SECURE DATA STORAGE MECHANISM FOR INTEGRATION OF WIRELESS SENSOR NETWORKS AND MOBILE CLOUD

Chengwei Hu
Guangzhou Civil Aviation College, China
31436138@qq.com

Submitted: Mar 23, 2016 ***Accepted: July 31, 2016*** ***Published: Sep. 1, 2016***

Abstract-Together with an explosive growth of the mobile applications and emerging of cloud computing concept, mobile cloud computing (MCC) has been introduced to be a potential technology for mobile services. Wireless Sensor Networks (WSN) is the technology that connects the virtual world and the physical world where nodes can autonomously communicate among each other and with intelligent systems. This paper describes the concept of wireless sensor networks and mobile cloud computing. Recently, much research has proposed to integrate wireless sensor networks (WSNs) with mobile cloud computing, so that powerful cloud computing can be exploited to process the sensory data accumulated by WSNs and provide these date to the mobile users on demand. The current WSN-MCC integration schemes have several drawbacks. This paper proposes a data processing framework, which aims at transmitting desired data to the mobile users in a rapid, reliable and even more secure manner. The proposed framework decreases the storage requirements for sensor nodes and networks gateway. And it minimizes the traffic overhead and bandwidth requirement for sensor networks. Additionally, the framework can predict the future trend of sensory data and provide security for this sensory data. This framework ensures the mobile users obtain their desired data faster.

Index terms*: Mobile cloud computing; Wireless Sensor Networks (WSN); Cloud Architectures; Secure Data Storage; framework; integration*

I. INTRODUCTION

Data gathering capability of wireless sensor networks (WSNs) as well as the data storage and processing ability of mobile cloud computing (MCC), WSN-MCC integration is attracting significant attention from both academia and industry. Focusing on processing of the sensory data in WSN-MCC integration, by identifying the critical issues concerning WSN-MCC integration and proposing a sensory data processing framework, which aims at transmitting desirable sensory data to the mobile users in a fast, reliable, and secure manner.

II. MOBILE CLOUD COMPUTING

Today, Mobile devices (e.g., Smartphone, Tablet Pcs, etc) are densely used in today's scenario and still get even more important since the usage of mobile Internet. The growth of the number of applications available for those devices in the last few years has shown that there is a high demand for mobile apps. With the emergence of Cloud computing in mobile web, mobile users can use infrastructure, platform, software provided by cloud providers on on-demand basis. Emergence of Cloud Computing with mobile devices gave birth to Mobile Cloud Computing.

a. Cloud computing

Cloud computing is a novel way to provide customers with Information Technology services, but with virtualization technologies in the background. Cloud computing uses networked infrastructure; software and computing power to provide resources to customers in an on-demand environment. With cloud computing, information is stored remotely in a centralized server farm and is accessed by the hardware or software thin clients that can include desktop computers, notebooks, handhelds and other devices. Typically, Clouds utilize a set of virtualized computers that enable users to start and stop servers or use compute cycles only when needed (also referred to as utility computing) [1].

Cloud computing (CC) gives its users the possibility to host and deliver services over the

Internet by dynamically providing computing resources. Cloud computing eliminates the requirement for users to plan ahead for acquiring different resources, such as storage and computing power, and therefore, is attractive to business owners. Moreover, enterprises can provide resources depending on service demand. In particular, resources can be dynamically added and released depending on service demand and with minimal management effort. [2]

b. Mobile Cloud Computing

Mobile Cloud Computing is a service that allows resource constrained mobile users to adaptively adjust processing and storage capabilities by transparently partitioning and offloading the computationally intensive and storage demanding jobs on traditional cloud resources by providing ubiquitous wireless access [2].Mobile cloud applications move the computing power and data storage away from mobile phones and into the cloud, bringing applications and mobile computing to not just Smartphone users but a much broader range of mobile subscribers".

However research still needs to be done in order to solve several open issues like resource discovery, session connectivity, Data delivery, Task division, better service as well as possible frameworks to support cloud computing on mobile devices. The mobile devices do not need a powerful configuration (e.g., CPU speed capacity) because all the complicated computing modules can be processed in the clouds. [3]There are many limitations in mobile devices like limited processing power, low storage, less security, unpredictable Internet connectivity, and less energy. To augment the capability, capacity and battery time of the mobile devices, computationally intensive and storage demanding jobs should be moved to cloud.

c. Mobile Cloud Computing (MCC) architecture.

From the concept of MCC, the general architecture of MCC can be shown in Fig. 1. In Fig. 1, mobile devices are connected to the mobile networks via base stations (e.g., base transceiver station (BTS), access point, or satellite) that establish and control the connections (air links) and functional interfaces between the networks and mobile devices. Mobile users' requests and information (e.g., ID and location) are transmitted to the central processors that are

connected to servers providing mobile network services. Here,mobile network operators can provide services to mobile users as AAA (for authentication, authorization, and accounting) based on the home agent (HA) and subscribers' data stored in databases. After that,the subscribers' requests are delivered to a cloud through the Internet. In the cloud, cloud controllers process the requests to provide mobile users with the corresponding cloud services. These services are developed with the concepts of utility computing, virtualization, and service-oriented architecture (e.g.,web, application, and database servers). [5]

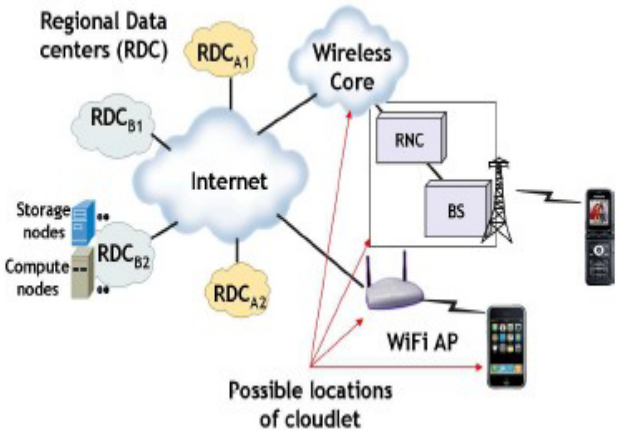

Fig. 1 Mobile Cloud Computing (MCC) architecture.

Such cloud computing is suitable and popular for small startups and medium-sized businesses, since the management of servers and many basic application services can be outsourced to the cloud. Its suitability for large organizations is still being proven in the marketplace, as each large company must investigate the price/performance tradeoff between building and managing their own private cloud or contracting out those services to a third party cloud as traffic scales to high volumes. A key consideration that factors into this decision is whether an organization wishes to store its private or proprietary data on a third party's cloud, and to what extent that cloud provider provides protection to ensure the privacy of such data. We envision that the future of cloud computing will be heterogeneous, and include many diverse clouds with different capabilities and protections, offered by different vendors. A large company that builds its private cloud may still bridge into a larger public cloud for some of its services. [7]The diverse application-level services embedded within these various clouds will likely be merged in a seamless manner via interoperable standards based on Web services that

span these heterogeneous clouds.

