to actual private keys of users. A user combines some secret information with partial private key received to compute its actual private key and KGC's system parameters to compute its public key.

Xu et al. proposed certificateless proxy re-encryption (CL-PRE) scheme for data sharing in cloud [111]. CL-PRE involves data owner, data recipients, cloud resident proxy server and storage server. This scheme lowers computation overhead of data owner by leveraging cloud for encryption and key management. The data owner is responsible for encrypting data using symmetric data encryption key before outsourcing to cloud and generating proxy re-encryption keys with all the recipients. The owner sends encrypted data, access control list and data encryption key encrypted with owner's public key to the cloud. The cloud further transforms his encrypted key to the re-encryption key which can be easily decrypted by recipient's private key.

Seo et al. [112] proposed first mediated certificateless proxy encryption scheme (mCL-PRE) for secure data sharing in public clouds without pairing operations. This scheme involves data owner, user, cloud as storage center, key generation center (KGC) and security mediator. Each user has to generate its own private key and public key pair during registration phase. The user then forwards its public key and identity to the KGC in the cloud which generates two partial keys (SEM-key stored in the cloud security mediator and a U-key given to the user) and public key KGC-key for the user [113]. The public key is combination of the user-generated public key and KGC-generated public key. The data owner encrypts the data using random session key based on access control policies and then encrypts session key using his public key KGC-key. It further sends encrypted data along with access control list to be stored on cloud. Whenever a user requests for data, security mediator verifies the user, retrieves encrypted data from cloud and decrypts the data partially using SEM-key and user decrypts partially decrypted data fully using his private key and U-key. This scheme allows the data owner to encrypt the encryption key once for a data item and provides some additional information to the security mediator so that multiple authorized users can decrypt the data using their private keys.

Tsai proposed an efficient certificateless short signature (CLSS) scheme [114] based on bilinear pairing suitable for devices with low storage communicating in low-bandwidth environment. In CLSS, signature length is smallest and it is impossible for any certificateless scheme to generate signatures smaller than this length. The signature generation requires only one multiplication point operation, and verification requires one pairing and two multiplication point operations. Initially, KGC takes a secure parameter as an input and generates master private–public key pair and public parameters. It further computes partial private keys for each user based on its identity. The user chooses a random secret value and runs an algorithm to calculate its full public key. The user takes as inputs public parameters, its full public key, a message and returns a computed signature. The signature verifier algorithm, on the other side, takes the system parameters, the master public key, its full public key, received signature, message as its input and outputs true when the signature is valid and false otherwise.

3.1.3 Hybrid cryptography

Symmetric cryptographic schemes have lower computational cost and faster encryption/decryption speed but less secure as compared to asymmetric key schemes which are computationally expensive but more secure. The combination of these schemes can provide an efficient security solution. A hybrid cryptosystem including symmetric and asymmetric algorithms provides an efficient security solution without compromising on any of their features. PKINIT is an extension of Kerberos-based authentication [115] to allow public key-based authentication instead of symmetric key authentication between user and key distribution center (KDC) to simplify key management issues in Kerberos. It integrates asymmetric key cryptography into the initial authentication exchange in pre-authentication data fields. The symmetric key cryptography is used afterward for better performance.

Harbitter and Menasce [116] proposed mobile, public key-enabled Kerberos protocol without and with proxy, namely M-PKINIT and MP-PKINIT which are lightweight version of PKINIT suitable for mobile platform. They are combination of public key-based Kerberos and Charon [117], which involves the use of proxy to assist mobile device such as PDA in computation-intensive operations and develops trust relationship between PDA and proxy. It enhances the security of Kerberos by involving minimal public key operations and proxy for load distribution. The less expensive private key operations are performed on mobile device and expensive public key operations on KDC to improve overall performance on mobile platform. It differs from Kerberos since the client generates session key and encrypts it with the private key of KDC if proxy is absent or with the private key of proxy if present and signs it with its own secret key. KDC returns standard Kerberos response to client. It requires significant key operations when a mobile user moves to other Kerberos realm.

Park et al. [118] proposed PKASSO (PKI-based authentication based on single sign-on) protocol, an enhancement over PKINIT protocol to solve resource constraint problem of mobile device by keeping the hardware and software less complex. PKINIT protocol failed to provide digital signature and non-repudiation. It suffers from high authentication latency as complex public and private key operations are required on mobile device. PKASSO offloads the complex PKI operations from mobile device onto a remote server having ample resources. PKASSO employed delegation server and referee server to provide SSO capability. The delegation server offloaded complex PKI operations on the behalf of mobile device and stored proxy certificates containing public and private keys signed by users. The referee server generated binding information between mobile devices and authentication messages which can be later used to provide non-repudiation mechanism. The authentication latency with PKASSO got improved to 0.082 second than conventional PKI-based in which authentication latency is about 5.01 seconds.

Yang et al. [119] modified the public verifiable data possession (PDP) scheme [120] for storage security in cloud computing for resource-constrained mobile devices to ensure privacy, confidentiality, integrity of mobile users' data stored on cloud. It involves TPA (trusted third party auditor) for providing secure services like encryption, decryption, authentication and signature verification to overcome processing overhead on mobile device. The mobile users just have to generate passwords and random numbers. The proposed scheme relies on Diffie–Hellman Key Exchange [121] for distributing symmetric keys over unsecured channel and thereafter, data files, asymmetric keys or secret information can be encrypted using this key and transferred safely. The bilinear mapping [86] and Merkle Hash Tree [70] are used for providing integrity with minimum communication overhead and storage.

Nagaty [122] presented a secure mobile health application based on hybrid cloud platform which integrates cryptographic techniques with role-based access control to protect healthcare data of patients against intruders and authenticate valid users. It allows the mobile clients to access the information stored on cloud securely through token generation in three different access modes such as same hospital access when a doctor from the same hospital who owned a particular patient's medical data at a hospital wants to access it, cross-hospital access when a doctor from some hospital wants to access the medical data of a patient owned by another hospital, and third is emergency mode when a patient entered an emergency center and the doctor wants to access the patient's medical data owned by a hospital. IBE is used to generate patient's public and private key pairs. The patient's medical data and corresponding ACL containing patient id, access rights and patient's secret key are double-encrypted firstly by using patient public key and then using hospital's public key. The signatures of patient's medical data file and access control list are generated using SHA1 or SHA2 and sent along with the files from hospital's private cloud server to public cloud server over secure transmission channel for storage. Sujithra et al. [123] proposed a hybrid cryptographic approach for mobile device data security by outsourcing mobile data to remote cloud with minimal performance degradation. In this paper, three-tier hybrid approach has been designed. The first tier includes data encryption using MD5 algorithm, second tier encrypts the data again using AES algorithm, and third tier is twofold; either it re-encrypts data using ECC or key using RSA algorithm. The comparative analysis of individual cryptographic algorithms such as DES, AES, ECC, MD5 [56] is done with hybrid approach, i.e., asymmetric with digital signature or symmetric algorithm on both local environment and remote cloud environment.

3.1.4 Proxy re-encryption

It is a cryptographic primitive which enables the re-encryption of ciphertext from one secret key to other without knowing the actual secret keys. It is used when one user A wants to reveal the contents of message M encrypted with his public key PK_A to another user B without revealing his private key SK_A to user B. A weaker form of proxy re-encryption is that proxy server possesses keys of both users': one for decrypting and other for encrypting. This was the only solution available till 1997. But user A wants that B should be able to read the contents without revealing message contents and keys to trusted third party say proxy. Later, researchers suggested more efficient approaches for proxy re-encryption.

In 1998, Blaze et al. [124] proposed BBS, an atomic re-encryption approach based on ElGamal cryptosystem [66] and re-encryption key $RK_{A \rightarrow B}$. Using $RK_{A \rightarrow B}$, a proxy can re-encrypt ciphertext without knowing the plain text. In order to compute $RK_{A \rightarrow B}$, one party must share his/her secret key with other party or rely on trusted entity in BBS scheme. Green and Ateniese [125] proposed first unidirectional and collision-resistant re-encryption scheme which avoids pre-sharing of any secret between communication parties. It is based on bilinear mapping. Therefore, proxy re-encryption is very useful in MCC environment as computationally intensive data access operations can be offloaded from resource-constrained mobile devices to trusted third party such as proxy, manager or cloud.

Jai et al. [126] proposed secure data service mechanism (SDSM) which ensures data privacy as well as fine-grained access control with minimum communication overhead based on IBE and proxy re-encryption. It allows mobile users to encrypt the data with his identity to protect from data leakage and simultaneously outsourcing data and security management to cloud in a trusted mode without disclosing any information. It involves three entities data owner as mobile device, data sharer and CSP rendering cloud services to data owner. Data owner is responsible for sharing data or files and granting various access privileges to data sharer. Data sharer and data owner communicate with the CSPs for data storage and retrieval. The cloud can grant access to an authorized user by re-encrypting the text with data sharer's identity which was earlier encrypted by data owner's identity. Data owner needs to forward re-encryption key with size much smaller than actual file to cloud and rest of the re-encryption tasks are done on cloud. SDSM is flexible enough to scale up as the number of data sharers increase.

Tysowski and Hasan [127] proposed manager-based re-encryption (MReS) and an efficient and scalable cloud-based re-encryption model to address the demands of mobile users. In MReS, all read requests initiated by mobile users are normally serviced through the manager which limits the scalability of the system. The manager under an organizational control maintains an access control list of authorized users and generates public and private keys for each user in the system and each partition on the cloud. During download request from user, manager has to generate re-encryption key and transformed ciphertext into re-encrypted ciphertext to be forwarded to the user. In CReS, the overall key management is done by client for security purpose, but computation-intensive data re-encryption is done by CSPs and key redistribution is kept minimal to reduce communication cost on mobile devices. All the members of group receive private key of partition initially through manager and are able to decrypt message encrypted by any member of the group. When a new user enters the group and wants to access data, current partition key becomes invalidated and new version of the partition key is generated by applying hash function to a combination of current key and random salt. The manager then generates re-encryption key and sends it to cloud provider to re-encrypt the data.

Khan et al. [128] proposed CMReS an enhancement over MReS and CReS. In CMReS, the data owner can directly download and decrypt the message without the involvement of any trusted entity unlike manager in MReS. The mobile users are responsible for generating shared key pair and disseminating the public key information. The re-encryption responsibilities are offloaded to cloud without disclosing any private keys in contrast to MReS and CReS. In CReS, the whole system gets affected if a particular group member is compromised because data partitions are encrypted by partition public key unlike in CMReS where message is encrypted by user's private key. It consists of 4 modules: mobile users, fully trusted encryption/decryption service provider, re-encryption service provider hosted on public cloud and cloud for storage. Re-encryption service provider module is responsible for maintaining the re-encryption keys and performing re-encryption tasks for authorized mobile users without knowing private keys of mobile users.

4 Incremental cryptography

Incremental cryptography involves designing of cryptographic algorithms in such a way that having applied algorithm once to a document, the result (say hash value)

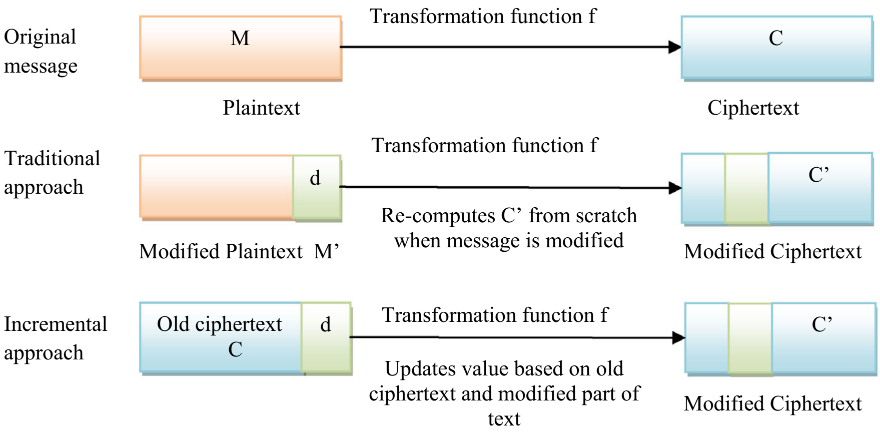
Original message
M
Transformation function f
C
Plaintext
Ciphertext
Traditional approach
d
Transformation function f
C'
Re-computes C' from scratch when message is modified
Modified Plaintext M'
Modified Ciphertext
Incremental approach
Old ciphertext C
d
Transformation function f
C'
Updates value based on old ciphertext and modified part of text
Modified Ciphertext

Fig. 8 Incremental cryptography

can be updated rapidly for a modified document without re-computing the result from scratch as shown in Fig. 8. Consider a modification in small part *d* of the original document *M* resulting in modified plaintext M'. In traditional approach, transformation function is proportional to size of message M', whereas in incremental cryptography, transformation function is proportional to size of modified part *d*. The concept of incremental cryptography was given by Bellare et al. in 1994 [129] and later modified in 1995 using pseudorandom functions [130] and in 1997 using collision-free hash functions such as modular multiplication (MuHASH) and addition (AdHASH) [131]. So, it turns out to be a useful tool when the contents of a file changes too frequently.

