Towards an Extensible and Secure Cloud Architecture Model for Sensor Information System

Pengfei You and Zhen Huang

National Laboratory for Parallel and Distributed Processing, College of Computer, National University of Defense Technology, Changsha, Hunan 410073, China

Correspondence should be addressed to Pengfei You; hbypf@outlook.com

Received 6 June 2013; Accepted 20 July 2013

Academic Editor: Tai-hoon Kim

Copyright © 2013 P. You and Z. Huang. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Sensor information system is a specific distributed information management system for applying sensor data and aims to effectively process, manage, and analyze data emanating from sensor networks. Recently, with the development of sensor networks, sensor information system encounters many challenges, such as huge and diverse data, heterogeneous clients, scalability, and security. In this paper, we propose an extensible and secure cloud architecture model for sensor information system. Firstly, we describe the composition and mechanism of the architecture model using cloud paradigm. Secondly, we design the security solution for accessing sensor data and information services inside the architecture. This security solution ensures legal access and use for sensor data and information services and avoids illegal breach for user data in the cloud environment. In particular, a certificate authority (CA) based Kerberos protocol is proposed to provide strong identity authentication. At last, we introduce the experiment deployment for the architecture. This architecture fully utilizes the virtualization technology and cloud service functions and builds an extensible cloud platform environment supplying scalable, flexible, reliable, and secure sensor information services, which aims to solve the challenges encountered by the present sensor information system. The simulation experiment results show that the architecture gets high performance and stability throughout, while keeping scalability and flexibility brought by cloud.

1. Introduction

Sensor information system aims to effectively process, manage, and analyze data emanating from sensor networks, which is a specific distributed information management system including sensor data management and sensor information management. Sensor data management is to manage and process data stream from the sensors, and sensor information management is to organize, analysis, mine, and apply sensor data [1].

With increment of sensor amount and type within sensor networks, and expansion of coverage area of sensor networks, scale of sensor information system is more and more huge. With diversity of sensor data source, data process manner is more and more complex. It needs more and more IT resources such as mainframe and high bandwidth communication networks to process, manage, and store sensor data. This situation brings challenges to sensor information system, including security management of data, transparent data process model, cost-effective resource deployment, and dynamic storage allocation.

Cloud computing supplies a feasible solution for these challenges. Cloud computing is a model for enabling on-demand access to a shared pool of IT resources that can be elastically used with low cost [2]. In this model, the capabilities of applications are exposed as services which provide dynamically scalable and often virtualized resources in web application as a service [3, 4]. Cloud computing brings flexibility, scalability, cost controlling, and on-demand provision to applications. These features supply measures to solve the challenges encountered by present sensor information system.

The main objective of this paper is to realize a general solution to effectively process sensor data and manage sensor information for various types of sensor networks which may be large or small, homogeneous or heterogeneous, and provide various sensor information services to heterogeneous clients in a secure mode. More specifically, we focus on...
building an extensible and secure sensor information system. In [5], we presented an early version of this work where we exploited system architecture based on cloud paradigm. In this paper, we go one step forward and we demonstrate a reliable security framework oriented to that cloud sensor information system. The security framework concentrates on avoiding sensor data from breach and supplying secure and reliable sensor information services to legal users in cloud environment. In particular, a CA based Kerberos protocol is proposed to provide strong identity authentication in the security framework. Additionally, we deploy and test the prototype of the cloud based sensor information system by simulation experiment. The experiment result shows that the proposed sensor information system is effective in terms of performance and gets satisfying flexibility, scalability, and access heterogeneity in cloud environment.

The rest of the paper is organized as follows. In Section 2, we introduce the current challenges encountered by sensor information system, including system application challenges and security challenges. In Section 3, we describe the service model and core technologies of cloud computing. In Section 4, we introduce the cloud architecture model of sensor information system. In Section 5, we present a security framework oriented to the cloud based sensor information system. In Section 6, we introduce the prototype deployment and simulation experiment for the cloud sensor information system. Finally, in Section 7, we conclude this paper.

2. Current Challenges of Sensor Information System

Presently, sensor networks develop fast. There are various types of sensor networks where sensor amount is huge, bandwidth is high, coverage area is large, process task is complex, and sensing data is heterogeneous, such as multimedia sensor network [6–9]. This situation brings challenges to traditional sensor information system [10–12].

2.1. System Application Challenges. Firstly, sensor information system lacks enough flexibility for processing sensor information. The sensor data type may be different or heterogeneous, which demands diverse process manners. If the data needs to be fused, the process would be more complex. When a new data type is coming, the system can not process it promptly [6].

Secondly, sensor information system lacks effective load balancing management. For example, when applications for sensor information process reach their peak points, they still own many idle resources for a long time. At the same time, other applications such as information analyses and information storage need more resources which are held by information process applications. This situation extremely brings down resource utilization rate and performance for the system [7].