Today's mobile applications have already begun to adapt to cloud computing. A common theme emerging from the large wave of mobile applications developed for smartphones such as the iPhone and Android is that these mobile applications are often linked to server instances operating in the cloud. However, there is much duplication of effort, as these server instances reimplement many of the same elements of mobile support, such as location awareness, adaptation to mobility, and computational partitioning of execution between the mobile and the cloud.

d. Security Services in Mobile Cloud Computing

To improve security for cloud computing, two basic security services are provided, namely, NS and CS services. NS service only uses basic security approaches such as authentication to validate the users, and it usually involves low-complexity computing and access control tasks. CS service provides more security services such as confidentiality, digital signature, access control, audition, anti-virus scanning, etc. To simplify the notations, we denote NS and CS services as l and h, respectively. In our model, the cloud resources are divided into K portions, and each portion represents a VI.

In the cloud, mobile users can choose the desired security services l or h, which occupies αl VIs and αh VIs, $(0 < \alpha l + \alpha h < K)$, respectively. With the limitation of cloud resources (i.e., VIs), it is critical to allocate the resources to maximize the system reward, i.e., leverage the cloud service incomes and system running expenses. In other words, the cloud should decide whether to accept or reject a security service request (l or h) based on the currently available cloud resources and the arrival rate of potential future security service requests. The arrival rates of security services l and h follow the Poisson distribution with mean rates λl and λh, respectively. The cloud resource occupation time follows the exponential distribution with mean $1/\mu l$ and $1/\mu h$, respectively. In the following, we present the system states, the actions, and the reward model for the presented mobile cloud computing system.

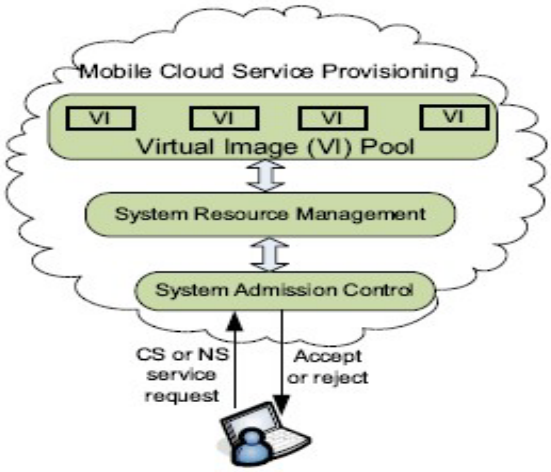

Fig. 2. Reference Model of Mobile Cloud Computing.

System States

An arrival request of security service l or h can be considered as an incoming event, and a departure of a service l or h can be considered as a leaving event. Thus, in the system model, we define three service events: 1) The cloud receives a request of security service l from a user, denoted by el; 2) The cloud receives a request of security service h from a user, denoted by eh; and 3) The transaction of a security service completes and associated VIs are released, denoted by ef . The number of security service l and security service h being served in the cloud are denoted as Nl and Nh, respectively. Therefore, the system state can be expressed as:

$$\mathcal{S} = \{s | s = \langle \widehat{s}, e \rangle\},$$

$$\text{where } \widehat{s} = \langle N_l, N_h \rangle, e \in \{e_l, e_h, e_f\}, \text{and } 0 \leq \alpha_l N_l + \alpha_h N_h$$

Actions

In system state _s, upon receiving a service request, (e.g., el or eh), two actions can be selected by the mobile cloud: accept and reject, which are denoted by a_bs,el/eh_ = 1 and a_bs,el/eh_ = 0, respectively.When a departure occurs, the cloud releases the cloud resources and there is no action in this case. Thus, the action set is

$$a_{\langle \widehat{s}, e_f \rangle} = 0. \quad A = \{a_{\langle \widehat{s}, e \rangle} | a_{\langle \widehat{s}, e \rangle} \in \{0, 1\}\}.$$

Reward Model

The system net reward can be evaluated based on the service incomes and the running expenses:

$$x(s, a) - \tau(s, a) y(s, a),$$

where x(s, a) is the net lump sum incomes for the cloud when action a is chosen at the current state s, y(s, a) is the service holding cost rate when the cloud is in state s and action a is selected, and τ (s, a) is the expected service time from the current state s to the next state when decision a is selected. x(s, a) is computed as:

$$x(s,a) = \begin{cases} 0, & a_{\langle \widehat{s},e \rangle} = 0, \\ R_l, & a_{\langle \widehat{s},e_l \rangle} = 1, \\ R_h, & a_{\langle \widehat{s},e_h \rangle} = 1, \end{cases}$$

where Rl and Rh are an income of the cloud when an l and an h security service request is accepted, respectively. The service holding cost rate y(s, a) is proportional to the occupied cloud resources, which is given by

$$y(s,a) = \begin{cases} \alpha_l N_l + \alpha_h N_h, & a_{\langle \widehat{s},e \rangle} \models 0, \\ \alpha_l (N_l + 1) + \alpha_h N_h, & a_{\langle \widehat{s},e_l \rangle} = 1, \\ \alpha_l N_l + \alpha_h (N_h + 1), & a_{\langle \widehat{s},e_h \rangle} = 1. \end{cases}$$

In this section, we evaluate the performance of the proposed SSAM using a simulator written in matlab. We set up a cloud system with the total number of VIs from 2 to 15. The request arrival rates of services l and h are 5 and 2 per unit time, respectively, and the average service holding time of each connection is μl = μh = 6 unit times, if not otherwise specified. A service h occupies two VIs while l occupies one VI when it is accepted. Accordingly, an income of 0.3 for l and 0.6 for h are added to the cloud system. We set the discount factor α = 0.1 to assure the convergence of the reward computation.

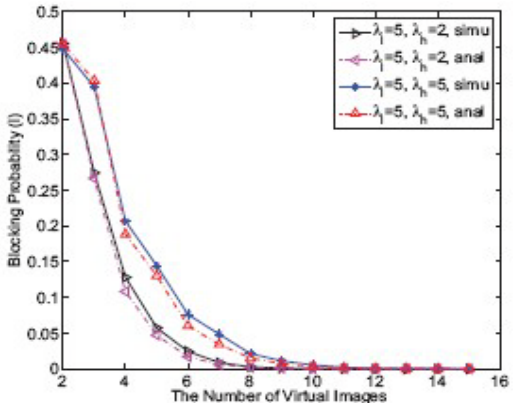

Fig. 3.Blocking probability of service l under various arrival rates

The blocking probabilities of services l and h under various arrival rates of service requests

are shown in Fig. 3 and Fig. 4, respectively. A lower blocking probability is achieved when more network resources, e.g., VIs, are available. Because service h requires two times cloud resources than service l, h is more likely to be rejected, especially when the cloud resource is limited, e.g., only two VIs in the cloud. Therefore, the blocking probability of service h is larger than that of service l accordingly. We further increase the arrival rate of service h from 2 to 5 per unit time. It can be seen that the blocking probability increases with the traffic arrival rates for a given the network resource.