Itani et al. [132] proposed first incremental cryptographic-based energy-efficient protocol for mobile cloud computing environment that provides integrity of mobile users' data stored on the cloud based on concept of proxy re-encryption and trusted computing. The trusted third party is responsible for installing coprocessors on cloud which distribute secret keys and generate message authentication code for clients. The protocol operation consists of 3 main phases:

- Initialization phase: In this phase, MAC is generated for every file to be uploaded on cloud using shared key. The files are stored on cloud and corresponding MAC on mobile device.
- Data update phase: In this phase, secure dynamic operations on cloud data such as block insertion and deletion are efficiently supported. The mobile user first requests CSP for block insertion or deletion in a file. CSP sends copy of requested file to both mobile client and trusted crypto-coprocessor. The coprocessor generates MAC using shared key and sends this MAC to mobile client. The mobile client re-calculates MAC from file received and verifies the integrity of the file by comparing calculated MAC value with received MAC value. The mobile client intended to insert a block at some position in file updates MAC value by using incremental cryptography and sends only inserted blocks to remote cloud. Similarly, while deleting a block, mobile client updates MAC value depending on deleted block and old MAC value.

- Data verification phase: The processing overhead involved in integrity verification is offloaded from mobile device to trusted crypto-coprocessor. The coprocessor receives selected files from remote cloud for integrity verification. It generates incremental MACs and sends them to mobile client for matching.

Khan et al. [76] proposed an incremental version of popular security schemes such as EnS, CoS and ShS for improving the block modification operations (insertion, deletion or updation) on resource-constrained mobile devices. The original file is divided into *b* blocks of equal size. The mobile user provides a password to be transformed into encryption and integrity keys as EnS [61] to provide confidentiality. DES and SHA-1 are used in EnS, and coding vector matrix is multiplied with blocks in CoS and XOR-based secret sharing for ShS for encryption and decryption tasks. The final encrypted file is generated by concatenation of individually encoded blocks and stored on cloud. Similarly, final MAC of file is generated by applying cryptographic hash function to the concatenation of individual blocks' MAC values.

$$\begin{aligned}
Encrypted\ block\ B_i' &= Encrypt_{EK}\ (FILE_i)\ where\ 1 \le i \le b \\
Encrypted\ file\ F' &= B_1'\ ||B_2'|\ |B_3'|\ |\ldots|\ |\ B_b' \\
MAC_{B_i} &= HMAC_{IK}\ (FILE_i)\ where\ 1 \le i \le b \\
Final\ MAC\ &= HMAC_{IK}(MAC_{B_1}\ ||MAC_{B_2}|\ |MAC_{B_3}|\ |\ldots|\ |\ MAC_{B_b}
\end{aligned}$$

The mobile user uploads encrypted file, H(FileName), individual blocks' MAC and final MAC. It stores only file name and corresponding number of blocks in that file in its local memory. The proposed scheme is efficient for block modification operation. The mobile user encrypts only modified block not the entire file and generates corresponding MAC. Thereafter, the mobile user has to specify block number for modification and sends encrypted block and MAC to CSP. CSP re-calculates the final MAC of file based on the updation received from mobile user.

Khan et al. [133] proposed incremental proxy re-encryption (I-PReS) as an extension of BSS which uses the concept of incremental cryptography for improving the file modification operations without compromising the confidentiality and integrity services. The information about the file and the number of blocks into which a file is divided is stored locally on mobile device. MAC for each block is calculated by applying hash function, and these codes are concatenated to calculate final hash code of the file. I-PReS shows significant improvement in terms of CPU processing overhead, energy consumption, turnaround time and memory storage of mobile device. The scheme consists of 6 phases:

Setup phase: I-PReS works on bilinear mapping $e : G_1 X G_1 \rightarrow G_2$ of groups of prime order q. The system parameters g ε G_1 and Z = e(g, g) ∈ G_2 are randomly generated.

Key generation phase: Proxy server is assumed to be a trusted entity which generates public and private key pair for authorized user *i* of data partition on cloud and securely disseminates keys.

$$\begin{aligned}
\text{Shared key for user } i(SK_i) &= x_i \\
\text{Public key for user } i\ (PK_i) &= g^{x_i} \quad \text{where } x_i \in Z_q^*
\end{aligned}$$

Encryption phase: Consider a mobile user A who wishes to upload file F on cloud. The mobile user splits the file into b blocks of size s except last part.

$$F = \bigcup_{j=1}^{b} B_j \textit{ where } B_j \textit{ is } j^{th} \textit{ block of file } F$$

Mobile user A chooses a random number $r \in Z_q^*$ and encrypts each block of file using his private key as follows:

$$\begin{aligned} C_A &= \mathrm{g}^{\mathrm{x_A.r}} \\ C_j &= \mathrm{B_j}.Z^r \text{ where } 1 \le j \le b \\ C &= \cup_{j=1}^{b} C_j \end{aligned}$$

Similarly, MAC for each block of file is calculated using hash function and concatenated to get final MAC for verifying integrity of file.

$$MAC = H_{SHA}\left(\bigcup_{j=1}^{b} H_{SHA}\left(B_j\right)\right)$$

The encrypted file of user (C_A, C), MAC for each block, number of blocks and final MAC are uploaded on cloud. Mobile user retains the filename and number of blocks for each file on the device.

Re-encryption phase: The mobile user B who wishes to view the file uploaded by mobile user A requests the proxy server for re-encryption. The proxy server downloads and re-encrypts the file C_A after verifying access rights of mobile user B as follows:

$$\begin{aligned} C_B &= e\left(\mathrm{g}^{\mathrm{x_A.r}}, g^{\frac{x_B}{x_A}}\right) = e\left(g, g\right)^{r.x_B} = Z^{r.x_B} \\ C_j &= \mathrm{B_j}.\, \mathrm{Z}^r \text{ where } 1 \le j \le b \end{aligned}$$

Proxy server sends (C_B, C), final MAC and number of blocks to mobile user B.

Decryption phase: Mobile user B decrypts the file contents using C_B and his private key.

$$\begin{aligned} \left(Z^{r.x_B}\right)^{\frac{1}{x_B}} &= Z^r \\ B_j &= \frac{B_j Z^r}{Z^r} \\ F &= \bigcup_{j=1}^{b} B_j \textit{ where } B_j \textit{ is jth block of file } F \end{aligned}$$

Integrity verification phase: Mobile user B compares calculated MAC with downloaded MAC. If these values match, integrity is ensured.

Block updation phase: If mobile user A wants to insert new blocks in the file uploaded on cloud earlier, he has to send update request to CSP, MAC of each new block to be inserted, final MAC, location information where new blocks are to be inserted and new encrypted blocks. Similarly, I-PReS shows improvement in results in comparison with PReS while performing block deletion and modification operations in terms of turnaround time and energy consumption.

Figure 9 summarizes the cryptographic preliminaries and building blocks associated with each data security scheme in MCC environment, proposed year and interdependency between the schemes. Table 3 summarizes comparative analysis of above-mentioned cryptographic-based techniques in MCC environment.

4.1 Biometric-based security schemes in MCC

The recent advances in cloud computing offer deployment opportunities by executing computationally intensive biometric matching tasks and providing accessible entry point for various applications on mobile devices. In March 2015, Stanford Research Institute International has licensed its iris on move technology to Samsung, and it has developed Samsung Galaxy Tab Pro 8.4 Tablet to provide biometric identity management solution to its users. It is 1000 times more accurate than fingerprint-based solutions. Besides, other smartphone manufacturers such as Apple, HTC, Nokia, Sony are incorporating fingerprint, facial and voice recognition to their mobile devices. Since the number of devices incorporating biometric technology are increasing, the volume, variety and heterogeneity of biological data are increasing exponentially. Large-scale biometric search requires highly scalable, low-latency and reliable cloud computing framework for sufficient storage and parallel processing capabilities for identifying an individual. The integration of biometric and cloud computing with mobile devices allow mobile users to take data out of their pockets without fear of data loss and access data anywhere and at any time.

The biometric recognition process comprises of four main steps: capturing sample, feature extraction, template comparison and matching. At enrollment phase, a person's biometrics is captured by the scanner or sensor. The next block image pre-processor performs necessary pre-processing, i.e., removing artifacts from noise, enhancing input, removing background noise and normalizing. The software part consisting of feature extractor, template generator and matcher converts the biometric input into a digital template and identifies specific points of data as "match points" and stores it in a database. A vector of input image intensities is used to create a template by extracting relevant characteristics from the image. In subsequent phase, biometric information is detected and compared with the information already stored in the form of template during enrollment. The information storage and retrieval tasks should be secure to make entire biometric system robust. The match points are further converted into a value using any of the matching algorithms and compared with biometric data in the database. The biometric templates are generally stored on smart card and/or within a database. In matching phase, the obtained template is passed to a matcher that compares it with other stored templates, estimating the hamming distance between them. The output obtained can be used for any designated purpose.

The attacks in a biometric system can be categorized as attacks at user interface level such as spoofing, attacks at interface between modules such as replay or brute force attack and on template database such as modifying stored templates. There are

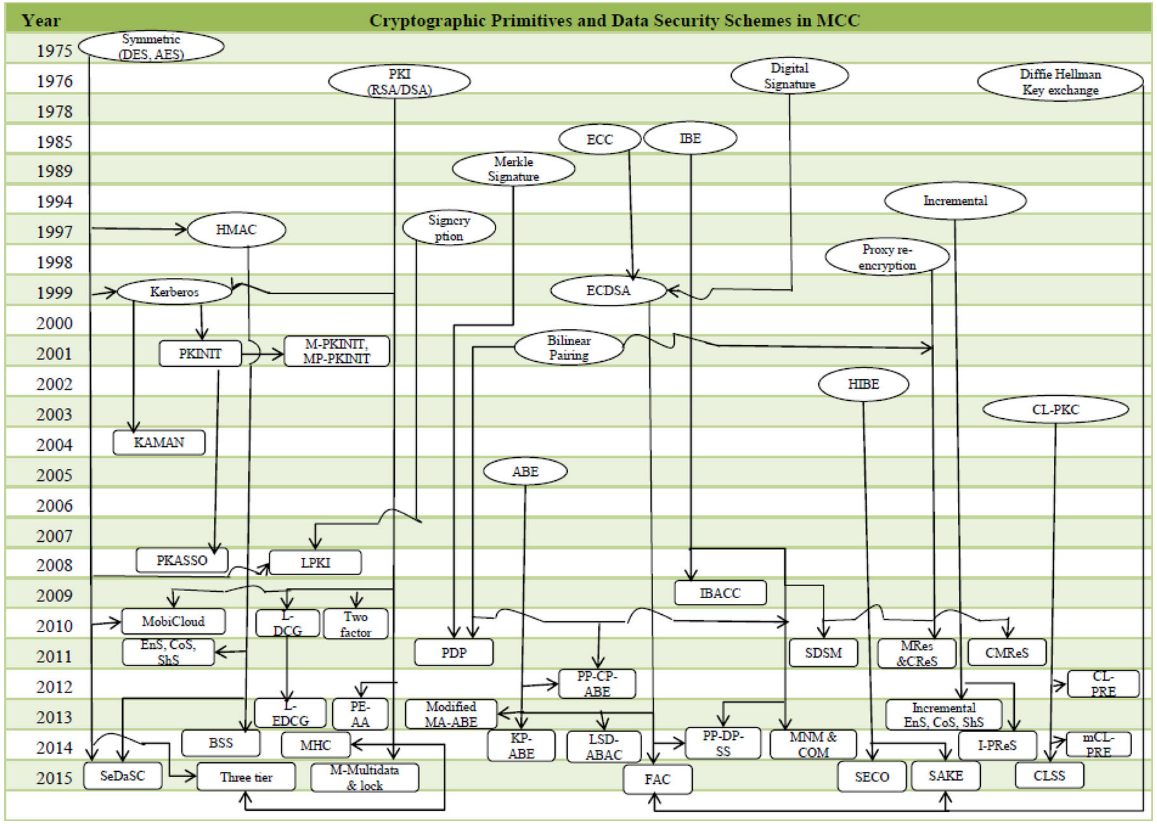

Fig. 9 Cryptography preliminaries and data security schemes in MCC ⬭. Represents cryptographic building blocks and ▭ represents data security protocols in MCC environment