Thirdly, sensor information system lacks enough scalability for IT resource. On the one hand, many servers are deployed for specific applications, and most of the time they own surplus computing and storage power. On the other hand, when workloads for applications are too high, the system has to add new resources to satisfy it, which leads to extra costs [8].

In addition, sensor information system lacks heterogeneous client support. With the exponential increment and utilization, sensor data is needed to supply to various types of users. It demands that sensor data be accessed from various clients, such as PC, smart phone, and tablet [13, 14].

Cloud computing is a general computing paradigm, which has been applied in many fields [2, 15]. Cloud computing consists of many enabling technologies such as virtualization technology, Web service and SOA, web 2.0, worldwide distributed storage system, and map-reduce programming model, which could solve the above challenging issues existing in traditional sensor information system [16, 17].

Virtualization technology is the base of cloud computing, which partitions hardware and thus provides flexible and scalable computing platforms. It supplies virtualized IT infrastructures and sensor networks access for users on demand, which could solve scalability and load balancing of resources utilization for sensor information system.

The MapReduce programming model has been used for processing and generating large data sets across the Google worldwide infrastructures. It firstly involves applying a “map” operation to some data records—a set of key/value pairs, and then processes a “reduce” operation to all the values that shared the same key. This model could also process heterogeneous data from sensor information system flexibly and efficiently.

Computing cloud services are normally exposed as Web services. The services organization and orchestration inside Clouds could be managed in a service oriented architecture (SOA). A set of cloud services furthermore could be used in a SOA application environment, thus making them available on various distributed platforms and could be further accessed across the Internet. These features make services for sensor information system be supplied to heterogeneous clients, such as PC and smart phone.

2.2. Security Challenges. With open usage for huge sensor data and rapid demand for sensor information services, sensor data is more likely to be exposed, modified, and broken, especially in cloud environment. Based on open cloud platform, sensor information system would face security issues as follows [18].

(i) The environment in which sensors work can be compromised by the adversary. For example, the adversary can artificially reduce or raise temperatures to cause the sensors to collect improper data.

(ii) Individual sensors can be vulnerable to attacks. This can happen when the adversary has physical access to the sensors or remote access through propagating malware.

(iii) Information flows within the cloud can be intercepted and stolen or modified by compromised cloud nodes.
(iv) The cloud client can be infected by malicious code implanted by an adversary, which can lead to further security breaches within a sensor-cloud system.

(v) The communication channels between the sensors and the cloud and between the client and the cloud are vulnerable to different types of attacks. Even when the data transferred over the channels is fully encrypted, side-channel information leaks constitute creditable threats.

3. Overview of Cloud Computing

Although there is still no unified definition for cloud computing, a widely accepted one is defined by [19] as follows: “cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort or service provider interaction.” With pools of computing power, memory, storage resources, and networks, the cloud provides IT resources to users on-demand in form of services. These services are categorized into three classes as follows.

(i) Software as a service (SaaS): in SaaS, the cloud vendor delivers their application product as a service over the net on a hosting environment, which can be accessed from various front-end clients. The application solutions may include anything form database processing to web based email and network inventory control. There are many service examples such as eHealth, industry applications, and business process [3].

(ii) Platform as a service (PaaS): in PaaS, the platform vendor provides a development environment or platform with tools, on which the consumer can create their cloud services and applications. In this model, many examples of services are provided, such as Java runtime, database, development tool, Web 2.0 application runtime, and middleware. Users need not to control underlying infrastructure but only to control the deployment and configuration of applications [2].

(iii) Infrastructure as a service (IaaS): in IaaS, IT infrastructures, such as processing, networks, storage, and other fundamental computing resources, are delivered as a service to the consumer. In this sense, consumer can deploy and run arbitrary operating systems and applications but need not to manage and control underlying infrastructure [20].

There are five essential characteristics of cloud computing: (1) on-demand services, (2) ubiquitous network access, (3) resource pooling, (4) rapid Elasticity, and (5) measured services. Virtualization, as one core technology of cloud computing, is used to meet the characteristics of cloud services. Virtualization is a traditional software technology. It uses a physical resource such as a server and divides it into some virtual resources called virtual machines (VMs) [21], through which virtualization breaks down the physical barriers inherent in isolated resources and automates the management of these resources as a single entity. Via virtualization, cloud computing allocates resources in a scalable manner, which fully raises utilization of resources, including CPU and memory; hence, it is suitable for mass data process [22, 23].

Similarly, sensors and sensor networks can also be virtualized. Sensor virtualization technology enables one physical sensor to be projected as multiple virtual sensors and vice versa and enables user applications to use sensor data in a transparent manner, which enhances resource sharing and allows virtual sensors scale up or down as needed [24–26]. Besides, sensor virtualization abstracts different platforms of the physical devices, greatly enhancing satisfaction and benefiting users. It significantly expands boundary for service capability for sensor networks. In addition, via cloud computing, multiple sensing tasks for different types of sensor networks can be performed simultaneously, which greatly enhances usability of the devices and sensor networks.

