Multi-layer Cloud Architectural Model and Ontology-based Security Service Framework for IoT-based Smart Homes

Ming Tao[a], Jinglong Zuo[b,*], Zhusong Liu[c], Aniello Castiglione[d], Francesco Palmieri[d]

[a] *College of computer and network security, Dongguan University of Technology, Dongguan, China*
[b] *College of Computer and Electronic Information, Guangdong University of Petrochemical Technology, Maoming, China*
[c] *School of Computer Science and Technology, Guangdong University of Technology, Guangzhou, China*
[d] *Department of Computer Science, University of Salerno, Via Giovanni Paolo II, 132 I-84084 Fisciano (SA), Italy.*

Abstract

The Smart Home concept, associated with the pervasiveness of network coverage and embedded computing technologies is assuming an ever-growing significance for people living in the highly developed areas. However, the heterogeneity of devices, services, communication protocols, standards and data formats involved in most of the available solutions developed by different vendors, is adversely affecting its widespread application. In this paper, promoted by several promising opportunities provided by the advances in Internet of Things (IoT) and Cloud Computing technologies for facing these challenges, a novel multi-layer cloud architectural model is developed to enable effective and seamless interactions/interoperations on heterogeneous devices/services provided by different vendors in IoT-based smart home. In addition, to better solve the heterogeneity issues in the presented layered cloud platform, ontology has been used as a promising way to address data representation, knowledge, and application heterogeneity, and an ontology-based security service framework is designed for supporting security and privacy preservation in the process of interactions/interoperations. Challenges and directions for future work on smart home management have been also discussed at the end of this paper.

Keywords: Smart Home, Heterogeneity, IoT, Cloud, Ontology, Security.

1. Introduction

The idea of smart home, defined as an intelligent environment that is able to acquire and apply knowledge about its occupants and surroundings to provide more humanized services and make home life more comfortable, safe and energy-efficient, has been considered as a challenging research and industrial topic for many years [1]. In a typical smart home setting, multiple (and often proprietary) devices and service platforms developed by different vendors, using heterogeneous communication protocols and standards, are deployed. Such heterogeneous devices and service platforms, however, need to be fully interoperable in order to support the joint and harmonized execution of household operations. Traditionally, to cope with heterogeneity issues, gateway technologies have been widely applied. In detail, a number of gateways need to be configured to convert a protocol into another one and/or re-map operational data between different formats. Unfortunately, these mediation and conversion operations significantly slow down the performance of the involved devices and often limit the degree of integration among them, so that, developing new strategies and architectural models to provide effective and seamless interactions/interoperations between heterogeneous hardware and software solutions in the smart home environment, still remains a fundamental challenge. Due to their significant impacts on the whole information and communications technology (ICT) scenario, the

*Corresponding author: ZuoJingL@126.com
Email addresses: taoming6723@126.com (Ming Tao), ZuoJingL@126.com (Jinglong Zuo), liuzs@gdut.edu.cn (Zhusong Liu), castiglione@ieee.org (Aniello Castiglione), fpalmieri@unisa.it (Francesco Palmieri)

recent advances in IoT and cloud computing technologies have provided some promising opportunities for addressing such challenge [2].

In our specific scenario, IoT does not refer to a single technology but, instead, to a new paradigm characterized by the pervasive presence around us of a variety of objects (referred as 'things') participating into the domestic activities, such as radio frequency identification (RFID) tags, sensors, actuators, mobile phones, connected each other by using a common multi-service converged IP network [3] and be able to interact and cooperate in order to achieve common home-related goals [4, 5]. Recently, the technological advances in IoT have fostered the rapid development of devices, services and applications that are perfectly suitable for smart homes [6]. These new devices and components are aimed to support new efficient and fully integrated services that leverage the existing ubiquitous and pervasive communication and computing facilities characterizing the home cyber environment. In fact, their convergence within the Internet arena significantly accelerates the massive deployment of various smart devices, appliances and solutions for automating and processing the information required by specific home services. However, the integrations of these home devices and services in specific domains characterized by strong cross-platform interactions/interoperations needs have resulted in several administration and operational problems, and, at the same time, the available communication platforms and hardware/software solutions empowering these services are still evolving and growing in quantity, by exacerbating the aforementioned interactions/interoperations issues.

Cloud computing, based on the concepts of converged infrastructure, unlimited scaling, elastic and shared services, can be the immediate response for the high dynamic nature, resiliency and adaptivity needs characterizing the processing and storage demands of the smart home [7]. Such runtime and storage capabilities are of paramount importance for implementing the aforementioned integrated intelligence facilities in the home scenario by transforming the traditional service provisioning model and facilitating the processing and storage of home-related data, as collected by the sensing/control and monitoring devices and/or available in most of the modern services/facilities (e.g., environmental monitoring, energy management, surveillance, lighting control, assisted living, entertainment, etc.). In the past few years, researchers have proposed some solutions that leverage cloud computing for implementing smart home systems accommodating multi-vendor services based on the service-oriented architectural model (SOA) [8]. These systems provide a number of software services (e.g., home management or home device control) re-mapped in a typical Software-as-a-Service (SaaS) cloud architecture to satisfy different requirements of household life. Such services are now required to interact with each other in order to exchange information and provide a solid basis for implementing collaborative home service in a fully distributed Internet-based environment (e.g. an intelligent building or, better, a Smart City) that reflects the organization of modern societies.

It should be also noted that the use of both IoT and cloud computing in smart home is still in its early stage and most of the existing proposals have not fully exploited the potential of these technologies for supporting interactional/interoperable architectures and solutions. To this end, we propose to use a combination of both the technologies as the enabling infrastructure for developing a multi-layer cloud architectural model for IoT-based smart home, in which, all the interactions/interoperations issues on the heterogeneous devices and services provided by different vendors will be properly solved in a systematic way. Indeed, we also argue that the combination of the semantic modeling and service-oriented technologies can support both interactions/interoperations and scalability in the above scenario. Accordingly, ontology has been identified as one of the most promising means that can be used to address data, knowledge, and application heterogeneity as well as to construct the security-oriented service framework in smart home environments.