Table 3 Comparison of cryptographic-based techniques in MCC environment

Scheme	Security principles	Technique(s) used	Tools used	Performance metrics	Merits	Demerits
KAMAN [58], 2004	Authentication	Kerberos, Symmetric key cryptography	Theoretical analysis	—	Kerberos server is no longer a single point of failure and suitable for mobile environment. It is immune against fabrication, battery exhaustion, impersonation and modification attacks. It employs availability check feature to minimize the risk of insider attack	It can't provide Internet-wide interoperability and authentication as symmetric cryptography requires pre shared secret. It doesn't support signature and non-repudiation mechanism which is basic requirement of many security applications
SeDaSC [60], 2015	Confidentiality, integrity and access control	Symmetric key cryptography, AES, SHA-256	Visual Studio C# using .Net4 framework, Amazon S3 as cloud server, Intel i3 CPU, RAM 4GB for client machines and CS	Time taken in key generation phase during group creation, the turnaround time for encryption and decryption, the file upload and download time	It does not use computation-intensive ECC or bilinear pairing operations. It provides security against insider attacks, forward and backward access control. Encryption and decryption tasks are carried by CS on cloud so suitable for mobile environment	It assumes that cryptographic server residing on public cloud as fully trusted entity. The data owner transfers unencrypted file to CS so it is vulnerable against data-in-transit issues
EnS, CoS, ShS [61], 2011	Confidentiality and integrity	Message digest, coding-based scheme, XOR-based secret sharing method	Theoretical analysis	Computation overhead	Lightweight in terms of computational overhead, resilient to storage compromise on mobile devices, and do not assume that trusted cloud servers are present	Extra file management overhead in CoS and involvement of lightweight operations in ShS may result in compromised data privacy

Table 3 continued

Scheme	Security principles	Technique(s) used	Tools used	Performance metrics	Merits	Demerits
BSS [62], 2014	Confidentiality and integrity	Standard cryptographic function, hash-based MAC such as SHA- 1 or MD-5	BlackBerry JDE 7.0.0, Google App Engine (GAE), AppEngineFile API	Energy consumption, CPU utilization, memory utilization, encryption time, decryption time and turnaround time	It consumes less energy, reduces the resources utilization, improves response time,and provides better security services to the mobile users in the presence of fully untrusted cloud server(s)	It does not provide non-repudiation service. It does not support block insertion, deletion, and modification operations
Two-factor authentication [73], 2010	Authentication	PKI and OOB channel	Theoretical analysis	—	One-time random code is valid for a single session, thus protecting users from phishing and replay attacks. It provides forward and backward secrecy	A single key distribution center can become a single point of failure and difficult to maintain with large number of users in a cloud environment
L-EDCG [75], 2013	Confidentiality and authentication	Hash function, asymmetric cryptography	Black Berry JDE 7.0.0 and Android SDK, JDK 1.6, GAE	Turnaround time and energy consumption with varying threshold and number of packets	Automatic dynamic credentials are generated for the identity verification of mobile users in cloud environment with reduced processing overhead, communication delay and energy dissipation on the mobile device. It can withstand MITM and impersonation attack	Limited scalability due to workload on trusted entity. If manager gets compromised, the privacy of whole system is affected

Table 3 continued

Scheme	Security principles	Technique(s) used	Tools used	Performance metrics	Merits	Demerits
PE-AA [77], 2013	Authentication and authorization	A strong authentication server, an XACML-based policy server for authorization, IDMS with a key server and local CA	Theoretical analysis	—	Flexible framework providing full anonymity and prevents identity theft by employing anonymous identities	No formal proof or analysis. Integrity of the message can be compromised
LPKI [80], 2008	Confidentiality, authentication and integrity	ECC for digital signature, symmetric cryptography for encryption, HMQV for key exchange	Theoretical analysis	—	It is lightweight so suitable for the resource-constrained mobile devices. It is compatible with the popular PKIX infrastructure	Formal and experimental analysis is not there. If CA gets compromised, the attacker can issue false certificates
MNM and COM [88]	Confidentiality	IBE, Bilinear pairing	Theoretical analysis	Encryption and decryption time	Both models enable the data owner to maintain the confidentiality and privacy of data. In COM model, same key is used for mobile clients belonging to same group so it is suitable for large environment	It suffers from key escrow problem as re-encryption key is generated every time in MNM

Table 3 continued

Scheme	Security principles	Technique(s) used	Tools used	Performance metrics	Merits	Demerits
IBACC [89], 2009	Authentication	Identity-based hierarchical model	GridSim, SimJava, P4 3.0 CPU, 2G memory, openssl0.9. for SAP	Authentication time, communication cost, computation time of client and server	The communication cost of IBACC is less and the authentication time is shorter than SSL Authentication Protocol (SAP). Highly scalable as computation time for client is 20% to that of server so more lightweight devices can connect to servers	It did not consider multiuser data sharing and focuses on data exchange between two individual parties. IBE generally allows users to generate encryption keys themselves but this scheme requires trusted cloud provider for generating encryption keys so communication cost involved is still high
SECO [90], 2015	Confidentiality and fine-grained access control	Multilevel hierarchical identity-based encryption (HIBE)	Python	Computation complexity, communication cost, user revocation cost and storage cost	SECO is highly efficient and has low overhead on computation, communication and storage. It is resilient against both internal intruders and unauthorized external users. It is collusion resistant and provides backward secrecy	Privacy issues and data synchronization problem was not considered. It suffers from key escrow problem. The scheme relies on PKGs and did not consider the case when it fails or gets compromised

Table 3 continued

Scheme	Security principles	Technique(s) used	Tools used	Performance metrics	Merits	Demerits
SAKE [91], 2015	Key management	Hierarchical identity-based signature and the Diffie–Hellman key exchange protocol	Java pairing-based cryptography library	Time consumed in different phases w.r.t hierarchy depth	Computational cost is independent of the users' hierarchy levels in the real MHN. It is secure against active attackers in MHNs	It does not provide collaborative data sharing and fine-grained access control
MobiCloud [101], 2010	Confidentiality and authentication	Attribute-based identity management, key management, VTaPD	Theoretical analysis	—	It provides a fundamental trust model including identity management, key management and security data access policy enforcement	Privacy of user's data are compromised with the loss of mobile device. It assumes cloud as trusted node. MobiCloud service delay needs to be further investigated
PP-CP-ABE and ABDS [102], 2012	Confidentiality and access control	Bilinear mapping, CP-ABE	Mote sensor (8 bit-7.37 MHZ ATMega128L, 4KB RAM), pocket PC (600 MHZ CPU) and PC (1GHZ CPU)	Computation overhead	The computation overhead is linear for ESP and DSP and constant for the mobile user with increasing access policy size. PP-CP-ABE is secure even if service providers and malicious users collude	It incurs significant computation overhead as a number of exponentiations are required to generate the blinded private key. It is secure under the random oracle model, which is less secure than the standard model. Data integrity is not provided

Table 3 continued

Scheme	Security principles	Technique(s) used	Tools used	Performance metrics	Merits	Demerits
Modified MA-ABE [104], 2013	Confidentiality and authentication	CP-ABE	Theoretical analysis	Computation cost and communication cost	Number of communications required is constant with increasing number of attribute authorities. Semi-trusted cloud authority can't decrypt data	It is based on static attributes only. The ciphertext length increases linearly with the increase in ciphertext attributes involving evaluation of more pairing and exponentiation operations while decrypting ciphertext
KP-ABE for MCC [106], 2014	Confidentiality and access control	KP-ABE	Pairing-based cryptography library, Linux OS with an Intel Xeon 3.2 GHz and 2GB RAM	Computation cost, communication cost and storage overhead	Immune to collision attack. The key generation and decryption overhead at attribute authority site are independent of the size of access policy	The total key generation overhead at attribute authority and cloud server is more than traditional KP-ABE schemes. Single point of failure as centralized attribute authority is involved

Table 3 continued

Scheme	Security principles	Technique(s) used	Tools used	Performance metrics	Merits	Demerits
LSD-ABAC [107], 2014	Confidentiality and authentication	CP-ABE	Java pairing-based cryptography library	Computation cost and communication cost with varying number of attribute authorities	Lightweight as it reduces the computational complexity. It provides run-time security of mobile data stored on cloud with dynamic attributes making it more secure than [104] and is resistant to collusion attack	It is based on centralized ABE scheme thus privacy can be compromised
PP-DP-SS [108], 2014	Confidentiality and fine-grained access control	CP-ABE and IBE	Python	Computation overhead, communication overhead, revocation cost and size of ciphertext	It ensures fine-grained data access control, backward secrecy and security against collusion of users with the cloud and supports user addition, revocation and attribute modifications	It relies on one-to-one encryption approach, i.e., encrypted data can be decrypted by a particular recipient only instead of one-to-many approach. It fails to provide data collaboration services in which multiple users can read/write encrypted data collaboratively

Table 3 continued

Scheme	Security principles	Technique(s) used	Tools used	Performance metrics	Merits	Demerits
FAC in MCC [109], 2015	Authentication and access Control	Diffie–Hellman key exchange, ECC-based signature and ABE	jPBC, a mobile device (HTC Desire HD A9191, Android 2.2)	Execution time and exponentiation operations during registration and authentication stage	It can withstand dictionary and phishing attack. It provides enhanced privacy as user database is not provided to cloud. It also provides user traceability	The privacy of data is not considered. Performance evaluation is done on the basis of execution time only
CL-PRE [111], 2012	Confidentiality	Certificateless cryptography	One-core,1GB memory Linux virtual machine, Intel i5 3.4GHz processor, Elliptic curve defined on 512 bits prime field with a generator of order 160 bits	The proxy re-encryption time is about 7–8 ms with 3k bits of both re-encryption key size and ciphertext size	It can solve key escrow and certificate revocation problem. It enhances security by utilizing multiple proxies in different cloud providers	It involves expensive bilinear pairing operations as compared to standard operations in finite fields. It is insecure against partial decryption attack. It only achieves CPA (Chosen Plaintext Attack) security which is insufficient to protect many real world applications

Table 3 continued

Scheme	Security principles	Technique(s) used	Tools used	Performance metrics	Merits	Demerits
mCL-PRE [112], 2014	Confidentiality and integrity	Certificateless cryptography, SHA1	32 bits GNU Linux kernel version 3.2.0-30 with an Intel Core CPU 2.40GHZ, V. Shoup's NTL library	Encryption and decryption time for different message sizes and different number of users accessing a same data item	It can solve key escrow and certificate revocation problem. It doesn't solve bilinear pairing so efficient for mobile devices. It is secure against partial decryption attack. It achieves both CPA (Chosen Plaintext Attack) and CCA (Chosen Ciphertext Attack) security	The key generation process is shifted to the shared multitenant public cloud environment which is not recommended from security point of view. It assumed public cloud as trusted entity. The decryption time increases as it is performed twice in the system
CLSS [114], 2015	Integrity	Certificateless cryptography, bilinear pairing, ECC	HTC Desire HD 1 GHz, single-core 1 GHz CPU and 768 MB RAM, jPBC library	Execution time required in ECC multiplication operation and pairing operation used for signature generation and verification	It is suitable for low-bandwidth mobile devices with low storage. It is secure against super type I and super type II adversaries under random oracle	It involves expensive pairing operations so computational cost is high