Cloud computing is an extensible architecture model for IT area, combining with which, a sensor information system could become effective, scalable, flexible, and reliable.

4. A Cloud Based Architecture Model of Sensor Information System

The proposed architecture model aims to build a general platform of sensor information system based on cloud computing, which is scalable, flexible, access secure, cost-effective, and of high performance. It can effectively operate and process sensor data, manage, analyze, and mine sensor information for various types of sensor networks and supplies information services to heterogeneous clients.

4.1. Basic Composition of the Architecture Model. The architecture model applies virtualization and cloud service technologies. It consists of user interface layer and five application service layers as shown in Figure 1.

(1) User interface layer: it is the interface for users to enter the system, including web browser and client application.

(2) Security management layer: it aims to realize a sort of reliable solution to prevent illegal users from accessing and breaching application data, provides secure sensor information services to users based on access privilege, and enables users to configure and manage security policies in cloud environment, thus ensures system security. The security solution concentrates on three aspects: identity and authentication unit, access control performer, and data encryption management. Identity and authentication unit ensures that only authorized users can access system and manages user’s identity information. Access control performer carries out configuration and management for service access policy and ensures that users with different roles access different levels of services. Data encryption management makes encryption policies and supplies encryption service to the users, and thus
protects user privacy. This solution avoids malicious users' access to the system, which realizes management security for the system. The detailed security solution will be discussed in Section 5.

(3) **Application layer**: it is the main business application layer for the system, including sensor data management and sensor information management. Sensor data management is responsible for operating, processing, and maintaining data emanating from virtual sensor network. Sensor information management supplies sensor information service, such as organization, analysis, mining, retrieval, and visualization for sensor data. This layer presents a logical application collection of sensor information services to the users.

(4) **Service platform layer**: it consists of core components which encapsulate various functions as specific types of services. These services are managed, monitored, and provided to business applications in the higher
layer in a reliable and on-demand manner. When a resource request service from application layer is processed, provision notification component will apply for new resources from virtualization management layer.

(5) Virtualization management layer: it realizes virtualization for IT resources from the lower layer and is responsible for managing and maintaining these virtual resources. This layer enables transparent access to physical infrastructures and dynamic resources provision to the high layer.

(6) Infrastructure layer: it consists of physical IT resources, such as servers, clusters, storage equipments, computer networks, and sensor networks.

4.2. Operating Mechanism of the Proposed Architecture Model.
Operating mechanism of the proposed architecture model mainly runs among three layers, as shown in Figure 2. Application layer is the core logical business layer where users employ sensor information applications. Service platform layer is the core service provision layer, and virtualization management layer is the core virtual resources layers. The operating mechanism mainly consists of virtualization management, runtime environment for service platform, and sensor information system application.

(i) Virtualization management: by applying virtualization technology, such as deploying hypervisors [21, 27] onto each physical node, it realizes a virtual resource pool for physical resources, which can be utilized by service platform layer in a simple way. It configures and manages the virtual resource pool. When more physical equipments are needed or not needed, they only to be virtualized and put into or moved from the virtual resource pool, resulting in no influence to the higher layer. Therefore, it realizes scalability for the system. In addition, it maintains an image repository storing various types of VM (virtual machine) images which are used to create specific VMs for the system. Moreover, it manages the templates of the virtual sensor networks and provisions them on the selected virtual server.

(ii) Runtime environment for service platform: it is a runtime environment for managing, maintaining, and processing specific services for application layer. It consists of some core components. Event management is responsible for detecting and processing events for application system. Service monitor is to monitor various services in the system. If some conditions are satisfied, the services will be handled by service scheduler component. For example, when some system statuses reach some thresholds, a request or return service for resources will be scheduled and then provision notification component is invoked to claim resources from the lower layer or give back resources to the lower layer. Thus, it realizes elastic use for resources. Operation for the entire runtime environment is adjusted by load balancing component, which can promote performance for the system. In addition, data content management supplies data service from virtual sensor networks to application layer.

(iii) Sensor information system application: it is the core business application. It uses data form data service of data content management in a transparent manner. It enhances flexibility and simplifies sensor data process model.

5. Security Solution

In information system based on cloud computing, many consumers request different services, which leads to diverse threats to system security. The main threat source comes from illegal users’ breach of system function and user data. Therefore, the management of user identity and access to system resources is very important to maintain system security. In this section, we give the security model and define the service components that are ensured by our cloud architecture. Then, we detail the operation mechanism of security model. This model integrates user identity authentication, access control and data encryption, which would ensure system security.