The rest of the paper is organized as follows. In Section 2, we provide a brief review of the applications of IoT and cloud computing technologies in the smart home environment. In Section 3, a multi-layer cloud architectural model for IoT-based smart home is firstly developed to improve scalability and provide the interoperability for multiple home devices and services from different vendors. In Section 4, in the presented layered cloud platform, an ontology-based security service framework to handle heterogeneity for effective and seamless interactions/interoperations is developed, concretely, smart home domain ontology and ontology-based device description model are firstly defined, on the basis, Semantic Web Rule Language (SWRL) is used to define the reasoning rules needed to implement the mutual understanding and interactions/interoperations on the heterogeneous devices and services, and ontology-based security management is then designed to achieve security and privacy preservation in the process of interactions/interoperations. In Section 5, evaluation of the proposed layered cloud architectural model, and proofs of security & privacy requirements within the proposed ontology-based security service framework are performed. Challenges and directions for future work on smart home management are discussed in Section 6. Section 7 presents our conclusion.

2. IoT and Cloud Computing in Home Intelligent

IoT explains a future in which a variety of physical objects and devices around us, such as various sensors, RFID tags, positioning facilities, and mobile devices will be associated to the Internet, and allows these objects and devices to connect, cooperate, and communicate within social, environmental, and user contexts for achieving common goals [9, 10]. As an emerging technology, IoT is expected to embed computer intelligence into the devices needed for conveniently managing modern home environments.

In recent years, some preliminary works using IoT technologies to design and implement smart home have been presented. Ghayvat et al. [11] present a universal IoT-based smart home model, in which, all the home devices and appliances are connected together and the home network is the integration of different wireless technologies. Soliman et al. [2] and Lin et al. [12] present a smart home approach which consists of embedding intelligence into sensors and actuators by using the Arduino platform, and networking smart things by using ZigBee technology. By integrating IoT and service component technologies, Li et al. [13] present a smart home system architecture which has considered the heterogeneous information fusion in IoT. Lee et al. [14] focus on security issues in IoT-based smart home system, including physical security, information acquisition and transmission as well as processing security to ensure the confidentiality, completeness and authenticity to the whole system.

Cloud computing has also been employed to reshape home services and applications in the home automation domain. As more and more home devices from different vendors are equipped with on-board modules that can access the Internet, new solutions emerged to integrate existing home networks, various sensors, on-board modules in home devices, home gateways and cloud computing for creating smart-home-oriented clouds. They suggest that smart-home-oriented clouds are technologically feasible and will have a significant impact on the family and society once they are built. Thus, both existing home applications and a variety of information resources are being virtualized and packaged into services which are often combined and used to implement the mapping, encapsulation, aggregation, and composition facilities allowing home devices to interact/interoperate each other in order to perform joint execution of household operations.

Using the modular multi-layer approach and SOA to integrate various home services and applications revealed to be one of the most promising options available for building smart home cloud platforms, the smart home architecture proposed by Wu et al. [15] is a peer-to-peer (P2P) model based on multiple Open Services Gateway Initiative (OSGi) platforms, where SOA and mobile-agent (MA) technology are used to support the interactions between system components. Also OSGi-based, Cheng et al. [16] proposed an extensible architecture for heterogeneous smart home systems enabling dynamic integrations of devices, services and protocols. By taking into account of the distributed nature of the home environment with heterogeneous devices, Perumal et al. [17] presented an integrated approach using the SOAP/XML protocol for implementing effective web-service-enabled smart home management systems. Considering privacy protection issues in cloud platforms, Fabian et al. [18] proposed a peer-to-peer (P2P) infrastructure for organized sharing and private querying of data formed by many smart devices operating across several homes, whereas Kirkham et al. [19] proposed a risk-driven integrated home device management approach to achieve wider data sharing between the home and external services.

With the technological advances of both IoT and cloud computing, a new generation of solutions leveraging both IoT and cloud computing technologies has been developed to bring many benefits into smart home management. With the home growing and efficient energy concerns, Kau et al. [20] propose a cloud-based technology to perform remote control and monitoring of electrical appliances on the Internet. Respective using ZigBee-based energy measurement modules to monitor the energy consumption of home appliances and PLC-based renewable energy gateway to monitor the energy generation of renewable energies, Han et al. [21] propose a smart home energy management system (HEMS) architecture. By using communication and sensing technologies, and machine learning algorithm, Hu et al. [22] present a hardware design of smart home energy management system (SHEMS) to detect consumers activities and intelligently help consumers lower total payment on electricity without or with little consumer involvement. With the single resident and elderly care concerns in smart home, Benmansour et al. [23] present an overview of existing approaches and current practices for activity recognition and the latest developments and highlights of the open issues in this field. Suryadevara et al. [24] model a framework of activity recognition by using forecasting and reasoning methods to analyze the sensed temporal and spatial contextual information, which allow timely detection of the anomalous behaviors of the elderly and take corrective actions accordingly. Wu et al. [25] firstly make use of spatial features together with temporal features to discover useful representative activity instances, and then use learning algorithms [26, 27, 28] to do activity recognition model adaption. Cloud- and IoT-based frameworks and approaches integrating ontology methodologies for activity monitoring in smart home scenarios have attracted many research interests as well [29, 30].

While acknowledging the achievements of applying IoT and cloud computing technologies in home intelligent in these proposals, which have been found to be efficient, in this paper, to address the issue of enabling effective and seamless interactions/interoperations on heterogeneous devices/services from different vendors in IoT-based smart home, a novel layered cloud architectural model is proposed, moreover, in which, ontology-based approach is employed to better solve the heterogeneity issues, and an ontology-based security service framework is designed on the basis to achieve effective security and privacy preservation in the process of interactions/interoperations.