Table 3 continued

Scheme	Security principles	Technique(s) used	Tools used	Performance metrics	Merits	Demerits
M-PKINIT and MP-PKINIT [116], 2001	Authentication	PKINIT, DES for secret key encryption, RSA with 1024-bit keys for public key encryption	Vadem Clio C-1000, Windows CE, 100 MHz MIPS R4000 CPU, 16 MB RAM, Pentium Proxy and KDC	Execution time in different phases such as pre-authentication, authentication, post-authentication	M-PKINIT and MP-PKINIT reduces authentication overhead on mobile device when public key operations are performed. MP-PKINIT provides the option to mobile client to preserve the encrypted data through the proxy	It integrates asymmetric cryptography which increases overall authentication time. High communication overhead. It also requires simultaneous access to three servers for initial authentication which is difficult for hostile mobile environment. It doesn't support non-repudiation
PKASSO [118], 2008	Authentication and non-repudiation	PKINIT protocol, SSO protocol, a delegation server and a referee server	Pseudoservice device and a PANDA which are coupled to a load generator	Operation time at client and server side, authentication latency with varying numbers of requests per second	The authentication latency (0.082 second) is shorter than the authentication latency of conventional PKI-based authentication latency (5.01 seconds). PKASSO is safe from replay and MITM attacks	The scheme lacks security consideration for attacks such as cryptanalysis and message slicing. CA can become a bottleneck for the system

Table 3 continued

Scheme	Security principles	Technique(s) used	Tools used	Performance metrics	Merits	Demerits
Modified PDP for mobile devices [119], 2011	Confidentiality, authentication and integrity	Diffie–Hellman key exchange, Bilinear signature and Merkle hash tree (MHT)	Theoretical analysis	Computation and space overhead	Less storage space and computing ability required by end user during verifying the data possession	The involvement of third party called trusted entity may degrade the performance of the system with an increase in number of the mobile users. A prototype system must be build to test the actual performance
Mobile Health Care on a Secured Hybrid Cloud [122], 2014	Confidentiality, integrity, non-repudiation, authentication and access control	Role-based access method, IBE, crypto-processors	Theoretical analysis	—	This system can alert patients against critical situations and can give real-time advice to them about treatments. It avoids key collisions and improves data privacy and confidentiality by using symmetric and asymmetric cryptography	It is computationally intensive

Table 3 continued

Scheme	Security principles	Technique(s) used	Tools used	Performance metrics	Merits	Demerits
Three-tier security [123], 2015	Confidentiality and integrity	MD5, AES, ECC, RSA	—	Mean processing time, speedup ratio	Mean processing time decreases with cloud environment than local environment with AES and combination of MD5, ECC, AES algorithm qualifies better than others in speedup ratio	It does not provide authentication. It is vulnerable to impersonation attack
SDSM [126], 2011	Confidentiality and fine-grained access control	IBE, Proxy re-encryption, ECC	MIRACL cryptographic library, Intel Pentium Dual-Core, Windows OS	Computation and communication cost in encrypting data and generating re-encryption key	The communication overhead involved in data sharing is proportional to the size of a re-encryption key cost at user site and can be reduced if cloud charges on the basis of communications. It can withstand collusion attack	Before uploading file, mobile users have to perform intensive cryptographic (pairing and exponential) operations which consumes considerable energy. The outsourcing of re-encryption and security management tasks to CSP may cost more to user

Table 3 continued

Scheme	Security principles	Technique(s) used	Tools used	Performance metrics	Merits	Demerits
MReS and CReS [127], 2011	Confidentiality and authentication	Asymmetric cryptography, Bilinear mapping, Re-encryption	jPBC, Apple iMac with quad-core 64-bit 3.4 GHz i7 processor, Google Nexus phone with a single-core 1 GHz ARM processor, GAE Web application instance	Encryption time, decryption time before and after re-encryption, re-encryption time	Efficient and scalable for mobile environment as re-encryption tasks are performed inside the cloud. Cloud provider has insufficient access to key values to decrypt the user data	Manager is allocated all re-encryption tasks in MReS so scalability is an issue. The whole system gets compromised if a group member is compromised. The excessive use of the cloud resources involved in re-encryption overcharged to an organization if membership changes are too frequent
CMReS [128], 2015	Confidentiality and authentication	Asymmetric cryptography, bilinear mapping, Re-encryption	Mobile client: Sony Xperia S, Android SDK, Dual-core 1.5 GHz, 1 GB RAM Trusted entity: JDK 1.6, Inter(R) Core i5-2400 CPU, Windows 7, 4 GB RAM	Turnaround time, energy and memory consumption and CPU utilization	Compromise of the group member does not affect the privacy of the entire system. The mobile user can offload encryption, decryption and re-encryption operations on manager and cloud to save battery of the mobile device	Trusted entity is burdened more as compared to CReS due to additional responsibility of the encryption and decryption operations. The presence of trusted entity may affect the scalability of whole system

Table 3 continued

Scheme	Security principles	Technique(s) used	Tools used	Performance metrics	Merits	Demerits
Energy-Efficient Incremental Integrity for Securing Storage [132], 2010	Integrity	Incremental cryptography and trusted crypto-coprocessor	Emulated cloud computing Unit, HP iPAQ Pocket PC device	Execution time and energy saving	The average execution time of the incremental MAC generation on the mobile device was 37 ms compared to 420 ms of the traditional integrity approach	The data privacy is overlooked. Computational overhead is high. Performance degradation with increasing number of mobile users interacting with trusted entity
Incremental EnS, CoS, ShS [76], 2013	Confidentiality and integrity	Incremental cryptography, DES, SHA-1, XOR-based secret sharing	Blackberry JDE 7.0.0	Turnaround time and energy consumption	The incremental versions show significant improvement in performance for block(s) modification operation	The incremental versions consume more resources of mobile devices during the initial encryption and uploading phase
I-PReS [133], 2014	Confidentiality and integrity	Incremental cryptography	Android SDK, GAE, Google Cloud Storage, java Pairing-Based Cryptography library	Turnaround time, energy consumption, CPU utilization and memory consumption	Significant improvement in results while performing file modification operations using limited processing capability of mobile devices	The computational complexity of bilinear pairing exists in this scheme. The performance was overlooked for file modification operation within a file

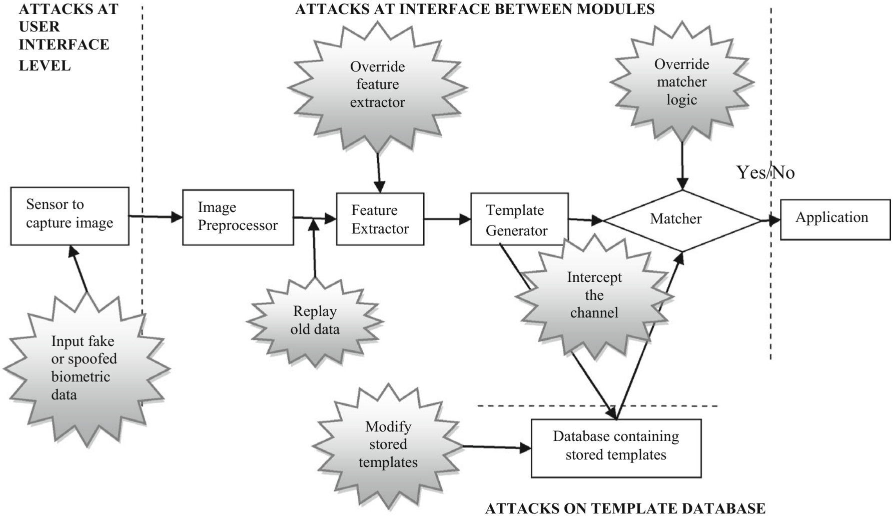
ATTACKS AT USER INTERFACE LEVEL
ATTACKS AT INTERFACE BETWEEN MODULES
Override feature extractor
Override matcher logic
Yes/No
Sensor to capture image
Image Preprocessor
Feature Extractor
Template Generator
Matcher
Application
Intercept the channel
Replay old data
Input fake or spoofed biometric data
Modify stored templates
Database containing stored templates
ATTACKS ON TEMPLATE DATABASE

Fig. 10 Biometric recognition process and associated vulnerabilities

several points where attacks may occur in a biometric recognition system as shown in Fig. 10:

- Presenting fake or spoofed biometrics at the sensor, for instance a fake finger or a face mask.
- A weak biometric system can allow an intruder to gain control over templates by launching replay attack.
- The features extracted from the source image can be replaced with a fraudulent feature set, i.e., overriding a feature extractor.
- The communication channel between database and matcher over which stored template is sent can be intercepted and modified.
- The matcher can also be compromised and corrupted to produce fake match scores.
- The stored templates in database can also be tampered either locally or remotely.

The various biometrics-based security systems have been designed by various researchers around unique characteristics of individuals. The probability of two people sharing the same biometric trait is virtually negligible. Eyeverify Inc. launched a biometric-based solution Eyeprint ID [48] for protecting mobile data without any additional hardware. It allows an individual to open a mobile device, logs into apps and perform payments securely. It provides a cost-efficient, extremely accurate and scalable private authentication solution to protect an individual's digital life. It uses 1+ megapixels smartphone camera to capture the features such as unique blood vessels around the white portion of eyes. It is more accurate than facial recognition and less expensive than sensors used for iris and fingerprint recognition. It can also be integrated with password to provide two-factor authentication.

Wang and Huaizhi [134] proposed an efficient and low-complexity solution to ensure cloud computing security by not revealing user's real facial data and face private matching identification result to the cloud. It encompasses three parts: user part that provides facial images; cloud initialization part containing face subspace that matches template database; and cloud private matching and identification part containing the core algorithm that compares two encrypted numbers under double-encrypted conditions. Derawi et al. [135] proposed a realistic biometric authentication scheme based on fingerprint recognition with embedded cameras on mobile device for user authentication. Most of the fingerprint recognition algorithms work well with high-resolution images, but the mobile camera outputs low-resolution images with high distortions which deteriorate system performance and increases error rate. The authors have applied the Neurotechnology and VeriFinger 6.0 extended SDK minutia extractor for the feature extraction and matching. This scheme aims at keeping error rate in an acceptable limit by lowering human effort involved in pre-processing and enhancing image quality. It captured 1320 fingerprint images from Nokia N95 and HTC desire that resulted in equal error rate of 4.5%.

Omri et al. [136] proposed handwritten password cloud-based authentication framework to allow secure access of data stored in cloud using mobile phone. The classifier employs parallel algorithms such as artificial neural network (ANN), K-nearest neighbor (K-NN) and Euclidean distance (ED) to improve accuracy of recognition and reduce error rate. The client side captures the handwritten image of password including writing speed and pressure profile using smartphone's touch screen and motion sensors and then encrypts this biometric image before sending to cloud. The authentication server at cloud pre-processes the image by performing various functions such as grayscaling, binarization, edge detection using Sobel algorithm, image dilation for expanding shapes, image segmentation, holes filling, extracting and normalizing digits. The different set of features are extracted from handwritten digits such as pixel density features, skeletonized features, speed and pressure profile which are fed as an input to ANN, K-NN and ED classifiers. The classifier algorithm uses the feature vector to verify an individual's identity. Each classifier is given different weight based on its classification accuracy. The final result is weighted fusion of individual decision from these classifiers.