5.1. Security Principles. Our model guarantees the following security demands.

Fine-Grained Access Control. Our solution ensures user data confidentiality and scalable fine-grained access control to sensor information services in cloud system.

Integrity and Authenticity. Our solution ensures message integrity during transfer between client and cloud system. Furthermore, it ensures data integrity in cloud environment. Also, each party authenticates the origin of each message received from the other party.

Availability. Our solution ensures availability of service for legitimate users when they need it, and it is resilient when a large groups of legitimate users’ access at the same time.

5.2. Security Model. The proposed security model aims to provide strong identity authentication and access control to cloud resources and protection to users’ private data in cloud environment, which brings strong security to the sensor information system. Figure 3 shows the framework of the security model, which is deployed in security management layer for the whole cloud system architecture. The model consists of three function parts: identity authentication unit, resource access control unit, and data encryption unit. In identity authentication unit for the security model especially, an authentication protocol is proposed, which improves Kerberos protocol [28, 29] by using certificate authority (CA) mechanism [30]. It can efficiently solve security deficiencies for regular Kerberos protocol on the premise of system availability, such as weak secret key protection and password attack.
5.2.1. Identity and Authentication Unit. Identity authentication is responsible to manage users’ identity and control users’ access to system resource securely. It mainly ensures legal users’ access to system and use for application services in following phrases.

(1) User Registration. As showed in Figure 3, when a new client wants to access the cloud system, he sends a registration request to identity authentication modular (IA) first. Both IA and user use Diffie-Hellman (DH) algorithm [31] to generate a common key. The steps are described as follows.

(1) When IA receives first request from client, a common secret key (K) is generated in both ends by using DH algorithm. In cloud system, IA gets its K by calling key generator modular which realizes HD algorithm.

(2) When gets system’s query, consumer encrypts his registration information (username and password, etc.) by using the secret key generated in step (1) and sends it to the system.

(3) Registration information is sent to certificate authority (CA) and is verified by CA based on some regulations. If not permitted, the rejection acknowledgement will be sent to the user, and the steps are over. Or else, the information is stored in identity manager modular (IM) via CA, and an acknowledgement of successful registration is sent to the consumer.

(4) After registering successfully, the user generates a public/private key pair (Kp, Kpr) and sends the following message which is encrypted by secret key:

\[ Kp \text{ and timestamp (ts) encrypted by private key of consumer (EKpr (ts))}. \]

(5) IA decrypts the message and calls CA to generate a user certificate based on user's public key Kp. Then the certificate against the username will be stored by IM. Encrypted message (EKpr (ts)) is decrypted using consumer’s Kp and ts is gotten.

(6) IA sends back a nonce (ts + 1) to the consumer encrypted by Kp. This nonce will be decrypted by consumer's private key Kpr.

(2) User Login. When a registered user wants to login to the system, he will first connect to IA (Figure 3). Both IA and user will use DH algorithm to generate a common key and then will communicate with each other as the following steps.

(1) When IA receives first request from the user, a common secret key (K) is generated in both ends by using DH algorithm.

(2) The Consumer then sends its login information (username, password) encrypted by the key K. The IA will decrypt and match user information and look for the consumer’s public key Kp stored in the IM.
(3) The system sends a nonce \((t, \text{username})\) encrypted using \(K_p\).

(4) The consumer receives and decrypts the nonce message using consumer’s private key \(K_pr\) and sends another message encrypted by \(K\). The message contains nonce \((t+1)\) and username which are encrypted using \(K_pr\).

(5) The system decrypts the message and then sends an acknowledgement containing username and nonce \((t+2)\) encrypted by \(K_p\).

When receives acknowledgement from the system, the consumer will send a particular service request to the system.

And then the Kerberos based authentication process will be started as the following section.

(3) Authentication Protocol. In cloud computing environment, security issues are more complex, which brings extreme challenge to traditional Kerberos protocol, such as replay attack, secret key leakage, and denial of service attacks. This situation demand stronger authentication mechanisms. Certificate authority (CA) signatures certificate for user’s public key, so as to distribute and validate the public key. The signature from CA makes attacker not fake and tampers the certificate. The proposed authentication protocol combines Kerberos protocol and CA, which brings strong authentication.

Figure 3: Security model for the cloud based sensor information system.
According to Figure 3, different messages are exchanged amongst IA, CA, TGS, and RAS. The authentication protocol is explained through the following steps. For simplicity in this description, all requests from client that go via IA are deemed as directly from IA. That is to say, in these steps, the role of the IA is looked as client.

1. Client sends a service request message to the CA requesting services needed by the user. This message includes user public key, username, and password.

2. The CA checks if the client is in the database for IM. If so, the CA sends back the following message to the client.

   Message A: public key certificate of the client and TGS (Cc, Ctgss) encrypted using the private key of the CA (PRKca).