3. Multi-layer Cloud Architectural Model for IoT-based Smart Homes

In presence of an ever-growing amount of information sources in the smart home scenarios, structured in multiple sensing and control platforms/applications connected through several wireless and wire communication facilities, the fundamental challenge consists in collecting, integrating, aggregating and processing the huge amount of data originated by these sources in order to transform them in the knowledge needed by smart services provided in the modern home. This may imply managing many heterogeneous devices and protocols/technologies as well as performing cross-platform harmonization of their produced data, that becomes really feasible only by relying on the virtually unlimited storage and computing resources provided by cloud infrastructures. Furthermore, the virtualization facilities provided by clouds can significantly boost the limited computing capacity of hardware-constrained sensing or actuator devices making them be able to handle the complex processing tasks needed by modern smart home applications.

Currently, from various considerations, the vendors of home devices and appliances prefer to develop proprietary smart home platforms reflecting their own interests. These platforms often bring their own solutions and service interfaces, such that different communication protocols and standards are typically deployed within each solution. Hence, interconnecting heterogeneous devices and services provided by different vendors, and providing seamless interactions/interoperations across the available platforms remain the main challenges.

Building a public cloud based platform providing virtualization of the involved objects and their interfaces, and allowing their orchestration into generalized on-demand smart home services, may be an effective strategy for facing the above challenges and avoiding conflicts between the different private platforms characterizing the legacy vendor solutions.

Figure 1 shows the layered scheme of our proposed cloud architectural model for IoT-based smart home. Generally, different layers have different purposes and the bottom layers provide foundational supports for the top layers. By integrating under a common cloud-based platform, various IoT devices, e.g., sensors, actuators, controllers, mobile phones, and other home appliances, interconnect by using the available wireless (e.g., Bluetooth, RFID, ZigBee, Wi-Fi, 3/4G, LTE, etc.) and wire communications technologies [31, 32]. A specific middleware stratum is used to hide the implementation details of the underlining technologies and to provide support for the integration of specific applications deployed on the smart home cloud. SOA here will also be employed to integrate different information and connect multiple devices from different vendors seamlessly through the smart home cloud. SOA allows smart home application developers to organize, aggregate and package applications into new advanced home services. In each legacy private platform, the

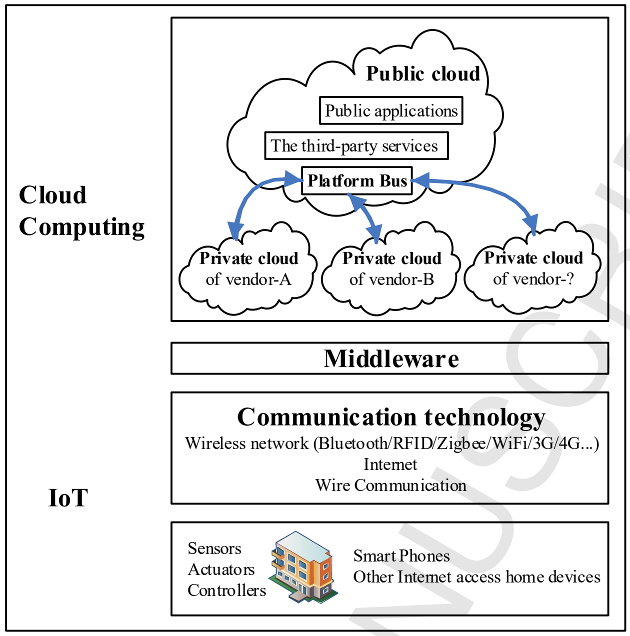


Figure 1: The layered cloud architectural model for IoT-based smart home.

used communication and access protocols and standards, as well as the device registration, authentication, management and manipulation methods, are individuated by the vendors. In the public cloud, providing the virtualized service and device/object interfaces for third party access to home services and devices, the platform bus implements protocol conversion and addressing operations with the IDs for all the registered devices in the platform. By leveraging such SOA- and IoT-based smart home cloud platform, innovative services can be developed by device vendors, government agencies and third-party service providers.

In the designed multi-layer cloud platform for IoT-based smart home, when the customer wants to manipulate a home device, the following two scenarios should be considered.

The operating process in the first scenario that the consumer and the target device are associated to the same private platform is shown in Figure 2, and the crucial operating procedures are simply described as follows.

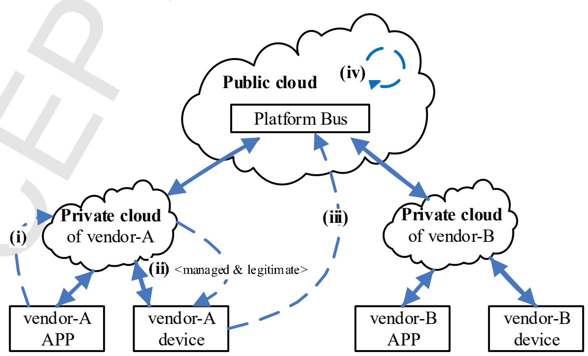


Figure 2: An illustration of the operation process in the scenario that the consumer and the target device are associated to the same private platform.

(i) The customer uses the vendor-specific companion *App* installed on the smart phone to send an operation command to the associated private platform directly.

(ii) The $DeviceID$ of the target device will be locally checked at first in such (presumably private cloud) infrastructure. If the target device is managed by the same associated private platform and the operation command is achieved on a legitimate basis, the operation command will be forwarded to the target device associated to it (e.g., connected to the corresponding private cloud).

(iii) After completing the requested manipulation, the target device sends its current status to the associated private platform. The relevant parameters about the current operating status of the target device then will be reported to the platform bus in the public cloud.

(iv) The platform bus synchronizes the device status with all the other associated private platforms.