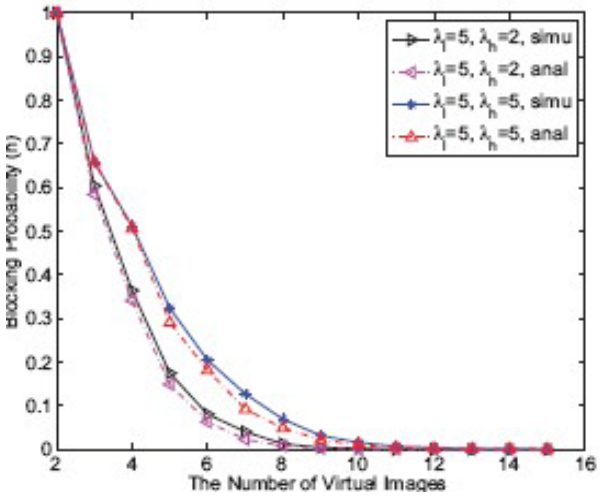

Fig. 4. Blocking probability of service h under various arrival rates

With a larger service holding time, the system cost of each mobile user increases, which results in a degraded system reward. Therefore, a new request is more likely to be rejected. The blocking probability decreases with the service occupation time for both services l and h. We derive the blocking probabilities of SSAM and conduct extensive simulations to validate our analysis. In the future, we will investigate the optimal system resources (i.e., the number of VIs) to obtain the maximal system rewards under the given blocking probability. In addition, we will incorporate more system metrics into the constructions of the reward function such as different application tasks as well.

d. Basic Mobile Cloud Computing services

We envision cloud computing providers will provide a set of basic services for mobile computing. There are three types of services. The first one is what we refer as platform

services, the second is application services, and the third is context-rich support services.

Platform services

Platform services include computing, storage, database, memcache, content distribution as shown in Figure 5. Currently all EC2 services accessible from mobile devices are considered platform services. Some of these basic services can benefit from application sharing. Take distributed memcache service for an example. Many applications may create same or access same data sets. With a shared memcache service, it will be more likely to have a cache hit due to the larger cache size. It will reduce computation demand to re-generate the cached results. Of course, sharing bring forth the issues of security, privacy as well as how much storage each application should have. Out of the basic platform service, one can already build very useful applications. For example, with storage service, and computing service, one can build file backup service, and file syncing service (keep all registered devices in sync of the user content). One can also build a data locker service [1]. In essence, the data locker protocol works with p2p protocols closely to service files on behalf of end hosts. It is particularly appealing in the mobile device context as it minimizes the usage of wireless access links. [8]

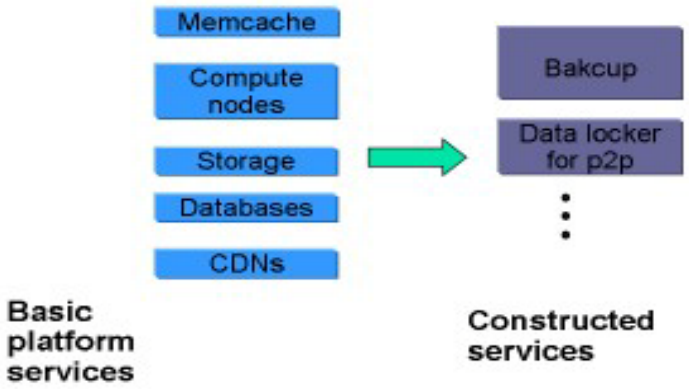

Fig. 5: Platform services

Application services

Public cloud provider can also offer a set of essential application services. For example, people may not trust each individual application and thus, may not reveal their location information. This can hamper the development of location based services. If mobile devices are using the cloud services, then there is prior trusted relationship. For example, Apple iCloud users are comfortable that their private data will be protected from un-authorized use.

So it is easier to trust the cloud provider for location privacy. Thus, a presence service can be an essential service so that any application that needs location information can talk to the presence service. The presence service will implement location privacy policies according to what are stipulated by the mobile subscribers. We recognize that different people have different level of privacy requirements. It is conceivable that some people may not want to sign up with a presence service. However, the presence service will facilitate the development of location-based services. Presence service will save resources as it is not replicated for each location based application. [9]

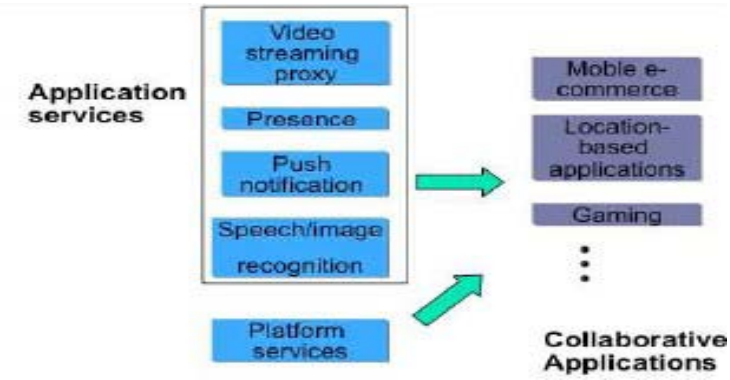

Fig. 6: Application services

Context-rich services

We envision that many mobile applications will become more personalized, and more context aware, recognizing not only the location of the user and the time of day, but also a user's identity and their personal preferences. To support these mCloud services, we believe mCloud providers need to provide a set of context-rich support services. Application developers can use these context-rich support services as building blocks to build a large class of new mCloud services. We envision several context-rich support services such as context extraction service, recommendation service, and group privacy service. Context extraction service provides data mining analysis of mobile data combined with other forms of data, such as social networking data and sensor network data, in order to extract contextual clues relevant to the user. For example, recognizing the user's activity based on mobile accelerometer and audio data is one such contextual mining service that is currently being explored [8]. The context extraction service will be a common service that relieves each context-rich application from replicating context extraction, thus saving energy and reduce computation costs of

mobiles. [11]

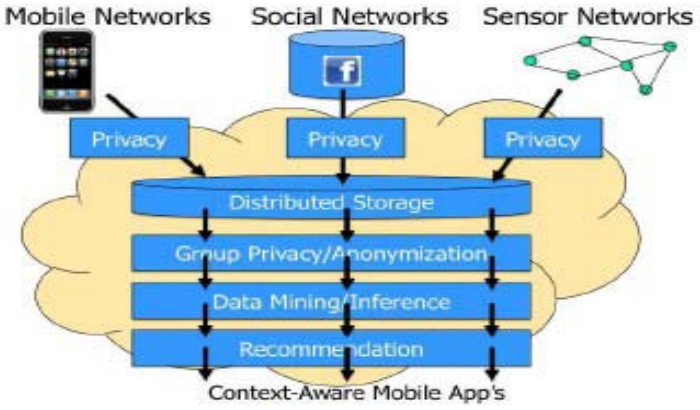

Fig. 7: Context-rich services in the Context-Aware Mobile Social Cloud.

d. Security of data/files in Mobile Cloud Computing

Mobile cloud computing is growing day by day due to the popularity of cloud computing and increasing uses of mobile devices. Many researchers are showing their interest towards this technology. There are many issues in mobile cloud computing due to many limitations of mobile devices like low battery power, limited storage spaces, bandwidth etc. Security is the main concern in mobile cloud computing.

The main issue in using mobile cloud computing is securing the data of mobile user stored on mobile cloud. The data/file of a mobile user is very sensitive; any unauthorized person can do changes in it, to harm the data. So the main concern of cloud service provider is to provide the security of data/files created and manipulated on a mobile device or cloud server. The data/file security is very essential for owner of the data/file as it can contain any confidential information of his. [12]

The data of owner is stored on the cloud server; once the data is stored the owner does not have that data on his own device. Thus, there is risk related to data security and confidentiality of the data. It is not accepted by the owner that his data/file is disclosed to someone who is not an authorized person. Before discussing why data security is needed there is a need to discuss the security threats to the data stored on the cloud. [13]There are following security risk related to data stored on the cloud server.