3. Once the client receives message A, it decrypts message A to obtain the public key of TGS (PKtgss). Then it sends the following message to the TGS.

   Message B: Cc, Client/TGS session Key 1 (CSKtgss1) and Authenticator (composed of the client ID and the timestamp) which is encrypted by the private key of the client (PRKc). The content of message B is encrypted using PKtgss.

4. Upon receiving messages B, the TGS decrypts it using its private key (PRKtgss). This gives it the Cc, CSKtgss1, and the Authenticator. It decrypts the Authenticator using the public key of the client (PKc) in Cc to get the client’s identity and verify it using Cc. If it is verified, the TGS sends the following messages to the client.

   Message C: client/TGS session key 2 (CSKtgss2) and authentication time encrypted by CSKtgss1.

   Message D: TGT (which includes the client ID, client network address, ticket validity period, and the CSKtgss2) encrypted using the PRKtgss.

5. Upon the receipt of messages C and D, it decrypts message C to get the CSKtgss2. The session key is used for further communications with TGS. Based on the enough information, the client can authenticate itself to the TGS. However, message D cannot be decrypted, as it is encrypted using TGS’s secret key.

6. When requesting services, the client sends the following two messages to the TGS.

   Message E: composed of the TGT from message D and the ID of the requested service.

   Message F: authenticator (which is composed of the client ID and the timestamp) encrypted using the CSKtgss2.

(7) Once receiving messages E and F, the TGS retrieves message D out of message E. It decrypts message D using TGS’s secret key, thus gets the CSKtgss2. Using this key, the TGS decrypts message F (Authenticator) and sends the following two messages to the client.

   Message G: client-to-server ticket (CST, which includes the client ID, client network address, validity period, and client/server session key (CSKs)) encrypted using the server’s secret key.

   Message H: CSKs encrypted with the CSKtgss2.

(8) Once receiving messages G and H from TGS, the client can authenticate itself to the request acceptance server (RAS). The client sends the following two messages to the RAS.

   Message I: composed of the message G received from the previous step (the CST, encrypted using RAS’s secret key).

   Message J: a new authenticator (which is composed of the client ID and timestamp) encrypted using CSKs.

(9) The RAS uses its own secret key to decrypt the ticket and get the CSKs. Decrypting the authenticator using the CSKs, the RAS sends the following message to the client to confirm its true identity.

   Message K: the timestamp from client’s authenticator plus 1, encrypted using the CSKs.

(10) The client decrypts the message using the CSKs and checks if the timestamp is correctly updated. If so, then the client can trust the RAS and start sending service requests to the RAS.

The proposed authentication protocol shows stronger security as follows.

1. Usage for CA mechanism further ensures security for public key distribution and communication, and thus ensures validation for user and server. In addition, it does not slow down identity authentication obviously.

2. Based on certificate signature mechanism for CA, timestamp verification mechanism can prevent replay attack in some time slots.

3. The breach for authentication server would not lead to disclose user’s private key and session key, while it would in traditional Kerberos protocol.

5.2.2. Access Control Process. Role-based access control (RBAC) is the most promising access control model because of its simplicity, flexibility in capturing dynamic requirements, and support for the principle of least privilege and efficient privilege management [32]. We use RBAC as our access control model. When every service request for a particular resource gets to the request acceptance server, it is forwarded to RBAC performer, which then reads the access
policies written in XML [33] from access policy database and takes corresponding decision, which is forwarded to the request acceptance server. Based on this decision, the server sends ACK/NACK to the user. It should be noted that the database stores all the relevant default access policies during registration of a user, thus it also connects to IA.

5.2.3. Private Data Encryption. Private data encryption unit is responsible for privacy needs and encryption of user data. It consists of encryption policy manager (EPM) and private data encryption service (PDES). PDES is a service packet which provides most popular encryption algorithms, such as DES, TripleDES, RC5, and Blowfish [34]; EPM is responsible to setup operation module for individual encryption algorithm or combined encryption algorithms based on encryption policies written in XML. Users can directly request EPM to setup his encryption module and encryption algorithm from PDES. Private data encryption service is also a basic system service which can be called by information services in application layer if needed.

5.2.4. Maintenance Measures for Availability under Threads. Availability for the cloud based sensor information system not only exists in system performance, but also in the situation with security threats, which ensures that the system is available for the user all the time. However, malicious attacks, such as denial of service (DoS) [35] attacks, concentrate on exhausting system resources, which leads to system unavailability.

Without effective measure, the system will break down when the requests from DoS attacker are in large numbers. In cloud environment, the requests are responded, and resources are provided as services to DoS attacks, which can be coped with by load balancing functions of cloud. Based on virtualization features of cloud, we used two-layer load balancing policy.

In layer 1, scheduling strategy on load balancing of virtual machine resources [36] is used in Service platform layer, which obtains the best load balancing in scheduling VM resources and reduces dynamic migration by using a genetic algorithm.