The operating process in the second scenario that the consumer and the target device are associated to different private platforms is shown in Figure 3, and the crucial operating procedures are simply described as follows.

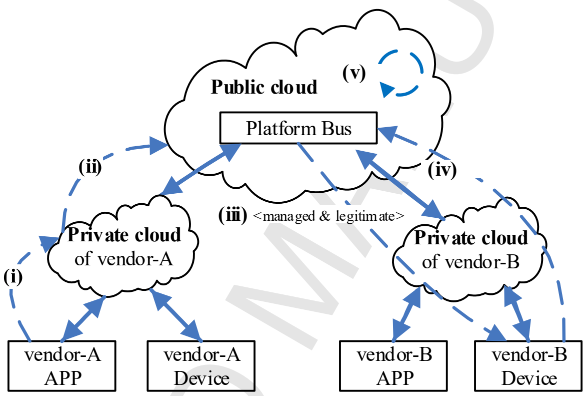


Figure 3: An illustration of the operation process in the scenario that the consumer and the target device are associated to different private platforms.

(i) The customer uses the vendor-specific companion *App* installed on the smart phone to send an operation command to the associated private platform directly.

(ii) The $DeviceID$ of the target device will be at first checked locally in such infrastructure. If the target device is not managed by the private platform associated by the consumer, the operation command is forwarded to the platform bus in the public cloud.

(iii) The platform bus then forwards the operation command to the corresponding private platform by performing the addressing operation with the $DeviceID$, and the operation legality will be verified in the private platform associated by the target device. If the operation command is achieved on a legitimate basis, it will be forwarded to the target device associated to it.

(iv) After completing the requested manipulation, the target device sends its current status to its associated private platform. The relevant parameters about the current operating status of the target device then will be reported to the platform bus in the public cloud.

(v) The platform bus synchronizes the updated device status in the whole cloud platform just as stated above.

From the above specifications, we can clearly see that, by checking the *DeviceID* of the target device in the private platform associated by the consumer, if the target device and the customer are associated to the same private platform, the associated private platform can directly trace the target device, and the operating process is relatively simple; otherwise, if the target device and the customer are associated to different private platforms, the public cloud platform will execute the addressing operation with the *DeviceID* of the target device and redirect the operation command to the private platform associated by the target device. After accomplishing the requested operation on the target device, the relevant parameters about the operating status will be synchronized in the whole cloud platform. Accordingly, we can come to a conclusion that, with such a multi-layer cloud architectural model, by generalizing the scope of each individual service represented by using an Internet-like structure and integrated into a common IoT service fabric for sharing and reusing in multiple operating homehold contexts, and enabling data collection and exchange among different platforms, interactions/interoperations among all the registered home devices and services from different vendors, it allows the seamless interworking of the legacy platforms (typically private clouds) provided by different vendors through the aforementioned public cloud layer, and real-time, cheap and on-demand home services could be efficiently enabled.

4. Ontology-based Security Service Framework

In the IoT-based smart home, the proposed mediation platform based on multi-layer cloud architectural model provides seamless interworking for the home devices from different vendors. Starting from such basis, ontology is used as a promising way for addressing data, knowledge, and application heterogeneity in the available devices in order to realize the aforementioned virtualized smart home service framework [33]. Such ontologies are able to model and describe the different aspects of the IoT resources involved in the smart home by defining their semantic properties, the information they can supply or the actions/controls they can perform. To this end, the smart home domain ontology and an ontology-based device description model are firstly defined in this section, on the basis, Semantic Web Rule Language (SWRL) is used to define the reasoning rules needed to implement the mutual understanding and interactions/interoperations among the heterogeneous devices and services, and ontology-based security management is finally discussed to achieve security and privacy preservation in the process of interactions/interoperations.

4.1. Domain Ontology of Smart Home

The smart home domain ontology is structured through a set of correlated concepts abstracted from the smart home scenario, but independent of any particular technology or implementation. In terms of the level of abstraction, the concepts are classified into several levels realizing a hierarchical structure. To the best of our knowledge about the services offered in smart home scenario, as shown in Figure 4, the top layer structure capturing the general features of home entities is defined as the following ontologies developed by Protégé, *Home_device*, *Entertainment*, *Environment*, *Data_communication* and *Security*. The home services corresponding to *Home_device* include automatic cooking and cleaning, household environment monitoring, surveillance, etc.. To make daily home life convenient, as well as improving efficiency and implementing energy savings policies, the *Environment* services are mainly related to managing temperature, humidity and lighting by providing automatic adjustment and adaption or remote control of air conditioning, lights, gas and other unnecessary appliances running in standby mode or being turned off in the case of leaving the house. The *Entertainment* services include providing various audio-visual feasts for the householder at any time, automatically recording family TV programmer preferences, quickly accessing into the network for interactive services, etc.. The *Security* services are mainly related to raising alerts and delivering them to householder via phone or Internet, and triggering relevant solutions to protect house safety when there are abnormal home situations, besides, supports a high abstraction level for dealing with security objectives in the process of interactions/interoperations. The *Data_communication* services mainly encompass data sharing between the home and external services via Internet, data exchanging between the home devices via short-distance wireless communications technology, etc..