These attacks affect the data stored on the cloud. For owner the integrity of the data is

very important. If any unauthorized person performs changes in data of other person then it can harm the integrity of the data. Any person after finding confidential information of other person can harm that person. So, data confidentiality is also a concern of data owner. Authentication of user is also important to verify who the originator of the file is.

Table 1 Different Security Threats

Name of the Attack	**Description**
Information disclosure □	The secure information of owner is disclosed to any unauthorized user. □
Tampering □	When any unauthorized person does some changes in other user's data
Repudiation □	When a person refused after sending a message that he did not send it. □
Viruses and worms □	These are very known attacks. These are the codes whish degrade the performance of any application. □
Identity Spoofing □	In this attack a person impersonate as someone who is the owner of the data.

e. Secure Data Storage Mechanism for Mobile Cloud Computing

For the last few years Mobile Cloud Computing has been an active research field, as mobile cloud computing is in initial stage, limited surveys are available in various domain of MCC. Our main focus is on securing the data storage in mobile cloud computing. Significant efforts have been devoted in research organizations to build secure mobile cloud computing. In this paper we provided a framework for mobile devices to provide data integrity for data stored in cloud server. Incremental cryptography has a property that when this algorithm is applied to a document, it is possible to quickly update the result of the algorithm for a modified document, rather than to re-compute it from scratch. In this system design three main entities are involved:

- Mobile User (MU): Mobile user/client is a person who utilizes the storage services provided by Cloud service provider (CSP).
- Cloud Service Provider (CSP): CSP provides storage services to client. CSP is also responsible for operating, managing and allocating cloud resources efficiently.
- Trusted Third Party (TTP): TTP installs coprocessors on remote cloud; who is associated with a number of registered mobile user/client. Coprocessor provides secret key (SEK) to mobile user and is also responsible for generating message authentication code for mobile client. There are a number of operations involved in this scheme shown in Fig. 5.

Updating File on the Cloud:

Before uploading file on cloud, mobile user is required to generate an incremental Message Authentication Code (MACfile) using SEK.

$$\text{MACfile} = \sum_{k=1}^{n} \text{HMAC}(\text{Filek} , \text{SEK}).$$

Where, n is total logical partitions of file and Filek is kth part of the file. After generating MACfile, mobile client uploads the file on the cloud and stores MACfile on local storage. [15]

Inserting or deleting a block:

At any time mobile client can insert (delete) a data block in file stored on cloud server. For this client sends request to CSP, in its response CSP sends requested file to mobile client as well as to trusted coprocessor (TCO) associated with that client. TCO generates MACtco and sends it to client to match this MAC generated by TCO (MACtco) with MAC stored in client's local storage (MACfile). If these two MAC matches , the client can perform insertion/deletion in the file and again computes MACfile with help of old MACfile, SEK and inserted/deleted block. For avoiding communication overhead only updated block is uploaded on cloud server.

Integrity Verification:

At any time mobile client can verify the integrity of data stored on cloud server by sending request to cloud server, on receiving request cloud server sends file to TCO for integrity verification. TCO generates incremental authentication code and sends it to mobile client directly. Now mobile client compares this MACtco with stored MACfile to verify integrity. If

these two matches then integrity is verified.

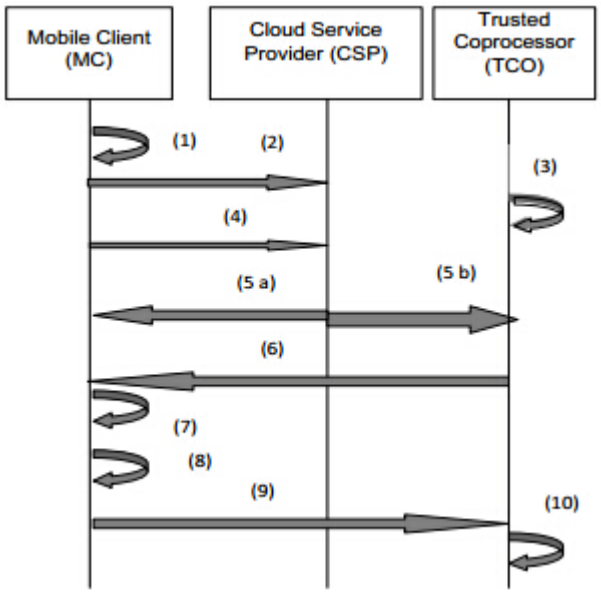

Fig 8: Communication between mobile clients, Cloud Service Provider and Trusted Coprocessor

Where,

(1): MC generate MACfile and stores MACfile in local memory

(2): MC uploads file on server

(3): CSP stores file on cloud

(4): MC sends request to CSP for performing insertion/deletion in the file

(5a): CSP sends requested file to MC

(5b): CSP forwards requested file to TCO

(6): TCO sends MACtco to MC directly

(7): MC compares MACfile and MACtco for verifying integrity

(8): MC insert/delete a block in file and computes MAC for that block

(9): MC uploads updated block on cloud

(10): CSP stores updated file.

III. WIRELESS SENSOR NETWORK

With the development of embedded system and network technology, there has been growing interest in providing fine-grained metering and controlling of living environments using low power devices. Wireless Sensor Networks (WSNs), which consist of spatially distributed self-configurable sensors, perfectly meet the requirement. The sensors provide the ability to monitor physical or environmental conditions, such as temperature, humidity, vibration, pressure, sound, motion and etc, with very low energy consumption.

The sensors also have the ability to transmit and forward sensing data to the base station. Most modern WSNs are bi-directional, enabling two-way communication, which could collect sensing data from sensors to the base station as well as disseminate commands from base station to end sensors. The development of WSNs was motivated by military applications such as battlefield surveillance; WSNs are widely used in industrial environments, residential environments and wildlife environments. Structure health monitoring, healthcare applications, home automation, and animal tracking become representative WSNs applications.

a. Wireless sensor network architecture

A typical Wireless Sensor Network (WSN) is built of several hundreds or even thousands of "sensor nodes". The topology of WSNs can vary among star network, tree network, and mesh network. [14] Each node has the ability to communication with every other node wirelessly, thus a typical sensor node has several components: a radio transceiver with an antenna which has the ability to send or receive packets, a microcontroller which could process the data and schedule relative tasks, several kinds of sensors sensing the environment data, and batteries providing energy supply. [15]