In layer 2, server-based load balancing for Internet distributed services [37] is used among the VMs, which helps in reducing the service response times by using a protocol that limits the redirection of requests to the closest remote servers without overloading them.

On the one hand, the two-layer load balancing policy fully utilizes the system resources and provides nice performance to the upper layer; on the other hand, the requests from DoS attacker will be efficiently limited and processed. Thus, it can ensures availability for the cloud system.

6. Experimentation and Evaluation

6.1. Organization Mode of the Proposed Cloud Architecture and Experiment Deployment. The proposed cloud architecture model is organized as shown in Figure 4. Various types and localities of sensor networks are connected to cloud via corresponding broker networks, and the sensor data from sensor networks is transmitted to cloud. Meanwhile, the data is processed and stored in cloud, and the cloud supplies data services to various types of clients, such as smart phone and PC.

According to the organization mode of the proposed architecture, we have deployed the prototype system of sensor information architecture in the cloud based experiment environment. The prototype system is implemented with JDK 1.6. The physical hardware consists of 8 compute nodes, a private network using switched gigabit Ethernet. Each node has two Intel Xeon Quad-core 2.27 GHz and 4 GB of RAM memory. All nodes use the Kernel-based Virtual Machine (KVM) hypervisor for virtualization [38]. Each node installs 2 VMs running Ubuntu 8.04.

Our experiment aims to test the availability and scalability of the system prototype. It is performed through two parts: client access simulation and sensor network simulation. TOSSIM [39] is a good discrete event simulator for TinyOS [40] sensor networks. The sensor network simulation is based on TOSSIM framework, in which we have simulated 100 motes that can collect sensor data randomly and send out hex sensor data by using sensor data collection service confirming to web services standard [41]. In client access simulation, a simulation program that can stimulate different numbers of clients running simultaneously has been used for requesting different types of sensor information services (from application layer) using data collected by sensor data collection service.

6.2. Performance Evaluation. Based on the same experiment environment, we consider three metrics as performance measure: response time (viz., the time from user requesting to receiving response), CPU utilization, and task throughput (namely, the number of completed tasks per elapsed time).

Figure 5 shows the change of average response time with the number of client requests. From Figure 5, we know that the response time is moderate when the number of clients who request the observation simultaneity is small. Even when the number of clients reaches 300, it fully considers the load of all compute nodes, thus can allocate system resources more reasonably, which makes the system available.

Figure 6 shows that average CPU utilization is increased with the numbers of the client requests, but the change rate is slow. The CPU utilization is suitable and effective.

Figure 7 shows the system services throughput changing with the number of client requests. It shows the inherent scalability of the cloud system; that is, it almost gives the same throughput in terms of number of services execution per second even when the requesting clients are increased.

From the results, we realized that the cloud based sensor information system owns nice performance for processing diverse sensor data. It can effectively allocate, schedule system resources, and maintain load balancing especially when facing the huge services for data process, and also keeps moderate availability and scalability.
7. Conclusion

With fast development for sensor networks, sensor information system encounters many challenges, such as flexibility, scalability, load balancing, heterogeneous information services, and security issues.

In this paper, we have proposed a general cloud architecture model of sensor information system aiming to solve above challenges. We have described the framework and operating mechanism of the architecture model, and then presented the security solution for the proposed system architecture. On one hand, by using virtualization and cloud computing services technologies, this architecture model builds an extensible cloud platform environment for sensor information system where IT resources are virtualized and supplied to heterogeneous users on demand as sensor information services. On the other hand, the proposed security framework oriented to the cloud system realizes system security from three aspects: identity authentication, access control, and data encryption. It ensures only legal users can...
access system resources based on their access privileges and prevents malicious breach to user data and system information. In particular, we have proposed CA-based Kerberos authentication protocol. This protocol takes advantage of certificate signature mechanism for CA, which solves issues in traditional Kerberos protocol, such as weak secret key protection and password attack. It enhances security for Kerberos protocol, but does not slow down authentication, which keeps system availability. At last, we deployed prototype experiment for the cloud based sensor information. The experiment shows that the cloud information system has high efficient load balancing, availability, and scalability, which satisfies our desired goal.

However, security issues in cloud environment are very complicated. Our security solution mainly aims to data, services, and management security from users to services layer. In fact, virtualization technology and service model for cloud have many security issues, most of which will lead to fatal security disaster [35, 42–44]. At following work, we will research security risks for sensor information system probably brought by service model and virtualization for cloud computing.

Disclosure

A previous work titled “An architecture model of sensor information system based on cloud computing” has been published in 2012 Information Science and Industrial Applications Conference [5]. This paper is an extended version of that work.