The details of general concepts and their features in each sub-domain are defined in the low-level structure of smart home domain ontology. *Home_device* for example, is the abstraction of device entities in smart

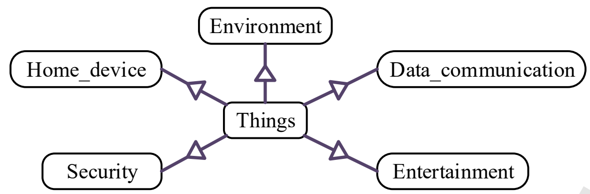


Figure 4: The top level structure of smart home domain ontology.

home, whose structure and concepts are shown in Figure 5. In the *Home_device* concept, the associated *Smart_Home_device* and *Common_Home_device* concepts are defined, together with their sub-concepts. Additionally, the inheritance relations between concepts are indicated by the solid arrow and the non-inheritance relations are indicated by the dotted one. For clarity sake, in Figure 5, only the non-inheritance relations of *Gas_sensor* are presented as examples, and the specific explanations of the defined non-inheritance relations will be illustrated in the following. *Environment* concept and its low-level concepts are shown in Figure 6, which has four low-level concepts: humidity, smoke, temperature and gas. Similarly, the basic concept and the low-level ones of *Entertainment*, *Data_communication* and *Security* can be defined in the same manner.

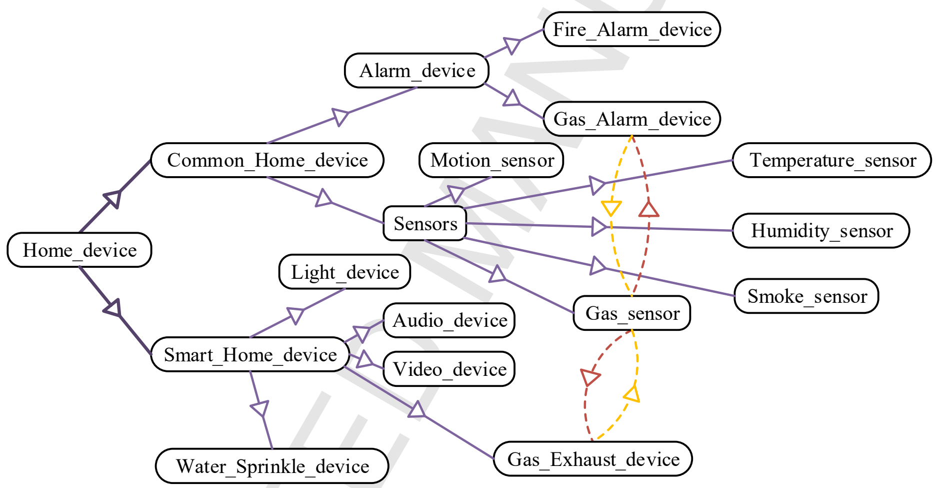


Figure 5: *Home_device* concept and its low-level concepts.

In the aforementioned smart home domain ontology, each abstracted concept is characterized by its properties and its relations with other concepts. To achieve the interactions/interoperations on the heterogenous home devices and services, the relations between concepts to be used as the basis of reasoning, should be defined. By considering *Gas_sensor*, for example, as shown in Figure 7, the two mutually-inverse relations of ‘sensor’ and ‘sensedby’ are defined for *Gas_sensor* and *Gas*. If *Gas_sensor* detects that the abnormal gas concentration exceeds the pre-defined standard threshold, *Gas_Exhaust_device* would be triggered to exhaust the abnormal gas. Hence, the two mutually-inverse relations of ‘trigger’ and ‘triggeredby’ should be defined for *Gas_sensor* and *Gas_Exhaust_device*. Similarly, since *Gas_Alarm_device* would be triggered by *Gas_sensor* in the same manner, the two mutually-inverse relations should also be defined for *Gas_sensor* and *Gas_Alarm_device*. Additionally, *Gas_Alarm_device* and *Gas_alarm* have the two mutually-inverse relations of ‘cause’ and ‘causedby’.

4.2. Ontology-based Device Description Model

In the process of interactions/interoperations on heterogeneous devices, for taking full advantage of their specific capabilities in order to support self-description and automated communication features, a description

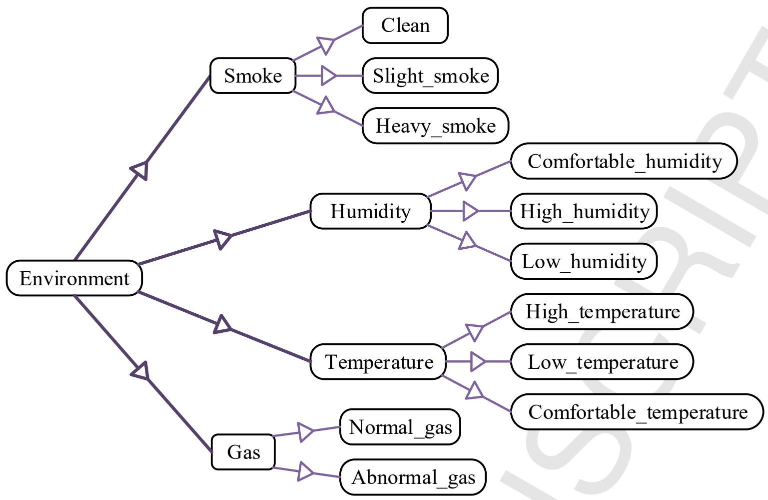


Figure 6: *Environment* concept and its low-level concepts.

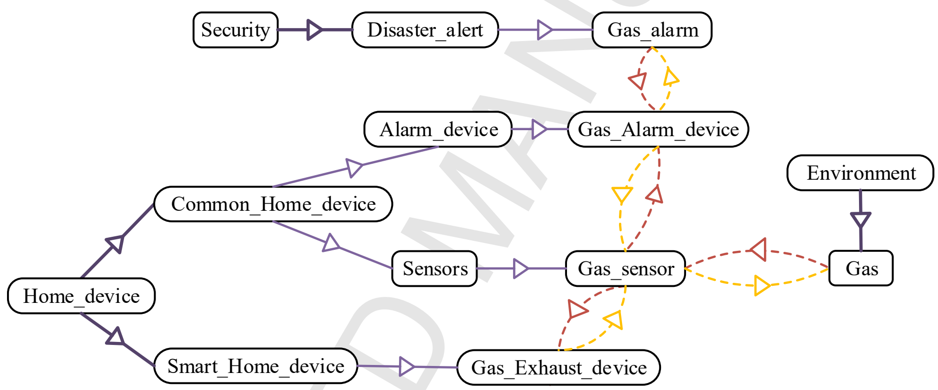


Figure 7: The relations between *Gas_sensor* and its neighbor concepts.

model of devices' capabilities (which could be processed and understood by other entities) should be provided.