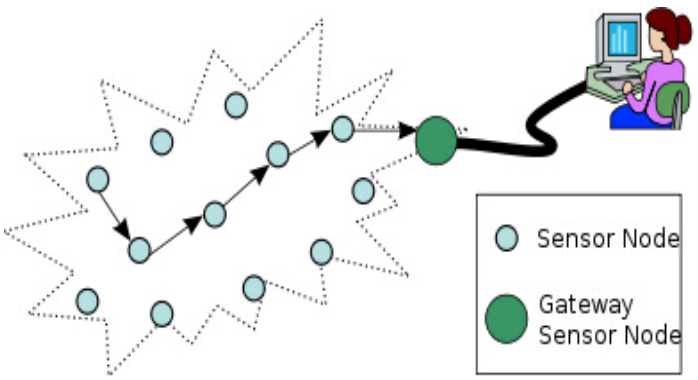

Figure 9. Typical multi-hop wireless sensor network architecture

Sensor networks are expected to play an essential role in the upcoming age of pervasive computing. Due to their constraints in computation, memory, and power resources, their susceptibility to physical capture, and use of wireless communications, security is a challenge in these networks. Current research on sensor networks is mostly built on a trusted environment. Several exciting research challenges remain before we can trust sensor networks to take over important missions [16].

b. Sensor Deployment and Coverage

In a typical sensor network application, sensors are to be placed (or deployed) so as to monitor a region or a set of points. In some applications we may be able to select the sites where sensors are placed while in others (e.g., in hostile environments) we may simply scatter (e.g., air drop) a sufficiently large number of sensors over the monitoring region with the expectation that the sensors that survive the air drop will be able to adequately monitor the target region. When site selection is possible, we use deterministic sensor deployment and when site selection isn't possible, the deployment is nondeterministic. In both cases, it often is desirable that the deployed collection of sensors be able to communicate with one another, either directly or indirectly via multihop communication. So, in addition to covering the region or set of points to be sensed, we often require the deployed collection of sensors to form a connected network. For a given placement of sensors, it is easy to check whether the collection covers the target region or point set and also whether the collection is connected. For the coverage property, we need to know the sensing range of individual sensors (we assume that a sensor can sense events that occur within a distance r, where r is the sensor's sensing range, from it) and for the connected property, we need to know the communication range, c, of a sensor. We have established the following necessary and sufficient condition for coverage to imply connectivity.

Theorem 1

When the sensor density (i.e., number of sensors per unit area) is finite, $\geq 2r$ is a necessary and sufficient condition for coverage to imply connectivity.

Theorem 2

When $c \geq 2r$, k-coverage of a convex region implies k-connectivity. Notice that k-coverage with k > 1 affords some degree of fault tolerance, we are able to monitor all points so long as no more than k − 1 sensors fail. Huang and Tseng [25] develop algorithms to verify whether a sensor deployment provides k-coverage. Other variations of the sensor deployment problem also are possible. For example, we may have no need for sensors to communicate with one another. Instead, each sensor communicates directly with a base station that is situated within the communication range. of all sensors. In another variant [23, 24], the sensors are mobile and self deploy. A collection of mobile sensors may be placed into an unknown and potentially hazardous environment. Following this initial placement, the sensors relocate so as to obtain maximum coverage of the unknown environment. They Step 1: [Achieve Coverage]

Let $\delta = (\frac{\sqrt{3}}{2} + 1)r$. Place a sensor at (i, jδ), i even and j integer as well as one at (i + r/2, jδ), I odd and j integer.

Step 2: [Achieve Connectivity]

Let $\beta = \frac{\sqrt{3}}{2}r$. Place a sensor at (0, jδ ± β), j odd

Communicate the information they gather to a base station outside of the environment being sensed. A distributed potential-field-based algorithm to self deploy mobile sensors under the stated assumptions is developed and a greedy and incremental self-deployment algorithm I developed in [23]. A virtual-force algorithm to redeploy sensors so as to maximize coverage also is developed by Zou and Chakrabarty [17]. Poduri and Sukhatme [18] develop a distributed self-deployment algorithm that is based on artificial potential fields and which maximizes coverage while ensuring that each sensor has at least k other sensors within its communication range.

c. Wireless sensor network protocol stack.

The sensor nodes are usually scattered in a sensor field. The protocol stack used by all sensor nodes is given in Fig. 10. This protocol stack combines power and routing awareness, integrates data with networking protocols, communicates power efficiently through the wireless medium, and promotes cooperative efforts of sensor nodes. The protocol stack consists of the application layer, transport layer, network layer, data link layer, physical layer, power management plane, mobility management plane, and task management plane.

Depending on the sensing tasks, different types of application software can be built and used on the application layer. The transport layer helps to maintain the flow of data if the sensor networks application requires it. The network layer takes care of routing the data supplied by the transport layer. Since the environment is noisy and sensor nodes can be mobile, the MAC protocol must be power aware and able to minimize collision with neighbors'broadcast. The physical layer addresses the needs of a simple but robust modulation, transmission and receiving techniques. In addition, the power, mobility, and task management planes monitor the power, movement, and task distribution among the sensor nodes. These planes help the sensor nodes coordinate the sensing task and lower the overall power consumption. [15]

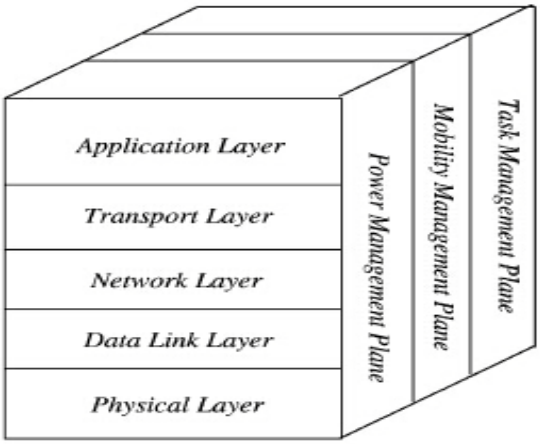

Figure 10 the sensor networks protocol stack.

The power management plane manages how a sensor node uses its power. For example, the sensor node may turn off its receiver after receiving a message from one of its neighbors. This is to avoid getting duplicated messages. Also, when the power level of the sensor node is low, the sensor node broadcasts to its neighbors that it is low in power and cannot participate in routing messages. The remaining power is reserved for sensing. The mobility management plane detects and registers the movement of sensor nodes, so a route back to the user is always maintained, and the sensor nodes can keep track of who are their neighbor sensor nodes. By knowing who the neighbor sensor nodes are, the sensor nodes can balance their power and task usage. The task management plane balances and schedules the sensing tasks given to a specific region. Not all sensor nodes in that region are required to perform the sensing task at the same time. As a result, some sensor nodes perform the task more than the others depending on their power level. These management planes are needed, so that sensor

nodes can work together in a power efficient way, route data in a mobile sensor network, and share resources between sensor nodes. Without them, each sensor node will just work individually. From the whole sensor network standpoint, it is more efficient if sensor nodes can collaborate with each other, so the lifetime of the sensor networks can be prolonged.

d. Wireless sensor network routing

Traditional routing algorithms for sensor networks are data centric in nature. Given the unattended and untethered nature of sensor networks, data centric routing must be collaborative as well as energy- conserving for individual sensors. Kannan et al. [19, 20] have developed a novel sensor-centric paradigm for network routing using game-theory. In this sensor-centric paradigm, the sensors collaborate to achieve common network-wide goals such as route reliability and path length while minimizing individual costs. The sensor-centric model can be used to define the quality of routing paths in the network (also called path weakness). Kannan et al. [20] describe inapproximability results on obtaining paths with bounded weakness along with some heuristics for obtaining strong paths. The development of efficient distributed algorithms for approximately optimal strong routing is an open issue that can be explored further.