Acknowledgments

This research work is supported by National Basic Research Program of China under Grant no. 2011CB302601 and National High-Tech R&D Program of China under Grant no. 2011AA01A202.

References

[1] B. Thuraisingham, “Secure sensor information management and mining,” IEEE Signal Processing Magazine, vol. 21, no. 3, pp. 14–19, 2004.
[2] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, “Cloud computing and emerging IT platforms: vision, hype, and reality for delivering computing as the 5th utility,” Future Generation Computer Systems, vol. 25, no. 6, pp. 599–616, 2009.
[3] A. M. Basal and A. L. Steenkamp, “A saas-based approach in an E-learning system,” International Journal of Information Science and Management, pp. 27–40, 2010, special issue.
[4] L. Xiao and Z. Wang, “Cloud computing: a new business paradigm for E-learning,” in Proceedings of the 3rd International Conference on Measuring Technology and Mechatronics Automation (ICMTMA’11), pp. 716–719, January 2011.
[5] P. You and Y. Peng, “An architecture model of sensor information system based on cloud computing,” in Proceedings of International Conference on Information Science and Industrial Applications (ISI ’12), pp. 145–150, 2012.
[6] A. Sharif, V. Potdar, and E. Chang, “Wireless multimedia sensor network technology: a survey,” in Proceedings of the 7th IEEE International Conference on Industrial Informatics (INDIN ’09), pp. 606–613, June 2009.
[7] E. Shakshuki, X. Xing, and H. Malik, “An introduction to wireless multimedia sensor networks,” 2009, http://www.igi-global.com/chapter/introduction-wireless-multimedia-sensor-networks/20991.
[8] Y. Charfi, N. Wakamiya, and M. Murata, “Challenging issues in visual sensor networks,” IEEE Wireless Communications, vol. 16, no. 2, pp. 44–49, 2009.
[9] I. T. Almalkawi, M. G. Zapata, J. N. al-Karaki, and J. Morillo-Pozo, “Wireless multimedia sensor networks: current trends and future directions,” Sensors, vol. 10, no. 7, pp. 6662–6717, 2010.
[10] A. Nikdel, M. Shavali Kohshoori, and S. Mahdi Jamei, “An intelligent and energy efficient area coverage protocol for wireless sensor networks,” International Journal of Grid and Distributed Computing, vol. 4, pp. 41–56, 2012.
[11] S. Li, P. Zhu, X. Liao, and Q. Fu, “Reliable data delivery in wireless sensor networks: an energy-efficient, cluster-based approach,” in Proceedings of International Conference on Next Generation Teletraffic and Wired/Wireless Advanced Networking (NEW2AN ’06), vol. 4003 of Lecture Notes in Computer Science, pp. 384–395, 2006.
[12] W. Cheng, X. Liao, C. Shen, S. Li, and S. Peng, “Reliable data delivery in wireless sensor networks: an energy-efficient, cluster-based approach,” in Proceedings of International Conference on Next Generation Teletraffic and Wired/Wireless Advanced Networking (NEW2AN ’06), vol. 4138 of Lecture Notes in Computer Science, pp. 478–489, 2006.
[13] Y. Charfi, N. Wakamiya, and M. Murata, “Challenging issues in visual sensor networks,” IEEE Wireless Communications, vol. 16, no. 2, pp. 44–49, 2009.
[14] R. Liu and I. J. Wassell, “Opportunities and challenges of wireless sensor networks using cloud services,” in ACM Workshop on Internet of Things and Service Platforms (IoTSP ’11), December 2011.
[15] J. Wang, H. Abid, S. Lee, L. Shu, and F. Xia, “A secured health care application architecture for cyber-physical systems,” Control Engineering and Applied Informatics, vol. 13, no. 3, pp. 101–108, 2011.
[16] L. Wang, G. Von Laszewski, A. Younge et al., “Cloud computing: a perspective study,” New Generation Computing, vol. 28, no. 2, pp. 137–146, 2010.

[17] C. Gong, J. Liu, Q. Zhang, H. Chen, and Z. Gong, “The characteristics of cloud computing,” in Proceedings of the 39th International Conference on Parallel Processing Workshops (ICPPW’10), pp. 279–279, September 2010.

[18] A. Kapadia, S. Myers, X. Wang, and G. Fox, “Secure cloud computing with brokered trusted sensor networks,” in Proceedings of the International Symposium on Collaborative Technologies and Systems (CTS ’10), pp. 581–592, May 2010.

[19] P. Mell and T. Grance, “The NIST definition of cloud computing,” Tech. Rep., National Institute of Standards and Technology, 2011.

[20] M. Fouquet, H. Niedermayer, and G. Carle, “Cloud computing for the masses,” in Proceedings of the 1st ACM Workshop on User-Provided Networking: Challenges and Opportunities (UNET ’09), pp. 31–36, ita, December 2009.