The devices in smart home are characterized by many related information, such as function, location, content, status and controller. Therefore, six related ontologies are constructed in the device description model shown in Figure 8, namely *Device*, *Function*, *Content*, *Location*, *Status* and *Controller*. The embedded device functions are described in the ontology of *Function*. All the possible device statuses are defined in the the ontology of *Status*. The locations of devices are the individuals of *Location* ontology. The content, such as a condition or a context needed to select which device implements a specific operation is defined in the the ontology of *Content* [34]. The identity of the authorized controller and the assigned control competence are defined in the *controller* ontology. The *Device* ontology can communicate with other ontologies through the object properties.

Moreover, the basic device information is defined by the *DeviceDescribeFile* class shown in Figure 9. The basic device information includes *DeviceName*, *DeviceID*, *DeviceType*, *Output*, *AccessURI*, *DriveURI*, *AccessPermission* and *Interface*. In the presented multi-layer cloud architectural model for IoT-based smart home, *DeviceID* should be divided into a local ID used in the proprietary platform, and a private ID and a public ID taken as the identifiers used for recognizing the device identity and achieving resource identification in different layers. Similarly, each *AccessURI* should be divided into local URI, URI in private

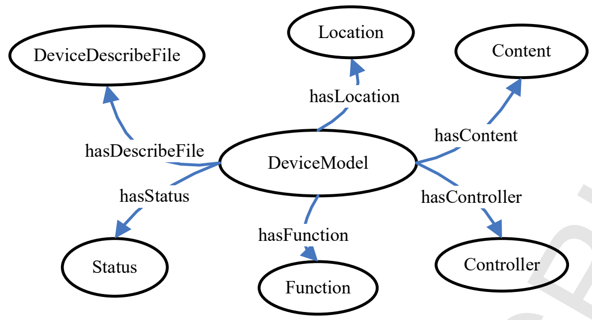


Figure 8: Device description model.

cloud (associated to the legacy platform) and URI in public cloud, and can give an entry point for accessing the command and operation list of the devices available in the different layers. *AccessPermission* also should be divided into local permission, permission in private cloud and permission in public cloud, and would be used to implement security and access control in the different layers.

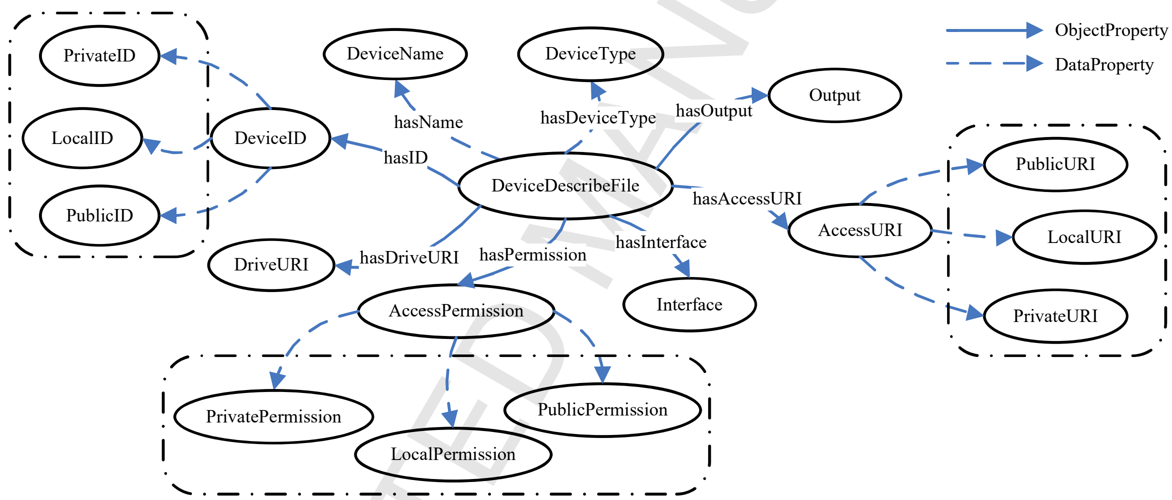


Figure 9: *DeviceDescribeFile* class.

4.3. SWRL-based Reasoning Description for Interactions/Interoperations

Reasoning is an important inherent function of ontology, and reasoning rules can be added as a part of the defined ontologies to infer the information implied into them [35, 36]. In this work, to achieve full and seamless interactions/interoperations on the heterogeneous home devices and services provided by different vendors, the above defined ontologies and the device description model are respectively taken as reasoning foundation and reasoning object. SWRL is used as the tool of choice for defining the reasoning rules necessary to implement the mutual understanding and interactions/interoperations among the involved heterogeneous devices and services [37].