Pawle and Pawar [137] proposed the facial recognition system (FRS) that encompasses two phases as shown in Fig. 11. One is new user registration phase in which user fills the registration form with necessary details when he/she wants to access cloud. FRS checks availability of username for uniqueness, and password is created by capturing facial image through Web camera or mobile camera. The next step is face detection which identifies the face in captured image eliminating other parts. The captured image needs to be aligned so that it is ready for recognition and its features are extracted to create a facial template which is stored in database. After registration process is completed, the registered user logs in to the cloud server by entering username and facial image is again captured and compared against stored template. If match occurs, access is given; otherwise, error message is displayed. Rassan and AlSaher [138] proposed SMCBA (securing mobile cloud computing with biometric authentication)-based solution that utilizes fingerprint recognition for secure access of data in mobile cloud. In SMCBA, fingertip image is captured by mobile camera without the aid of any external hardware. The captured image is pre-processed, and ridge structure is extracted. The template storage and matching are performed on cloud

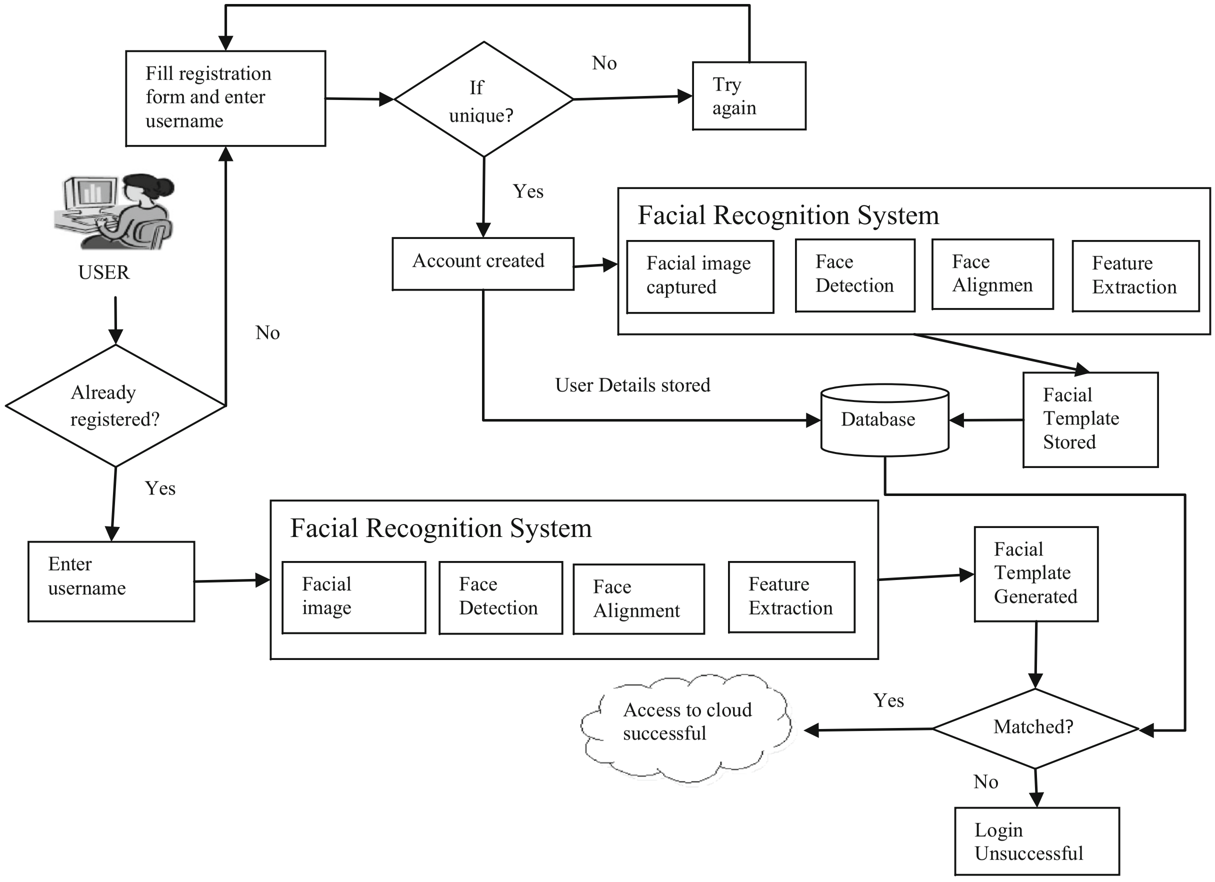

Fig. 11 FRS Architecture

that provides database with PaaS. It is immune to injection attacks as only fingertip image is taken as input from user.

Bommagani et al. [139] proposed a framework for cloud-empowered mobile biometrics which allow mobile device to perform facial recognition by offloading heavy computations to cloud. It takes benefit of available cloud resources by dividing and executing recognition and enrollment tasks such as biometric template generation and matching in parallel. It also generates the cancellable biometrics to ensure the privacy of biometric data stored in cloud. The framework utilizes Viola-Jones detector [140] for face detection and local binary pattern (LBP) [141] histograms for template generation. LBP histograms are converted to cancellable biometrics by applying mathematical operations on orthogonal matrix generated through Gram-Schmidt orthogonalization, permutation matrix and blinding vector. During matching phase, Euclidian distance of probe image represented as LBP-labeled image is compared against all the stored templates.

4.2 Multifactor-based security schemes in MCC

The biometrics, the password and the storage device are the core elements of three-factor authentication schemes. Fan et al. [142] proposed three-factor authentication scheme that involves user password, smart card and biometric information. It ensures security by storing encrypted string of user's identity in the remote server. This scheme is implemented in two steps: registration step and authentication step. In the first step, user template is encrypted by combining biometric characteristics with randomly chosen string using XOR operation. In second step, fingerprints obtained from sensor are encrypted using different randomly chosen string and these two strings are then sent for matching. This scheme preserves user's privacy in authentication phase as exact value of biometric data is not required for matching but is vulnerable to user's fingerprint minutia theft by internal dishonest staff during registration phase as certification authority is involved for issuing certificates for valid users.

Khan et al. [143] proposed key hash fingerprint-based remote authentication scheme for mobile devices which makes use of bitwise NOR operator to withstand password guessing attack. But it is vulnerable against user impersonation attack and de-synchronization attack. Wu et al. [144] proposed a biometric-based three-factor remote authentication scheme which ensures user's privacy for mobile client-server architecture to overcome the weakness of Khan's scheme [143]. The mobile user and server rely on ECCDH for calculating session key and symmetric cryptography for encryption and decryption. It employs fuzzy extractor to handle the imperfect biometrics input during the registration and the authentication processes. It is immune against insider attack because the password and the biometric secret string are both protected by the SHA and a random number. Their scheme is vulnerable to user impersonation attack in the registration phase and offline password guessing attack in the login and password change phase. It also fails to provide user revocation when the mobile device is lost or stolen. Jiang et al. [145] proposed a privacy-preserving three-factor authentication protocol by combining password, mobile device and biometrics for real-life application environments. This protocol enhances Wu's scheme [144] to provide missing security features and maintain the desired features of that scheme. The registration phase is modified to deal with revocation and re-registration. It also added revocation and re-registration phase to protect the system in case of lost/stolen mobile device. The login and password change phase are enhanced to defend offline password guessing attack which was demerit of Wu's scheme. Fuzzy extractor is used to protect the biometric template privacy. This scheme ensures mutual authentication, perfect forward secrecy, user anonymity and untraceability. The authors validate their protocol through Burrows–Abadi–Needham logic [146] and informal security analysis.

Xi et al. [147] presented an efficient fingerprint-based bio-cryptographic security protocol designed for client–server authentication for mobile computing platform. In this scheme, fingerprint templates are protected by PKI and ECC and are used for both verification and cryptographic key generation. They used VeriFinger 5.0 [148] for minutiae extraction and FOMFE [149,150] for singular point detection. The genuine fingerprint information is mixed with fake information to create a fuzzy vault list. This locally matched fuzzy vault index is sent for authentication to a remote server over an insecure network. After successful mutual authentication, communication between mobile user and server is protected by biometric-based symmetric session key. Wang et al. [151] proposed a remote authentication scheme based on secret splitting concept which provides an efficient way of enhancing cloud security by protecting against insider dishonest staff as well as hacking certificate authority. The biometric data are divided into two parts: One part is encrypted and stored on cloud, while other is encrypted independently and stored on authentication database server. The biometric data matching is done at the terminal so there is a threat of the template leakage in this scheme. This scheme is comprised of three different phases: initialization phase, registration phase and authentication phase as follows [151,152]:

- In initialization phase, smart card manufacturer embeds security parameters (p, α, AN, K) onto smart cards where p is a large prime number, α is its root, K is 128-bit symmetric key, and AN is authentication number based on pre-defined coding rule and forwards card to certificate authority for distribution to various clients.
- In registration phase, an individual user gets itself registered to cloud server by sending password and fingerprint image. The terminal applies hash function independently to both parts of fingerprint template and sends one part to user, whereas other to authentication server.
- In authentication phase, user inserts his smart card containing a partial encrypted template and access request is sent to authentication server. The session key is exchanged between terminal and server. The server generates symmetric key using AES for comparing hash value of encrypted template with that of stored information.

Chen et al. [153] proposed a fingerprint and password-based remote authentication scheme for implementing mutual authentication with low computation overhead instead of smart card based or biometrics based. The user sends identity, password, nonce value and fingerprint template for registration through secure communication channel, e.g., SSL to remote server. The secret key is protected by one-way hash functions. The remote server did not save any password, biometric database or verification table making it immune to stolen-verifier attack and sends back some secret information which is stored on mobile device. The mobile device generated new nonce during each login to prevent replay attack. This secret information is later on used for proving its identity to sever. Cheng et al. [154] designed an integrated secure data access scheme based on identity-based cryptography and biometric. The integrated process consists of parameter setup, key distribution, feature template creation, cloud data processing and secure data access. Cloud computing can easily handle the computational complexity of this scheme although it may be a drawback for smart network, such as wireless sensor network. There is high computational overhead involved in this scheme. The client data will be exposed when the client has leaked his login information and the adversary has made fingerprint falsification successfully.

Table 4 highlights comparison of various multifactor authentication schemes in MCC, techniques and tools used by them, their merits and demerits.

5 Open research directions

Even though migrating to the mobile cloud is a tempting trend from a financial perspective, protecting user privacy and data secrecy from an adversary is an essential factor for the success of MCC paradigm. Since mobile devices are resource-constrained, protecting them from numerous security threats is more difficult than that for resourceful devices. Mobile devices are deployed in heterogeneous networks where mobile nodes access the cloud through various network interfaces and radio technologies such as GPRS, CDMA, WCDMA, WLAN, WiMAX. Due to heterogeneity, there arises an issue of handling wireless connectivity while designing security solution which should satisfy certain MCC requirements such as always-on connectivity, on-demand scalability and energy efficiency of mobile nodes. Several studies have been conducted for data security in cloud computing to build a level of trust between cloud service providers and consumers, but these solutions may not be much effective for MCC environment, and there exist many challenges in the security policies as MCC paradigm is still in the preliminary stages of research. This section presents some open issues that serve as a platform for future research works.

- **Mobile identity management and access (IAM) control:** Mobile linked to cloud computing allows user to access data and services from various endpoint devices running on different platforms. For example, content created on Android Tablet is accessed from Apple iPhone. The cloud servers being highly virtualized and federated are not visualized as trustworthy so they need identity management and fine-grained access control policies allowing encrypted data to be disclosed to authorized users only. The need to support multiple mobile identities including creating adequate trust frameworks for mobile devices and securing data outside the perimeter of an organization's direct control remains one of the greatest challenges facing MCC today. Many IAM solutions such as open authentication (OAuth), lightweight directory access protocol (LDAP), security assertion markup language (SAML), openID connect (OIDC) have been proposed for cloud users. But MCC IAM requirements are different from cloud IAM due to resource limitations and mobile communications, so aforementioned solutions may not be much

Table 4 Comparison of multifactor authentication schemes in MCC

Scheme	Technique used	Tools used	Performance metrics	Merits	Demerits
Remote Password Authentication Scheme with Smart Cards and Biometrics [142], 2006	Password, smart card, biometric	Theoretical analysis	Computation cost in login, registration and authentication phase	Secure protocols allow the credentials to freely roam in cloud computing environment. It can withstand replay and offline dictionary attack	Credential store is the repository for credentials, posing serious threat of being hacked
Key hash-based fingerprint remote authentication scheme [143], 2014	Fingerprint biometric, password and hashing	Formal security analysis	Computation cost in login, registration and authentication phase, memory required by mobile device and communication cost	It can withstand server impersonation attack using an intercepted login request through ID guessing, mobile device loss attack, DoS attack, replay attack and password guessing attack	It lacks user anonymity and strong forward security. It is vulnerable against user impersonation attack and de-synchronization attack
Provably secure biometrics-based scheme for mobile client-server networks [144], 2015	ECC, Fuzzy Extractor, SHA1, AES	SHA1 with value length 160 bits, AES with block size 128 bits	Communication cost, and the transmission times in login and authentication phases, encryption/decryption time	It is resistance to the insider attack, offline guessing attack, server spoofing attack and de-synchronization attack. It provides user anonymity	It is vulnerable to user impersonation attack in the registration phase only and offline password guessing attack in the login and password change phase if mobile device is lost