Energy conservation is an overriding concern in the development of any routing algorithm for wireless sensor networks. This is because such networks are often located such that it is difficult, if not impossible, to replenish the energy supply of a sensor. Three forms–unicast, broadcast and multicast–of the routing problem have received significant attention in the literature. The overall objective of these algorithms is to either maximize the lifetime (earliest time at which a communication fails) or the capacity of the network (amount of data traffic carried by the network over some fixed period of time). Assume that the wireless network is represented as a weighted directed graph G that has n ver-tices/nodes and e edges. Each node of G represents a node of the wireless network. The weight w(i, j) of the directed edge (i, j) is the amount of energy needed by node i to transmit a unit message to node j.In the most common model used for power attenuation, signal power attenuates at the rate a/rd, where a is a media dependent constant, r is the distance from the signal source, and d is another constant between 2 and 4 [48]. So, for this model, $w(i, j) = w(j, i) = c * r(i, j)d$, where r(i, j) is the

Euclidean distance between nodes i and j and c is a constant. In practice, however, this nice relationship between w(i, j) and r(i, j) may not apply. This may, for example, be due to obstructions between the nodes that may cause the attenuation to be larger than predicted. Also, the transmission properties of the media may be asymmetric resulting in. $w(i,j) \neq w(j,i)$.

e. Security Architecture and requirements of Wireless sensor network

Depending on the application, a sensor network must support certain QoS (guaranteed delivery [9]) aspects such as real-time constraints (e.g., a physical event must be reported within a certain period of time), robustness (i.e., the network should remain operational even if certain well defined failures occur), tamper-resistance (i.e., the network should remain operational even when subject to deliberate attacks), eavesdropping resistance (i.e., external entities cannot eavesdrop on data traffic), and unobtrusiveness or stealth (i.e., the presence of the network must be hard to detect). These requirements may impact other dimensions of the design space such as coverage and resources [6]. Current security mechanisms in ad-hoc sensor networks do not guarantee reliable and robust network functionality. Even with these mechanisms, the sensor nodes could be made non-operational by malicious attackers or physical break-down of the infrastructure. Measurement of the network characteristics in a 'threat' of network failure is essential to understand the behavior of these networks. The security architecture (security map) of security issues in WSN is drawn as in the following figure:

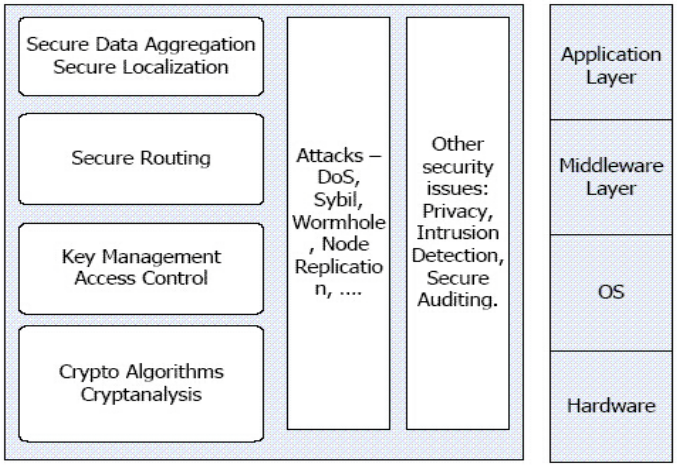

Figure 11: Security Architecture for WSN

The security requirements [9] of a wireless sensor network can be classified as follows:

Authentication:

As WSN communicates sensitive data which helps in many important decisions making. The receiver needs to ensure that the data used in any decision-making process originates from the correct source. Similarly, authentication is necessary during exchange of control information in the network.

Integrity:

Data in transit can be changed by the adversaries. Data loss or damage can even occur without the presence of a malicious node due to the harsh communication environment. Data integrity is to ensure that information is not changed in transit, either due to malicious intent or by accident.

Data Confidentiality:

Applications like surveillance of information, industrial secrets and key distribution need to rely on confidentiality. The standard approach for keeping confidentiality is through the use of encryption.

Data Freshness:

Even if confidentiality and data integrity are assured, we also need to ensure the freshness of each message. Data freshness suggests that the data is recent, and it ensures that no old messages have been replayed. To ensure that no old messages replayed a time stamp can be added to the packet.

Availability:

Sensor nodes may run out of battery power due to excess computation or communication and become unavailable. It may happen that an attacker may jam communication to make sensor(s) unavailable. The requirement of security not only affects the operation of the network, but also is highly important in maintaining the availability of the network.

Self-Organization:

A wireless sensor network believes that every sensor node is independent and flexible enough to be self-organizing and self-healing according to different hassle environments. Due to random deployment of nodes no fixed infrastructure is available for WSN network management. Distributed sensor networks must self-organize to support multihop routing.

Time Synchronization:

Most sensor network applications rely on some form of time synchronization. In order to conserve power, an individual sensor’s radio may be turned off periodically.

Secure Localization:

The sensor network often needs location information accurately and automatically. However, an attacker can easily manipulate nonsecured location information by reporting false signal strengths and replaying signals, etc.

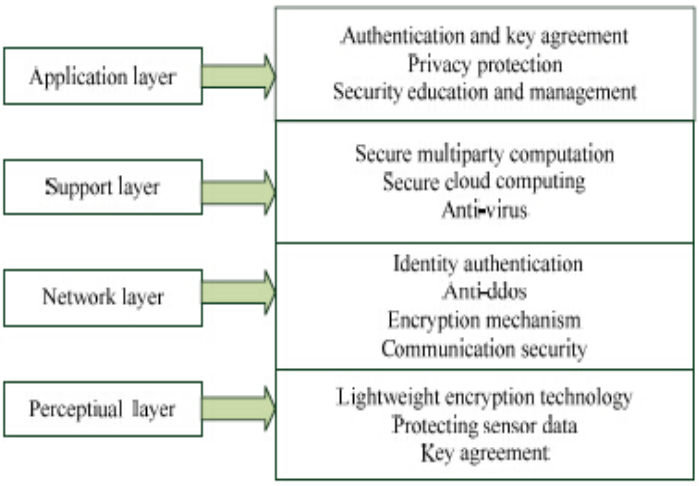

Figure 12. Security Requirements in WSNs classification

IV. INTEGRATED OF WSN AND MCC

The data gathering ability of WSN and the powerful data storage and processing capacities of MCC, the integration of WSN and MCC grabbed more attention from both academia and industry. The main idea of the WSNMCC integration is that to use the powerful sensors in the sensor networks to collect the data from the environment and these datas can be stored on the powerful servers in the CC platforms. These sensory datas are processed and then transmit those processed sensory data to the mobile users, when they are requesting. The following figure shows the WSN-MCC integration framework. In this figure WSNs gathers the weather, humidity, traffic, temperature, pressure, and house information within a certain area. The collected sensory datas are first send to the cloud for processing and storage. Then the cloud sends this data to the mobile users

when they are requested i.e., in an on demand manner.

a. Proposed WSN-MCC integration

The Fig.13 shows the proposed WSN-MCC integration, and fig. shows the flowchart of how the sensory data are processed over the framework.