For example, the reasoning rule for detecting and handling abnormal gas concentrations is defined in Figure 10 (a). If the gas sensor at somewhere detects that the gas concentration exceeds a pre-defined standard threshold, the gas exhausting device would be triggered and the alarm flagging the presence of abnormal gas would be triggered as well.

$$
\begin{aligned}
& Location(?x) \\
& \wedge(Gas\_sensor(?sensor\_ID) \wedge atLocation(?sensor\_ID,?x) \wedge Environment\_Gas(?g,?x) \wedge swrlb: greaterThan(?g, concentration\_threshold)) \\
& \wedge(GasDevice(?device\_ID) \wedge hasFunction(?device\_ID, exhaust) \wedge atLoaction(?device\_ID,?x)) \\
& \wedge(GasAlarmDevice(?device\_ID) \wedge atLocation(?device\_ID,?x)) \\
& \rightarrow TriggerGasDevice(?device\_ID) \wedge TriggerGasAlarmDevice(?device\_ID)
\end{aligned}
$$

(a) Detecting and handling abnormal gas

$$
\begin{aligned}
& Location(?x) \\
& \wedge((Smoke\_sensor(?sensor\_ID) \wedge atLocation(?sensor\_ID,?x) \wedge Environment\_smoke(?s,?x) \wedge swrlb: greaterThan(?s, concentration\_threshold)) \\
& \vee(Temperature\_sensor(?sensor\_ID) \wedge atLoaction(?sensor\_ID,?x) \wedge Environment\_temperature(?t,?x) \wedge swrlb: greaterThan(?t, temperarure\_threshold))) \\
& \wedge(WaterDevice(?device\_ID) \wedge atLoaction(?device\_ID,?x) \wedge hasFunction(?device\_ID, sprinkle)) \\
& \wedge(FireAlarmDevice(?device\_ID) \wedge atLocation(?device\_ID,?x)) \\
& \rightarrow TriggerWaterDevice(?device\_ID) \wedge TriggerFireAlarmDevice(?device\_ID)
\end{aligned}
$$

(b) Detecting and handling fire alarm

Figure 10: Reasoning descriptions for detecting and handling abnormal gas and fire alarm.

Similarly, the reasoning rule for fire alarm is defined in Figure 10 (b). If the smoke sensor at somewhere detects that the smoke concentration exceeds a pre-defined standard threshold, or the temperature sensor detects that the environment temperature exceeds another pre-defined threshold, the water sprinkling device and fire alarm device would be triggered.

4.4. Ontology-based Security Management for Interactions/Interoperations

In the developed cloud architectural model for IoT-based smart home, complexity of interactions/interoperations between the service providers and customers still impose significant security requirements. To satisfy the security and privacy requirements, it is prerequisite to elaborately design relevant security polices to achieve security and privacy preservation [38]. Here, by developing the ontology of *Security* that defines a common security vocabulary shared by service providers and customers, ontology-based security management for supporting effective and security interactions/interoperations is discussed.

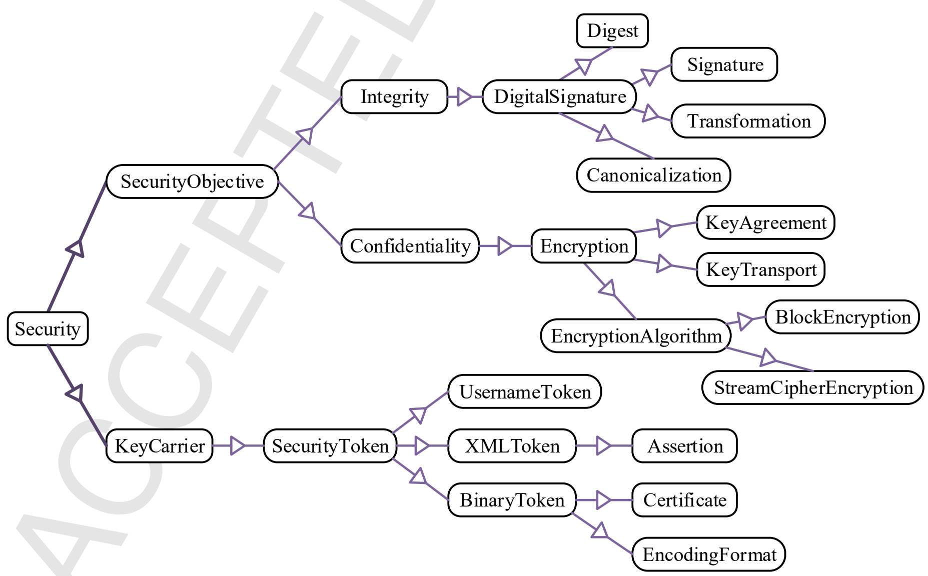


Figure 11: The ontology of *Security*.

In Figure 11, the ontology of *Security* is presented, which consists of the main classes, properties, associations and relationships, and supports a high abstraction level for dealing with security objectives for interactions/interoperations. Certainly, it can be enriched with additional security technologies by introducing new classes and properties. In the defined ontology, the top-level class named *Security* has some properties, mainly including *SecurityObjective* and *KeyCarrier*. The *SecurityObjective* class indicates the security objectives (e.g., integrity and confidentiality) in the process of interactions/interoperations, which can be captured in the ontology by defining two subclasses, *Integrity* and *Confidentiality*. The mechanism of digital signature as a technique is represented by *DigitalSignature* class and is associated with the security objective of integrity. It has the following properties. *Digest* class is used to capture digest algorithms including instances, i.e., MD5 (Message Digest Algorithm), SHA1 (Secure Hash Algorithm), SHA256 and SHA 512. *Signature* class is used to represent instances of signature algorithms, including DSA-SHA1(Digital Signature Algorithm-SHA) and RSA-SHA1 (Rivest Shamir Aldeman-SHA). *Transformation* class is used to specify transformation algorithms including instances, i.e., XSLT (eXtensible Stylesheet language Transformation), XPath (XML Path Language), Enveloped Signature, SOAP (Simple Object Access Protocol) Message Normalization and Security Token Reference (STR) Dereference Transform. *Canonicalization* class includes these instances, such as, XML Canonicalization and Exclusive XML Canonicalization. The mechanism of encryption as a technique is represented by *Encryption* class and is associated with the security objective of confidentiality. It has the following properties. *KeyAgreement* class is used to specify key agreement algorithms including Diffie-Hellman instance. *KeyTransport* is used to represent key transport algorithms including these instances, such as, RSA-v1.5, and RSA-OAEP (RSA-Optimal Asymmetric Encryption Padding). *EncryptionAlgorithm* class is used to capture encryption algorithms, which has two subclasses, *BlockEncryption* and *StreamCipherEncryption*. The former includes these instances, i.e., 3DES (Triple Data Encryption Standard), AES-128 (Advanced Encryption Standard), AES-192 and AES-256, while the instance, i.e., RC4 (Rivest Cipher), is included in the latter one.