Table 4 continued

Scheme	Technique used	Tools used	Performance metrics	Merits	Demerits
Privacy-preserving protocol for e-health clouds [145], 2016	ECC, Fuzzy Extractor, SHA1, AES	Informal security analysis using BAN logic	Time complexity for scalar multiplication of ECC, symmetric key encryption/decryption, hash function during registration phase, login phase and authentication phase	It is immune against insider attack, online/offline guessing attack with stolen mobile device, user impersonation attack in registration phase and MITM attack. It provides user anonymity. There is a provision for revocation and re-registration	The computation cost is higher than Wu’s scheme [144] as it involves more scalar multiplications of ECC
A fingerprint-based bio-cryptographic security Protocol [147], 2011	Fingerprint biometric are protected by PKI and ECC	MATLAB, CLDC emulator using Java ME, Sun Java Wireless Toolkit (WTK) 2.5	Computational time of messages with different lengths and memory usage	Cryptographic key cannot be independently generated without biometric information. It is robust against Trojan horse attacks, brute force attack, replay attack and MITM attack	Sever side attacks were not considered. Sophisticated fingerprint matching algorithms and cancelable biometric template protection was not given due consideration

Table 4 continued

Scheme	Technique used	Tools used	Performance metrics	Merits	Demerits
Fingerprint Authentication Scheme Based on Secret Splitting [151], 2011	Secret splitting, symmetric encryption (128-bit AES), Diffie–Hellman key exchange, Authentication Server	Formal security analysis and comparative mathematical analysis	Computational cost of login and authentication phases	The user's identity is protected even if the card is lost or stolen. It is robust toward network attacks like MITM, replay, dictionary attacks and interior identity thefts originating from centralized authority issuing cards	High computation cost compared to existing schemes. CA cannot re-issue new smart card immediately in events of lost/stolen cards as they do not possess the complete fingerprint template
Mobile device integration of fingerprint biometric remote authentication scheme [153], 2012	Fingerprint biometric, password and hashing	Comparative mathematical analysis	Communication cost and computational complexity in login and authentication phase	Low communication and computational complexity. It can withstand various attacks such as insider, impersonation, DoS	It does not provide user anonymity. It is vulnerable against the forgery attack, replay attack and offline password guessing attack
IBE and Biometric Authentication Scheme [154], 2012	IBE-160 bit, and biometric authentication	Hadoop, Intel core Duo 2 CPU E7500 processor and Aralek fingerprint capturing and identification devices	Average encryption and decryption time, data security probability of client data under ID, password leakage and CSPs maloperation probability	Security level has little change under different leakage probability of cloud client ID and password so it outperforms role-based access control scheme	Few parameters are considered for evaluation and high computational overhead

effective for MCC. There is a need to implement new IAM solutions that meet the requirements of mobile and cloud users and allow dynamic changes to the access policies.

- **Encryption and key management:** Securing applications running on mostly uncontrolled mobile environments is a real challenge. Designing secure resource-efficient context-aware applications to improve QoS is an essential requirement for unleashing the power of mobile cloud computing in heterogeneous environment toward unrestricted ubiquitous computing. Before outsourcing data in cloud, data encryption can prove to be an efficient way for ensuring confidentiality and protecting data. Key management (key generation, distribution and monitoring) is a critical element for data encryption in MCC environment. One of the future directions of research is leveraging cloud for encryption and key management operations which lowers overhead on mobile devices. The widespread use of mobile devices, cloud computing, virtual storage among organizations and consumers generates a huge amount of data which are vulnerable to loss and theft. This creates new opportunities for growth of database encryption market in various forms such as multilevel, file-level, column-level encryptions and key management. The efficient and robust key management approaches is also a necessity which could extend traditional approaches such as CP-ABE, KP-ABE, CL-PKC to mobile cloud computing. Since encrypted data are in unintelligible form, our traditional keyword search techniques are slow because of heavy computational overhead. Coupled this with the fact that the cloud servers majorly employ data stores after encryptions, puts a big question mark on the usability of these enormous data coffers. Searchable encryption provides searching operations in encrypted data for secure retrieval of data stored in cloud and holds a promising future. We need privacy-preserving keyword search techniques over encrypted data for efficient retrieval of data.
- **Remote data auditing:** We inhabit in an era of large-scale data where trillions of files and zettabytes of data are stored in the remote distributed data servers. Therefore, large-scale data auditing scheme which can minimize communication, processing and storage overhead is primarily an open issue. Designing efficient remote data auditing protocols for verifying integrity of data stored on the cloud without downloading the copy of data on mobile device is an important security issue gaining attention of researchers' community. Further, dynamic data updates being a crucial facet of remote data auditing methods both for single and distributed cloud servers become imperative to reduce the computation and communication overhead on mobile user and cloud server. Among various types of distributed-based remote data auditing approaches, such as replication based, erasure coding based, and network coding based, most of the existing methods belong to the replication-based category. Therefore, implementing a dynamic data auditing for network coding-based distributed servers using incremental cryptography serves as prominent upcoming research challenge for further investigation.
- **Security and privacy of IoT and MCC:** The Internet of Things (IoT) is a new technology **taking the world by storm.** The integration of MCC and IoT is creating an explosion of trillions of smart and powerful cloud-ready devices amplifying information security and reliability issues. The interconnection of various devices

with no embedded security assists hackers in creating botnets to launch high-impact DDoS attacks against various Internet services. We all are amateurs when it comes to the novel field of IoT and MCC which in turn provides an opportunity to unveil and unravel challenges with contemporary ideas and solutions for wide deployment and proliferation of IoT devices.

- **Post-quantum cryptography:** The security of public key cryptographic schemes such as DSA, RSA, ECC, class groups depends on hard mathematical problems such as prime factorization of large integers and computing discrete logarithm. Fifteen years or so down the line, a possible inception of quantum computers puts the aforementioned security schemes in jeopardy. These schemes can be easily compromised by running Shor's algorithm [155] on large quantum computers in the near future. The post-quantum cryptographic systems relying on symmetric cryptography, code-based cryptography, lattice-based cryptography and Merkle hash-based signature [156] might hold the key to unlock the predicament being faced.
- **Biometric systems in jeopardy:** Existing digital lockers rely on cloud servers for biometric matching which provides the decryption key to the client after successful authentication. But if server is hacked, both encryption keys and biometric data are disclosed leading to great havoc. One can change the lost or stolen password or change a door lock if key is lost. What if your bank account is locked using the image of your palm and this database of palm prints is stolen? It is imperative to take all these factors into consideration before designing such systems which otherwise might betray us with a false sense of security.
- **Multimodal biometric system for authentication:** Biometric recognition systems involving single sources of information known as unimodal systems show satisfying performance, but still suffer from inherent limitations of biometric traits of non-universality, permanence, susceptibility to circumvention and often fail to correctly verify an individual with desired level of accuracy. Multimodal biometric system is a promising area offering solutions in crucial times when human lives are at stake in the healthcare, financial and surveillance industries. The designing of such solutions would involve fusion of multiple biometric traits such as fingerprint, iris to correctly authenticate an individual with enhanced level of accuracy and security.

6 Conclusion

The survey provides a comprehensive and structured review of various security issues involved in deploying mobile applications on cloud and critically investigates existing security frameworks proposed to address these security issues. The comparative analysis based on the strength and weakness of existing predominant solutions suggest the need for futuristic security mechanisms to leverage heterogeneous cloud resources and mobile device capabilities in an efficient way. This paper also highlights future research issues that open up space for extending existing techniques and devising new techniques for security and privacy in MCC. Biometric techniques for identifying mobile users using one or more unique physical or behavioral traits are an emerging area of research in MCC environment. There is a need to design a robust and secure data access scheme integrating cryptography and biometric remote authentication to transform a normal mobile cloud computing platform to more trusted platform. This will assure enhancement in security and privacy, secure data transmission, more flexibility and capability to meet the new demand of today's complex and diverse network.

References

1. Abolfazli S et al (2013) Cloud-based augmentation for mobile devices: motivation, taxonomies, and open challenges. IEEE Commun Surv Tutor 16(1):1–32
2. Sanaei Z, Abolfazli S, Gani A, Buyya R (2014) Heterogeneity in mobile cloud computing: taxonomy and open challenges. IEEE Commun Surv Tutor 16(1):369–392
3. Dinh HT, Lee C, Niyato D, Wang P (2013) A survey of mobile cloud computing: architecture, applications, and approaches. Wirel Commun Mob Comput 13(18):1587–1611
4. Williams B (2012) What is cloud computing?—The journey to cloud. In: Economics of Cloud Computing: An Overview for Decision Makers, 1st edn. Cisco Press, Indiana, pp 1–13
5. Branch R et al (2014) Cloud computing and big data: a review of current service models and hardware perspectives. J Softw Eng Appl 7(8):686–693
6. Contu R, Kavanagh KM (2013) Market Trends: Cloud-Based Security Services Market Worldwide, Gartner, G00253813
7. Cisco Visual Networking Index (2016) Global mobile data traffic forecast update, 2015–2020. White paper, 3rd February 2016
8. Mobile BPM Market by Solution, Service (Maintenance and Support, Integration and Design, Consulting, and Others), End User (SMBs and Enterprises), Vertical, Deployment Model (Public, Private, Hybrid), and Region—Global Forecast to 2020, Markets and Markets. http://www.marketsandmarkets.com/Market-Reports/mobile-bpm-market-31726258.html. Accessed 15 Feb 2016
9. Rahimi MR et al (2014) Mobile cloud computing: a survey, state of art and future directions. Mob Netw Appl 19(2):133–143
10. Fernando N, Loke SW, Rahayu W (2013) Mobile cloud computing: a survey. Future Gener Comput Syst 29(1):84–106
11. Lei L (2013) Challenges on wireless heterogeneous networks for mobile cloud computing. IEEE Wirel Commun 20(3):34–44
12. Alizadeh M, Hassan WH (2013) Challenges and opportunities of mobile cloud computing. In: Proceedings of the 9th International Wireless Communications and Mobile Computing Conference (IWCMC). Sardinia, Italy, pp 660–666
13. Ra MR et al (2011) Odessa: enabling interactive perception applications on mobile devices. In: Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services. ACM, Bethesda pp 43–56
14. Kovachev D, Cao Y, Klamma R (2011) Mobile cloud computing: a comparison of application models. Comput Res Repos, CoRR. arXiv preprint. arXiv:1107.4940
15. Guan L, Ke X, Song M, Song J (2011) A survey of research on mobile cloud computing. In: Proceedings of the 10th IEEE/ACIS International Conference on Computer and Information Science, (ICIS'11). Sanya, China, pp 387–392
16. Rahimi MR, Venkatasubramanian N, Vasilakos AV (2013) Music: mobility-aware optimal service allocation in mobile cloud computing. In: Proceedings of the 6th International Conference on Cloud Computing. IEEE, Santa Clara, pp 75–82
17. Kumar K, Liu J, Lu YH, Bhargava B (2013) A survey of computation offloading for mobile systems. ACM/Springer MONET 18(1):129–140
18. Xia F et al (2014) Phone2Cloud: exploiting computation offloading for energy saving on smartphones in mobile cloud computing. Inf Syst Front 16(1):95–111
19. Yang S et al (2013) Fast dynamic execution offloading for efficient mobile cloud computing. In: Proceedings of the International Conference on Pervasive Computing and Communications (PerCom). IEEE, San Diego, pp 20–28