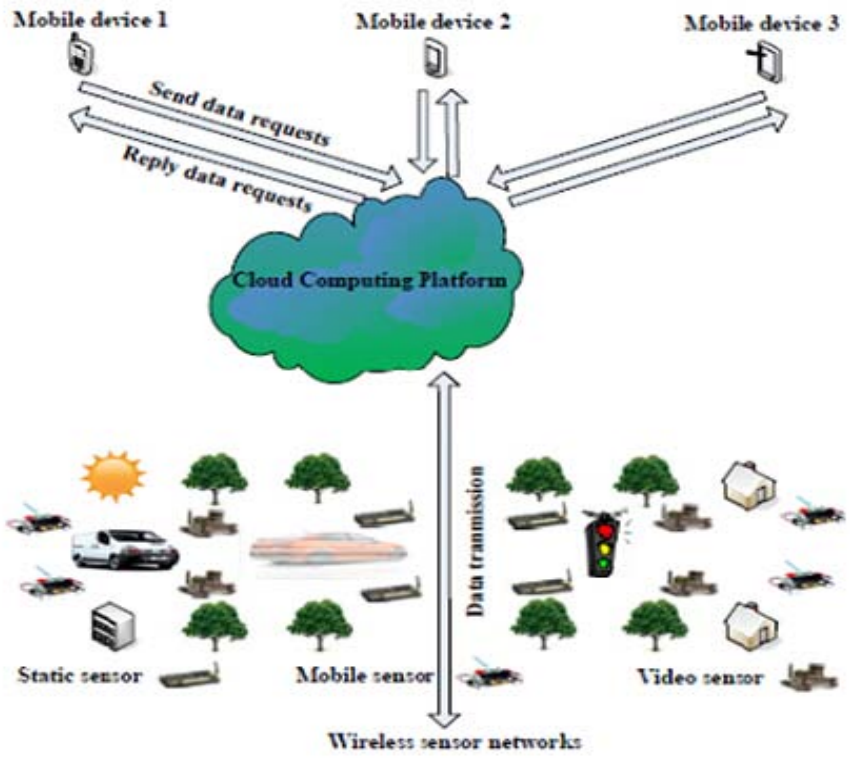

Fig 13. WSN-MCC Integration framework

The steps that are taking place in the WSN-MCC integration are given below:

First, there is a sensor gateway for each cluster of WSN collecting sensory data. The sensors in the sensor network gather the sensory data and send this sensory data to the sensor gateway. The further processing of the collected sensory data are taking place at the sensor gateway.

Second, when the sensor gateway receives the sensory data, the sensor gateway processes this data. The sensory data is processed through the following five components: data traffic monitoring unit, data filtering unit, data prediction unit, data compression unit, and data encryption unit. The unit in the sensor gateway filters the sensory data traffic according to a set of predefined rules, monitors the data traffic, and predicts the future sensory data. Then the sensory data are compressed and encrypted. Detailed descriptions of these five processing units will be given later. After the sensor gateway has processed the data, the faulty datas are discarded and the remaining normal datas are further transmitted to the cloud gateway.

Third, then the cloud gateway receives the sensory data from the sensor gateway, the cloud gateway processes the received data by decrypting the data with the data decryption unit and

then decompressing the data with the data decompression unit.

Fourth, the decrypted and decompressed sensory data from the cloud gateway are stored and processed by the powerful servers in the cloud, so that they are suitable for presentation to requesting mobile users. Also, the cloud uses the data recommendation unit to analyse the data feature information required by mobile users.

Fifth, the cloud encrypts the required sensory data with the encryption unit at the cloud gateway whenever the mobile user requesting the data. The mobile users decrypt the received data with the data decryption unit in the respective mobile device. When the mobile users issue data requests, they also encrypt the data requests, and the data requests are further decrypted by the decryption unit of the cloud gateway.

Finally, the cloud gives feedback to the WSN manager whenever the cloud obtains the data feature information required by mobile users. This feedback contains feature information; this feature information is encrypted with the encryption unit at the cloud gateway. The corresponding sensor gateway decrypts the information with the data decryption unit, and then, the WSN manager can take corresponding countermeasures (e.g., deploying more sensors to the area that mobile users are interested in).

b. Descriptions of Data Processing Tasks

The data processing tasks described above are explained in the following section.

Data Traffic Monitoring: Normally, the sensors have a set frequency (e.g., every 30 s). They collect data by using this set frequency. The size of data records are used check whether there is too much data or very few data. If there is too much data traffic which is more than or lesser than a normal acceptable threshold value for a particular time interval, then some sensors are compromised, and the network manager check whether the situation is true to avoid further harm from the compromised sensors to the network.

Data Filtering: The values of data collected by the sensors should fall within an acceptable range, according to the design of the sensors. However there is chances to occur sensory data values that are out of range due to various reasons. The data filtering unit checks whether the collected sensory data values are in a particular range. The data values that are out of range are the faulty data and they are discarded. [30]

Data Prediction: Here, we consider that time-series sensory data are collected by the WSN, and apply the secondary exponential smoothing model (SESM) for data prediction. The SESM is a widely used technique that can be applied to time-series data, either to produce smoothed data for presentation or to make forecasts.

Data Compression and Decompression: Compression and decompression are performed at the respective gateways to reduce packet losses due to network congestion. However, important sensory data could still be lost if the decompressed data do not perfectly match the data that are compressed, i.e., the compression/decompression process is lossy. To avoid this problem, we utilize lossless compression/decompression techniques. A deflate algorithm that combines Huffman coding and LZ77 is used here for lossless data compression.

Data Encryption and Decryption: Here Rivest–Shamir–Adleman (RSA) algorithm is used for security. RSA algorithm has the following characteristics: RSA is based on the factorization of large numbers, which is rather difficult to break. RSA is a publickeybased cryptographic algorithm, and thus, the security of keys is high RSA is widely used in reallife applications due to its simplicity and ease of implementation. [28]

c. Framework Characteristics

Based on the descriptions of the data processing tasks included in the proposed framework, we can see that the proposed framework has the following desirable characteristics.

Extend the Sensor Network Lifetime: By offloading data processing from the sensors to the cloud, energy consumption due to extensive data processing at the sensors will be significantly reduced, and the lifetime of the sensor network will be extended.

Reduce the Storage Requirement of the Sensor and the Sensor Gateway: In the proposed framework, complex signal processing functions are included to the cloud. There is no need for the sensor, the sensor gateway, or the cloud gateway to store a large amount of data for processing. Thus, the storage requirements of the sensor and the sensor gateway are minimized.

Decrease the Sensory Data Transmission Bandwidth Requirement and Traffic: Because the sensory data are filtered and compressed before transmitting to the cloud, the traffic load and transmission bandwidth requirements for sensory data are reduced.

Predict the Future Trend of the Sensory Data: We can predict the future trend of the sensory data by using the SESM method. Since peoples are aware about the future conditions, peoples can take measures in advance to prevent the occurrence of dangerous events.

Monitor the Sensory Data Traffic: Based on the data traffic monitoring unit in the sensor gateway, the sensory data traffic is monitored. If the sensory data traffic is too high or too low, then there is some error occurred with some sensors. Only the true data values are accepted. Faulty data values are discarded. [29]

Improve the Security of Transmitted Data: Since the compressed data are encrypted with RSA before transmission to the cloud, there will not be any hacking.