[21] A. Gavrilovska, S. Kumar, K. Raj et al., “High-performance hypervisor architectures: virtualization in HPC systems,” in Proceedings of the 1st Workshop on System-Level Virtualization for High Performance Computing, 2007.

[22] C. Quinton, R. Rouvoy, and L. Duchien, “Leveraging feature models to configure virtual appliances,” in Proceedings of the 2nd International Workshop on Cloud Computing Platforms (CloudCP ’12), April 2012.

[23] R. D. Caytiles, S. Lee, and B. Park, “Cloud computing: the next computing paradigm,” International Journal of Multimedia and Ubiquitous Engineering, vol. 7, pp. 297–302, 2012.

[24] H. B. Lim, M. Iqbal, and T. J. Ng, “A virtualization framework for heterogeneous sensor network platforms,” in Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems (SenSys ’09), pp. 319–320, November 2009.

[25] A. Kapadia, S. Myers, X. Wang, and G. Fox, “Secure cloud computing with brokered trusted sensor networks,” in Proceedings of the International Symposium on Collaborative Technologies and Systems (CTS ’10), pp. 581–592, May 2010.

[26] M. Yuriyama and T. Kushida, “Sensor-cloud infrastructure physical sensor management with virtualized sensors on cloud computing,” in Proceedings of the 13th International Conference on Network-Based Information Systems (NBiS ’10), pp. 1–8, September 2010.

[27] V. Mauch, M. Kunze, and M. Hillenbrand, “High performance cloud computing,” Future Generation Computer Systems, vol. 29, no. 6, pp. 1319–1644, 2012.

[28] D. Harkins and D. Carrel, “The Internet Key Exchange (IKE), RFC 2409,” IETF Network Working Group, 1998, http://www.ietf.org/rfc/rfc2409.txt.

[29] F. Ricciardi, “MIT Kerberos Consortium: Kerberos protocol tutorial,” 2007, http://www.kerberos.org/software/tutorial.html.

[30] S. Kent and T. Polk, “Public-key infrastructure (x.509) (pkix) charter,” http://datatracker.ietf.org/wg/pkix/charter/.

[31] E. Rescorla, “Diffie-Hellman key agreement method, RFC 2631,” IETF Network Working Group, 1999, http://www.ietf.org/rfc/rfc2631.txt.

[32] J. B. D. Joshi, E. Bertino, U. Latif, and A. Ghafoor, “A generalized temporal role-based access control model,” IEEE Transactions on Knowledge and Data Engineering, vol. 17, no. 1, pp. 4–23, 2005.

[33] T. Bray and S. J. DeRose, “Extensible markup language (XML),” World Wide Web Journal, vol. 2, no. 4, pp. 27–66, 1997.

[34] N. Provos and D. Mazières, “A future-adaptive password scheme,” in Proceedings of the Annual Conference on USENIX Annual Technical Conference, p. 32, 1999.

[35] P. You, Y. Peng, W. Liu, and S. Xue, “Security issues and solutions in cloud computing,” in Proceedings of the 32nd International Conference on Distributed Computing Systems Workshops (ICDCS ’12), pp. 573–577, 2012.

[36] J. Hu, J. Gu, G. Sun, and T. Zhao, “A scheduling strategy on load balancing of virtual machine resources in cloud computing environment,” in Proceedings of the 3rd International Symposium on Parallel Architectures, Algorithms and Programming (PAAP ’10), pp. 89–96, December 2010.

[37] A. M. Nakai, E. Madeira, and L. E. Buzato, “Load balancing for internet distributed services using limited redirection rates,” in Proceedings of the 5th Latin-American Symposium on Dependable Computing (LADCI11), pp. 156–165, April 2011.

[38] Red Hat, “Kernel-based Virtual Machine,” http://www.linux-kvm.org/.

[39] P. Levis and N. Lee, “TOSSIM: A Simulator for TinyOS Network,” 2003.

[40] “TinyOS Tutorial,” http://www.tinyos.net/tinyos-1.x/doc/tutorial/.

[41] “Web Services Architecture, W3C,” February 2004, http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/wsa.pdf.

[42] Q. Wang, C. Wang, K. Ren, W. Lou, and J. Li, “Enabling public auditability and data dynamics for storage security in cloud computing,” IEEE Transactions on Parallel and Distributed Systems, vol. 22, no. 5, pp. 847–859, 2011.

[43] N. Cao, C. Wang, M. Li, K. Ren, and W. Lou, “Privacy-preserving multi-keyword ranked search over encrypted cloud data,” in Proceedings of IEEE Conference on Computer Communications (INFOCOM ’11), pp. 829–837, April 2011.

[44] S. Subashini and V. Kavitha, “A survey on security issues in service delivery models of cloud computing,” Journal of Network and Computer Applications, vol. 34, no. 1, pp. 1–11, 2011.