20. Yang S et al (2014) Techniques to minimize state transfer costs for dynamic execution offloading in mobile cloud computing. IEEE Trans Mob Comput 13(11):2648–2660
21. Kaewpuang R, Niyato D, Wang P, Hossain E (2013) A framework for cooperative resource management in mobile cloud computing. IEEE J Sel Areas Commun 31(12):2685–2700
22. Suo H, Liu Z, Wan J, Zhou K (2013) Security and privacy in mobile cloud computing. In: Proceedings of the 9th International Wireless Communications and Mobile Computing Conference (IWCMC). Sardinia, Italy, pp 655–659
23. Shahzad A, Hussain M (2013) Security issues and challenges of mobile cloud computing. Int J Grid Distrib Comput 6(6):37–50
24. Xiao Z, Xiao Y (2013) Security and privacy in cloud computing. IEEE Commun Surv Tutor 15(2):843–859
25. Sun Y, Zhang J, Xiong Y, Zhu G (2014) Data security and privacy in cloud computing. Int J Distrib Sens Netw 2014:1–9. doi:10.1155/2014/190903
26. Gonzalez N et al (2012) A quantitative analysis of current security concerns and solutions for cloud computing. J Cloud Comput 1(1):1–18
27. Shaikh R, Sasikumar M (2015) Trust model for measuring security strength of cloud computing service. Procedia Comput Sci 45:380–389
28. Li J et al (2015) Identity-based encryption with outsourced revocation in cloud computing. IEEE Trans Comput 64(2):425–437
29. Sood SK (2012) A combined approach to ensure data security in cloud computing. J Netw Comput Appl 35(6):1831–1838
30. Kumar R, Rajalakshmi S (2013) Mobile cloud computing: standard approach to protecting and securing of mobile cloud ecosystems. In: Proceedings of the International Conference on Computer Sciences and Applications (CSA). Wuhan, China, pp 663–669
31. Khan AN, Kiah MLM, Khan SU, Madani SA (2013) Towards secure mobile cloud computing: a survey. Future Gener Comput Syst 29(5):1278–1299
32. Alizadeh M et al (2016) Authentication in mobile cloud computing: a survey. J Netw Comput Appl 61(2):59–80
33. Somorovsky J et al (2011) All your clouds are belong to us: security analysis of cloud management interfaces. In: Proceedings of the 3rd ACM Workshop on Cloud Computing Security Workshop, New York, USA, pp 3–14
34. Sermersheim J (2006) Lightweight directory access protocol (LDAP): the protocol. RFC 4511. http://www.ietf.org/rfc/rfc4511.txt. Accessed 10 Jan 2016
35. Grobauer B, Walloschek T, Stocker E (2011) Understanding cloud-computing vulnerabilities. IEEE Secur Priv 9(2):50–57
36. Liu F et al (2013) Gearing resource-poor mobile devices with powerful clouds: architecture, challenges and applications. IEEE Wirel Commun Mag 20(3):14–22
37. Modi C et al (2013) A survey on security issues and solutions at different layers of Cloud computing. J Supercomput 63(2):561–592
38. Jansen W, Grance T (2011) Guidelines on security and privacy in public cloud computing. NIST Spec Publ 800(144):10–11
39. Anati I, Gueron S, Johnson S, Scarlata V (2013) Innovative technology for CPU based attestation and sealing. In: Proceedings of the 2nd International Workshop on Hardware and Architectural Support for Security and Privacy. Tel Aviv, Israel, pp 1–7
40. Zissis D, Lekkas D (2012) Addressing cloud computing security issues. Future Gener Comput Syst 28(3):583–592
41. Lamport L (1981) Password authentication with insecure communication. Commun ACM 24(11):770–772
42. Ren W, Zeng L, Liu R, Cheng C (2016) F2AC: a lightweight, fine-grained, and flexible access control scheme for file storage in mobile cloud computing. Mob Inf Syst 2016:1–9. doi:10.1155/2016/5232846
43. Messerges TS, Dabbish EA, Sloan RH (2002) Examining smart-card security under the threat of power analysis attacks. IEEE Trans Comput 51(5):541–552
44. Shoup V, Rubin A (1996) Session key distribution using smart cards. In: Proceedings of the 15th Annual International Conference on Theory and Application of Cryptographic Techniques (EUROCRYPT 96). Saragossa, Spain, pp 321–331

45. Yang G et al (2008) Two-factor mutual authentication based on smart cards and passwords. J Comput Syst Sci 74(7):1160–1172
46. Halevi S, Krawczyk H (1999) Public-key cryptography and password protocols. ACM Trans Inf Syst Secur 2(3):230–268
47. Hwang MS, Li LH (2000) A new remote user authentication scheme using smart cards. IEEE Trans Consum Electron 46(1):28–30
48. Eyeprint ID. http://www.eyeverify.com/product-technology/core-technology. Accessed 3 April 2016
49. BioID Web Services (BWS). http://www.bioid.com/products/bws.html. Accessed 3 April 2016
50. Face.com Opens Free Facial Recognition API. http://www.programmableweb.com/api/face.com. Accessed 3 April 2016
51. Animetrics Face Recognition (FaceR) API. http://animetrics.com/facer-api-for-face-recognition-applications/. Accessed 3 April 2016
52. Cloud Based Biometric Authentication Solution for an Online Testing and Assessment company. http://www.calsoftlabs.com/resources/cloud-biometric-authentication-solution.html. Accessed 3 April 2016
53. Zonouz S et al (2013) Secloud: a cloud based comprehensive and lightweight security solution for smartphones. Comput Secur 37(9):215–227
54. Simmons GJ (1979) Symmetric and asymmetric encryption. ACM Comput Surv (CSUR) 11(4):305–330
55. Neuman BC, Ts'O Theodore (1994) Kerberos: an authentication service for computer networks. IEEE Commun Mag 32(9):33–38
56. Stallings W (2006) Cryptography and network security: principles and practice. Prentice Hall, Upper Saddle River
57. Krawczyk H, Canetti R, Bellare M (1997) HMAC: Keyed-hashing for message authentication. https://tools.ietf.org/html/rfc2104. Accessed 10 Jan 2015
58. Pirzada AA, McDonald C (2004) Kerberos-assisted authentication in mobile ad hoc networks. In: Proceedings of the 27th Australasian Conference on Computer Science (ACSC '04). Dunedin, New Zealand, pp 41–46
59. Carman DW, Kruus PS, Matt BJ (2000) Constraints and approaches for distributed sensor network security. DARPA Project report (Cryptographic Technologies Group, Trusted Information System, NAI Labs) 1(1)
60. Ali M et al (2015) SeDaSC: secure data sharing in clouds. IEEE Syst J PP(99):1–10. doi:10.1109/JSYST.2014.2379646
61. Ren W et al (2011) Lightweight and compromise resilient storage outsourcing with distributed secure accessibility in mobile cloud computing. J Tsinghua Sci Technol 16(5):520–528
62. Khan AN et al (2014) BSS: block-based sharing scheme for secure data storage services in mobile cloud environment. J Supercomput 70(2):946–976
63. Diffie W, Hellman ME (1976) New directions in cryptography. IEEE Trans Inf Theory 22(6):644–654
64. Koscielny C, Kurkowski M, Srebrny M (2013) Public key infrastructure. In: Modern Cryptography Primer. Springer, Berlin, pp 175–191. doi:10.1007/978-3-642-41386-5_7
65. Hellman M (1978) An overview of public key cryptography. IEEE Commun Mag 16(6):24–32
66. Rivest R, Shamir A, Adleman L (1978) A method for obtaining digital signatures and public-key cryptosystems. Commun ACM 21(2):120–126
67. ElGamal T (1985) A public-key cryptosystem and a signature scheme based on discrete logarithms. IEEE Trans Inf Theory 31(4):469–472
68. Miller V (1985) Use of elliptic curves in cryptography. CRYPTO. Lect Notes Comput Sci 85:417–426
69. Lamport L (1979) Constructing digital signatures from a one-way function. Technical Report CSL-98, SRI International, Palo Alto
70. Merkle R (1990) A certified digital signature. In: Gilles Brassard G (ed) Advances in Cryptology—CRYPTO '89. Springer, Berlin, pp 218–238
71. Mehuron W (1994) Digital Signature Standard (DSS). US Department of Commerce, National Institute of Standards and Technology (NIST). Information Technology Laboratory (ITL). FIPS PEB 1994:186
72. Han JH, Kim YJ, Jun SI, Chung KI, Seo CH (2002) Implementation of ECC/ECDSA cryptography algorithms based on Java card. In: Proceedings of the 22nd International Conference on Distributed Computing Systems Workshops. Austria, Vienna, pp 272–276

73. Lee S et al (2010) Two factor authentication for cloud computing. J Inf Commun Converg Eng 8(4):427–432
74. Xiao S, Gong W (2010) Mobility can help: protect user identity with dynamic credential. In: Proceedings of the 11th International Conference on Mobile Data Management (MDM). IEEE, Kansas City, pp 378–380
75. Khan AN, Kiah MM, Madani SA, Ali M (2013) Enhanced dynamic credential generation scheme for protection of user identity in mobile-cloud computing. J Supercomput 66(3):1687–1706
76. Khan AN et al (2013) A study of incremental cryptography for security schemes in mobile cloud computing environments. In: Proceedings of the IEEE Symposium on Wireless Technology and Applications (ISWTA). Kuching, Malaysia. IEEE, pp 62–67
77. Khalid U et al (2013) Cloud based secure and privacy enhanced authentication & authorization protocol. Procedia Comput Sci 22:680–688
78. Annappaian DH, Agrawal VK (2015) Multilevel cryptography with metadata and lock approach for storing data in cloud. Trans Netw Commun 2(6):47–55
79. Lauter K (2004) The advantages of elliptic curve cryptography for wireless security. IEEE Wirel Commun 11(1):62–67
80. Toorani M, Beheshti A (2008) LPKI-a lightweight public key infrastructure for the mobile environments. In: Proceedings of the 11th International Conference on Communication Systems (ICCS). IEEE, Guangzhou, pp 162–166
81. Krawczyk H (2005) HMQV: a high-performance secure Diffie-Hellman protocol. In: Advances in Cryptology—CRYPTO'05. LNCS, vol 3621. Springer, Berlin, pp 546–566
82. Baek J, Steinfeld R, Zheng Y (2007) Formal proofs for the security of signcryption. J Cryptol 20(2):203–235
83. Myers M et al (1999) X.509 Internet Public Key Infrastructure Online Certificate Status Protocol-OCSP, RFC 2560. http://www.ietf.org/rfc/rfc2560.txt. Accessed 10 March 2016
84. Pinkas D, Housley R (2002) Delegated Path Validation and Delegated Path Discovery Protocol Requirements, RFC 3379. http://www.ietf.org/rfc/rfc3379.txt. Accessed 12 March 2016
85. Boneh D, Franklin MK (2001) Identity-based encryption from the Weil pairing. In: Kilian J (ed) Advances in Cryptology-CRYPTO 2001. Springer, Berlin, pp 213–229
86. Boneh D, Franklin MK (2003) Identity-based encryption from the Weil pairing. SIAM J Comput 32(3):586–615
87. Gentry C, Silverberg A (2002) Hierarchical ID-based cryptography. In: Zheng Y (ed) Proceedings of Asiacrypt 2002. Springer, Berlin, pp 548–566
88. Ragini P, Mehrotra S, Venkatesan, (2014) An efficient model for privacy and security in Mobile Cloud Computing. In: Proceedings of the International Conference on Recent Trends in Information Technology (ICRTIT). Chennai, India, pp 1–6
89. Li H, Dai Y, Tian L, Yang H (2009) Identity-based authentication for cloud computing. In: Jaatun M (ed) Cloud Computing. Springer, Berlin, pp 157–166
90. Dong X et al (2015) SECO: secure and scalable data collaboration services in cloud computing. Comput Secur 50(5):91–105
91. Liu W et al (2015) SAKE: scalable authenticated key exchange for mobile e-health networks. Secur Commun Netw. doi:10.1002/sec.1198
92. Sahai A, Waters B (2005) Fuzzy identity-based encryption. In: Cramer R (ed) Advances in Cryptology-EUROCRYPT. Springer, Berlin, pp 457–473
93. Goyal V, Pandey O, Sahai A, Waters B (2006) Attribute-based encryption for fine-grained access control of encrypted data. In: Proceedings of the 13th ACM Conference on Computer and Communications Security. Alexandria, Virginia, USA, pp 89–98
94. Lv Z, Hong C, Zhang M, Feng D (2012) A secure and efficient revocation scheme for fine-grained access control in cloud storage. In: Proceedings of the 4th International Conference on Cloud Computing Technology and Science (CloudCom'12). Taipei, Chennai, India, pp 545–550
95. Yang K, Jia X, Ren K, Zhang B, Xie R (2013) DAC-MACS: effective data access control for multi-authority cloud storage systems. IEEE Trans Inf Forensics Secur 8(11):1790–1801
96. Bethencourt J, Sahai A, Waters B (2007) Ciphertext-policy attribute-based encryption. In: Proceedings of 2007 IEEE Symposium on Security and Privacy. California, USA, Berkeley, pp 321–334
97. Hur J, Noh DK (2011) Attribute-based access control with efficient revocation in data outsourcing systems. IEEE Trans Parallel Distrib Syst 22(7):1214–1221