V. CONCLUSIONS

The integration of WSN with MCC is a very important research topic. Focusing on the sensory data processing aspect in integrated WSN–MCC, in this paper, we have proposed a framework to process the sensory data collected by the sensors, before transmitting the sensory data to mobile users in a fast, reliable, and secure manner. This framework includes data traffic monitoring, filtering, prediction, compression, and decompression capabilities are incorporated in the sensor gateway and the cloud gateway. Data encryption and decryption techniques are applied in the cloud, mobile devices, and sensor and cloud gateways to increase capacity. Due to the advanced capabilities and high performance of the proposed framework the mobile users can securely obtain their desired sensory data fast

REFERENCES

[1] D.-Y. Chen and J.-T. Tsai, “Resource-limited intelligent photo management on mobile platforms,” in Machine Learning and Cybernetics (ICMLC), 2011 International Conference on, Jul 2011, pp. 627–630.

[2] P. Angin, B. Bhargava, and S. Helal, “A mobile-cloud collaborative traffic lights detector for blind navigation,” in Proceedings of the 2010 Eleventh International Conference on

Mobile Data Management, ser.MDM ’10. Washington, DC, USA: IEEE Computer Society, 2010, pp. 396–401.

[3] G. Yu, H. Song, and J. Gao, "Unmanned Nmanned Aerial Vehicle Path Planning Based On Tlbo Algorithm," International Journal on Smart Sensing & Intelligent Systems, Vol. 7, pp. 1310-1325, 2014.

[4] ZH. Yu, L. Feng, H. Jie, “Amn-Pso Method For Jamming Unmanned Aerial Vehicle Network”, International Journal on Smart Sensing & Intelligent Systems, Vol. 8, No. 4, pp. 2042-2064, 2015.

[5] M. Goharimanesh, A. Akbari, “Optimum parameters of nonlinear integrator using design of experiments based on Taguchi method”, Journal of Computational Applied Mechanics, Vol. 46, No. 2, pp.233-241, 2015.

[6] M. Alvarado, F. Gonzalez, A. Fletcher, and A. Doshi, "Towards The Development Of A Low Cost Airborne Sensing System To Monitor Dust Particles After Blasting At OpenPit Mine Sites," in IEEE Sensors, Vol. 15, pp. 19667-19687, Busan, South Korea, Nov. 01-04, 2015.

[7] Y. Peng, W. Guo, M. Liu, and S. Xie, "Active Modeling Based Yaw Control of Unmanned Rotorcraft," International Journal on Smart Sensing & Intelligent Systems, Vol. 7, pp. 380-399, 2014.

[8] Y. LI, C. Chen, and W. Chen, "Research On Longitudinal Control Algorithm For Flying Wing UAV Based On LQR Technology," International Journal on Smart Sensing and Intelligent Systems, Vol. 6, pp. 2155-2181, 2013

[9] Y. Takabe, K. Matsumoto, M. Yamagiwa, and M. Uehara, “Proposed sensor network for living environments using cloud computing,” inProc. 15th Int. Conf. Netw.-Based Inf. Syst., 2012, pp. 838–843.

[10] G. Fortino, M. Pathan, and G. D. Fatta, “Bodycloud: Integration of cloud computing and body sensor networks,” in Proc. IEEE 4th Int. Conf. Cloud Comput. Technol. Sci., 2012, pp. 851–856.

[11] B.-G. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti, “Clonecloud: elastic execution between mobile device and cloud,” in Proceedings of the sixth conference on Computer systems, ser. EuroSys ’11. New York, NY, USA: ACM, 2011, pp. 301–314.

[12] B.-G. Chun and P. Maniatis, “Augmented smartphone applications through clone cloud execution,” in Proceedings of the 12th conference on Hot topics in operating systems, ser. HotOS’09. Berkeley, CA, USA: USENIX Association, 2009, pp. 8–8.

[13] X. H. Li, H. Zhang, and Y. F. Zhang, “Deploying Mobile Computation in Cloud Service,” in Proceedings of the First International Conference for Cloud Computing (CloudCom), 2009, p. 301.

[15] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, “Enabling public verifiability and data dynamics for storage security in cloud computing,”in European Symposium on Research in Computer Security (ESORICS) 2009, Saint Malo, France, Sep 2009.

[16] S. Zhu, S. Setia, and S. Jajodia., LEAP: Efficient Security Mechanisms for Large-Scale Distributed Sensor Networks., CCS'03, October 2010

[17] Y. Zou and K. Chakrabarty, Sensor deployment and target localization in distributed sensor net-works, ACM Transactions on Embedded Computing Systems, 3, 1, 2004, 61-91.

[18] S. Poduri and G. Sukhatme, Constrained coverage for mobile sensor networks, IEEE Intl. Conf. on Robotics and Automation (ICRA’04), 2004, 165-171.

[19] R. Kannan, S.Sarangi, S.S. Iyengar and L. Ray, Sensor-centric quality of routing in sensor networks,INFOCOM, 2003.

[20] R. Kannan, S. Sarangi, S. Ray and S. Iyengar, Minimal sensor integrity: Computing the vulnerability of sensor grids, Info. Proc. Letters, 86, 1, 2003, 49-55.

[21] R. Kannan and S. S. Iyengar, Game-theoretic models for reliable, path-length and energy-constrained routing in wireless sensor networks, IEEE Journal on Selected Areas in Communications, 2004

[22] M. Yuriyama and T. Kushida, “Sensor-cloud infrastructure—Physical sensor management with virtualized sensors on cloud computing,” in Proc. 13th Int. Conf. Netw. Based Inf. Syst., 2010, pp. 1–8.

[23] S. Slijepcevic and M. Potkonjak, Power efficient optimization of wireless sensor networks, IEEE Intl. Conf. on Communications, 2011.

[24] A. Spyropoulos and C. Raghavendra, Energy efficient communications in ad hoc networks using directional antenna, IEEE INFOCOM, 2012.

[25] I. Stojmenovic, and Xu Lin, Power-aware localized routing in wireless networks, IEEE Transactions on Parallel and Distributed Systems, 2010.

[26] R. Szewczyk, E. Osterweil, J. Polastre, M. Hamilton, A. Mainwaring and D. Estrin, Habitat moni-toring with sensor networks, CACM, 47, 6, 2014, 34-40.

[27] Q. Zhang, L. Cheng, and R. Boutaba, “Cloud computing: State-of-theart and research challenges,” J. Ineternet Serv. Appl., vol. 1, no. 1, pp. 7–18, May 2010.

[28] H. T. Dinh, C. Lee, D. Niyato, and P. Wang, “A survey of mobile cloud computing: Architecture, applications, and approaches,” Wireless Commun. Mobile Comput., vol. 13, no. 18, pp. 1587–1611, Dec. 2013.

[29] C. Zhu, V. C. M. Leung, X. Hu, L. Shu, and L. T. Yang, “A review of key issues that concern the feasibility of mobile cloud computing,” in Proc. IEEE Int. Conf. Cyber, Phys. Soc. Comput., 2013, pp. 769– 776.

[30] S. Wang and S. Dey, “Adaptive mobile cloud computing to enable rich mobile multimedia applications,” IEEE Trans. Multimedia, vol. 15, no. 4, pp. 870–883, Jun. 2013.