98. Yu S, Wang C, Ren K, Lou W (2010) Achieving secure, scalable, and fine-grained data access control in cloud computing. In: Proceedings of IEEE Conference on Computer Communications. San Diego, California, USA, pp 1–9
99. Sun GZ (2011) CP-ABE based data access control for cloud storage. J China Inst Commun 32(7):146–152
100. Sandhu RS, Coyne EJ, Feinstein HL, Youman CE (1996) Role-based access control models. Computer 29(2):38–47
101. Huang D et al (2010) Mobicloud: building secure cloud framework for mobile computing and communication. In: Proceedings of the 5th IEEE International Symposium on Service Oriented System Engineering (SOSE). Nanjing, China, pp 27–34
102. Zhou Z, Huang D (2011) Efficient and secure data storage operations for mobile cloud computing. In: Proceedings of the 8th International Conference on Network and Service Management. Laxenburg, Austria, pp 37–45
103. Yu S, Wang C, Ren K, Lou W (2010) Attribute based data sharing with attribute revocation. In: Proceedings of the 5th ACM Symposium on Information. Computer and Communications Security, Beijing, China, pp 261–270
104. Li F, Rahulamathavan Y, Rajarajan M, Phan RC (2013) Low complexity multi-authority attribute based encryption scheme for mobile cloud computing. In: Proceedings of the 7th International Symposium on Service Oriented System Engineering (SOSE). IEEE, San Francisco, pp 573–577
105. Chase M, Chow SS (2009) Improving privacy and security in multi-authority attribute-based encryption. In: Proceedings of the 16th ACM Conference on Computer and Communications Security (CCS '09). Illinois, USA, Chicago, pp 121–130
106. Lv Z, Chi J, Zhang M, Feng D (2014) Efficiently attribute-based access control for mobile cloud storage system. In: Proceedings of the 13th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom). IEEE, Beijing, pp 292–299
107. Fei L, Rahulamathavan Y, Rajarajan M (2014) LSD-ABAC: Lightweight static and dynamic attributes based access control scheme for secure data access in mobile environment. In: Proceedings of the 39th Conference on Local Computer Networks (LCN). IEE, Edmonton, Alberta, Canada, pp 354–361
108. Dong X et al (2014) Achieving an effective, scalable and privacy-preserving data sharing service in cloud computing. Comput Secur 42(5):151–164
109. Wu W et al (2015) Towards secure and cost-effective fuzzy access control in mobile cloud computing. Soft Comput, pp 1–7. doi:10.1007/s00500-015-1964-2
110. Al-Riyami S, Paterson K (2003) Certificateless public key cryptography. In: Laih CS (ed) Advances in Cryptology-ASIACRYPT. Springer, Berlin, pp 452–473
111. Xu L, Wu X, Zhang X (2012) CL-PRE: a certificateless proxy re-encryption scheme for secure data sharing with public cloud. In: Proceedings of the 7th ACM Symposium on Information. Seoul, Republic of Korea, Computer and Communications Security, pp 87–88
112. Seo SH, Nabeel M, Ding X, Bertino E (2014) An efficient certificateless encryption for secure data sharing in public clouds. IEEE Trans Knowl Data Eng 26(9):2107–2119
113. Yang C, Wang F, Wang X (2007) Efficient mediated certificates public key encryption scheme without pairings. In: Proceedings of the 21st International Conference on Advanced Information Networking and Applications Workshops (AINAW'07). Niagara Falls, Ontario, Canada, pp 109–112
114. Tsai J (2015) A new efficient certificateless short signature scheme using bilinear pairings. IEEE Syst J PP(99):1–8. doi:10.1109/JSYST.2015.2490163
115. Zhu L, Tung B (2006) RFC 4556: public key cryptography for initial authentication in Kerberos (PKINIT). IETF Network Working Group
116. Harbitter A, Menasce DA (2001) The performance of public key-enabled Kerberos authentication in mobile computing applications. In: Proceedings of the 8th ACM Conference on Computer and Communications Security. Pennsylvania, Philadelphia, pp 78–85
117. Fox A, Gribble SD (1996) Security on the move: indirect authentication using Kerberos. In: Proceedings of the 2nd Annual International Conference on Mobile Computing and Networking. Rye, New York, USA, pp 155–164
118. Park KW, Lim SS, Park KH (2008) Computationally efficient PKI-based single sign-on protocol, PKASSO for mobile devices. IEEE Trans Comput 57(6):821–834
119. Yang J et al (2011) Provable data possession of resource constrained mobile devices in cloud computing. J Netw 6(7):1033–1040

120. Wang Q et al (2009) Enabling public verifiability and data dynamics for storage security in cloud computing. In: Proceedings of the 14th European Conference on Research in Computer Security (ESORICS '09). Saint-Malo, France, pp 355–370
121. Carts DA (2011) A review of the Diffie–Hellman algorithm and its use in secure internet protocols. http://www.sans.org/reading_room/whitepapers/vpns/review-diffiehellman-algorithm-secure-internet-protocols_751. Accessed 10 April 2016
122. Nagaty KA (2014) Mobile health care on a secured hybrid cloud. J Sel Areas Health Inform 4(2):1–9
123. Sujithra M, Padmavathi G, Narayanan S (2015) Mobile device data security: a cryptographic approach by outsourcing mobile data to cloud. Procedia Comput Sci 47:480–485
124. Blaze M, Bleumer G, Strauss M (1998) Divertible protocols and atomic proxy cryptography. In: Nyberg K (ed) Advances in Cryptology–EUROCRYPT'98. Springer, Berlin, pp 127–144
125. Green M, Ateniese G (2007) Identity-based proxy reencryption. In: Katz J (ed) Applied Cryptography and Network Security. Springer, Berlin, pp 288–306
126. Jia W et al (2011) SDSM: a secure data service mechanism in mobile cloud computing. In: Proceedings of the IEEE Conference on Computer Communications Workshops. Shanghai, China, pp 1060–1065
127. Tysowski PK, Hasan MA (2011) Re-encryption-based key management towards secure and scalable mobile applications in clouds. In: IACR Cryptology ePrint Archive 668
128. Khan AN, Kiah MM, Ali M, Shamshirband S (2015) A cloud-manager-based re-encryption scheme for mobile users in cloud environment: a hybrid approach. J Grid Comput 13(4):651–675
129. Bellare M, Goldreich O, Goldwasser S (1994) Incremental cryptography: the case of hashing and signing. Advances in Cryptology-CRYPTO'94. Springer, Berlin, pp 216–233
130. Bellare M, Goldreich O, Goldwasser S (1995) Incremental cryptography and application to virus protection. In: Proceedings of the 27th Annual ACM Symposium on Theory of Computing. ACM, Riva, pp 45–56
131. Bellare M, Micciancio D (1997) A new paradigm for collision-free hashing: incrementality at reduced cost. Advances in Cryptology-EUROCRYPT'97. Springer, Berlin, pp 163–192
132. Itani W, Kayssi A, Chehab A (2010) Energy-efficient incremental integrity for securing storage in mobile cloud computing. In: Proceedings of the International Conference on Energy Aware Computing (ICEAC '10). IEEE, Cairo, pp 1–2
133. Khan AN et al (2014) Incremental proxy re-encryption scheme for mobile cloud computing environment. J Supercomput 68(2):624–651
134. Wang C, Huaizhi Y (2010) Study of cloud computing security based on private face recognition. In: Proceedings of the International Conference on Computational Intelligence and Software Engineering (CiSE). Wuhan, China, pp 1–5
135. Derawi MO, Yang B, Busch C (2011) Fingerprint recognition with embedded cameras on mobile phones. In: Prasad R (ed) Security and Privacy in Mobile Information and Communication Systems. Springer, Berlin, pp 136–147
136. Omri F, Foufou S, Hamila R, Jarraya M (2013) Cloud-based mobile system for biometrics authentication. In: Proceedings of the 13th IEEE International Conference on ITS Telecommunications (ITST). Tampere, Finland, pp 325–330
137. Pawle A, Pawar P (2013) Face recognition system (FRS) on cloud computing for user authentication. Int J Soft Comput Eng 3(4):189–192
138. Rassan IA, AlShaher H (2014) Securing mobile cloud computing using biometric authentication (SMCBA). In: Proceedings of the 2014 International Conference on Computational Science and Computational Intelligence (CSCI'14). IEEE, Las Vegas, pp 157–161
139. Bommagani AS, Valenti MC, Ross A (2014) A framework for secure cloud-empowered mobile biometrics. In: Proceedings of the 2014 Military Communications Conference (MILCOM). IEEE, Baltimore, pp 255–261
140. Viola P, Jones M (2001) Rapid object detection using a boosted cascade of simple features. In: Proceedings of the 2001 IEEE Conference on Computer Vision and Pattern Recognition, Kauai, Hawaii, USA, pp I-511–I-518
141. Ahonen T, Hadid A, Pietikainen M (2006) Face description with local binary patterns: application to face recognition. IEEE Trans Pattern Anal Mach Intell 28(12):2037–2041
142. Fan CI, Lin YH, Hsu RH (2006) Remote password authentication scheme with smart cards and biometrics. In: Proceedings of 49th Annual IEEE Global Telecommunications Conference (GLOBECOM). California, USA, San Francisco, pp 1–5

143. Khan MK, Kumari S, Gupta MK (2014) More efficient key-hash based fingerprint remote authentication scheme using mobile device. Computing 96(9):793–816
144. Wu F, Xu L, Kumari S, Li X (2015) A novel and provably secure biometrics-based three-factor remote authentication scheme for mobile client-server networks. Comput Electr Eng 45(7):274–285
145. Jiang Q et al (2016) A privacy preserving three-factor authentication protocol for e-Health clouds. J Supercomput, pp 1–24. doi:10.1007/s11227-015-1610-x
146. Wen J, Zhang M, Li X (2005) The study on the application of BAN logic in formal analysis of authentication protocols. In: Proceedings of the 7th International Conference on Electronic Commerce. Xi'an, China, pp 744–747
147. Xi K et al (2011) A fingerprint based bio-cryptographic security protocol designed for client/server authentication in mobile computing environment. Secur Commun Netw 4(5):487–499
148. VeriFinger SDK. http://www.neurotechnology.com/verifinger.html. Accessed 15 Dec 2015
149. Wang Y, Hu J, Phillips D (2007) A fingerprint orientation model based on 2D Fourier expansion (FOMFE) and its application to singular point detection and fingerprint indexing. IEEE Trans Pattern Anal Mach Intell 29(4):573–585
150. Wang YA, Hu J (2011) Global ridge orientation modeling for partial fingerprint identification. IEEE Trans Pattern Anal Mach Intell 33(1):72–87
151. Wang P, Ku CC Wang TC (2011) A new fingerprint authentication scheme based on secret-splitting for cloud computing security. Recent Application in Biometrics. InTech Open Access Publisher, Europe, pp 183–196
152. Bhatia T, Verma AK (2014) Biometric authentication for cloud computing. In: Deka GS (ed) Handbook of Research on Securing Cloud-Based Databases with Biometric Applications. IGI Global, pp 209–235
153. Chen CL, Lee CC, Hsu CY (2012) Mobile device integration of a fingerprint biometric remote authentication scheme. Int J Commun Syst 25(5):585–597
154. Cheng H et al (2012) Identity based encryption and biometric authentication scheme for secure data access in cloud computing. Chin J Electron 21(2):254–259
155. Shor PW (1999) Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM Rev 41(2):303–332
156. Bernstein DJ, Buchmann J, Dahmen E (2009) Post-quantum cryptography. Springer Science & Business Media, Berlin