Because both signature and encryption should use security keys, *KeyCarrier* class is introduced to represent mechanisms of carrying security keys. As a common used carrying mechanism of keys, tokens are employed to hold keys within or outside of the messages in the process of interactions/interoperations, and *SecurityToken* class as a subclass of *KeyCarrier* is defined. Because different types of tokens have different manners of attaching them to the messages, *SecurityToken* has the following three subclasses. *UsernameToken* class provides a method of verifying usernames in the process of interactions/interoperations. *BinaryToken* class defining binary-formatted security tokens includes two properties. Note that, *EncodingFormat* property defines the encoding formats of tokens. *XMLToken* class defines XML-based security tokens, and *Assertion* as its subclass represents security assertions.

Based on the above defined ontology of *Security*, security polices can be designed to indicate the abilities of interactions/interoperations between the service providers and customers along with the security management in the developed cloud architectural model for IoT-based smart home. Specifically, from different perspectives, both service providers and customers should define different policies describing security properties in the process of interactions/interoperations, and different policies should be able to achieve intersections to enable the implementation of interactions/interoperations with reasonable security levels.

```
<p: Policy>
    <p: ExactlyOne>
        ( <p: All>
            ( <Assertion …> … </Assertion> )*
          </p: All>* )
    </p: ExactlyOne>
</p: Policy>
```

Figure 12: Normal form of security policy.

The normal form of designed polices including indispensable components is shown in Figure 12, where, p is a prefix for the namespace URI of policies, $Policy$ is the root element indicating a policy, $ExactlyOne$ as an operator is used to gather policy alternatives represented by All operators, $Assertion$ as the elements gathered by All operator are used to represent the security requirements put forwarded by customers or the service security capabilities released by providers, they can use the concepts defined in the $Security$ ontology. In addition to these indispensable components, the following general-purpose components can be included in the policies to facilitate the manipulation, i.e., either $Name$ attribute or ID may be used to represent the identification of a policy, $Service$ elements may be contained in a policy of provider to represent the service implementation, or contained in a policy of customer to represent the requested component services and interaction/interoperation capabilities, $Reference$ element may be used to nest the content of a policy into another policy.

In the process of interactions/interoperations, the intersection operations among defined security policies that are compatible are usually necessary to determine the services released by providers whose security policies are suitable for customer policies. Determined by using OWL-based operators, if the capability of a assertion of one policy alternative could satisfy the requirement of a assertion of another policy alternative, the two assertions belong to different policy alternatives would be compatible and the two policy alternatives could be taken as compatible ones, and at least, if a pair of alternatives between two policies are compatible, the two policies would be taken as compatible ones as well.

With the developed $Security$ ontology and designed security polices, in the case that a service customer imposes the ontological concept of $Confidentially$ into the policy corresponding to a component service in the process of interactions/interoperations, the confidentiality preservation must be enabled in the service implementation. Examples of encryption assertion and token assertion extracted from a policy for the service implementation are shown in Figure 13. Encryption assertion indicates that the body of service content is encrypted by using one kind of encryption algorithms, i.e., AES-128, and the encryption mechanism will use one kind of tokens with some encoding format defined in token assertion. As shown in the example of token assertion, the token of X.509 PKI Path Version 1 with the format encoded by Base64 is used.

Example of encryption assertion

```
<sec:AES-128>
    <sec:Token>
        <sec:Reference URI="# X.509PKIPathToken "/>
    </sec:Token>
    <sec:EncryptedParts>
        <sec:Body/>
    </sec:EncryptedParts>
</sec:AES-128>
```

Example of token assertion

```
<sec:X.509PKIPath-v1
    u:Id=" X.509PKIPathToken"
    EncodingFormat="sec:Base64"/>
```

Figure 13: Examples of encryption assertion and token assertion.

5. Experiments and Analysis

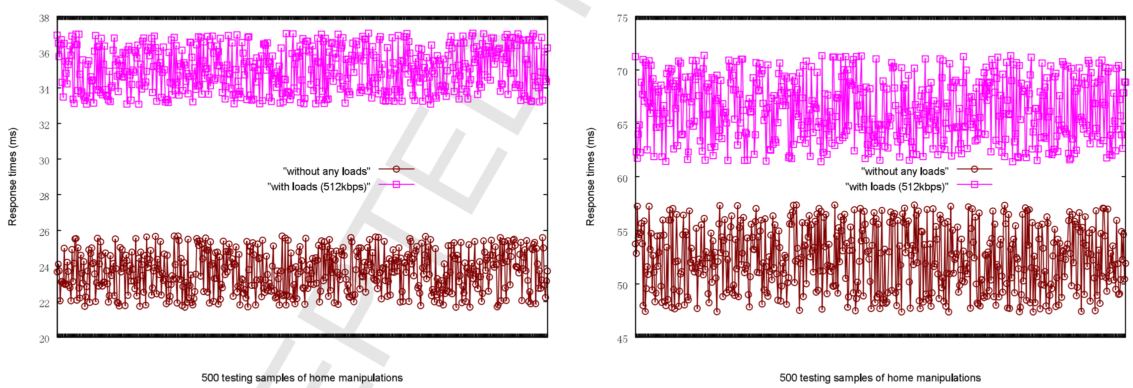
5.1. Evaluation of Layered Cloud Architectural Model

To qualitatively analyze and evaluate the performance of multi-layer cloud architectural model which is proposed to address the issue of interactions/interoperations, we design a prototype consisted of a public cloud provided by Amazon EC2, a private smart home cloud platform supported by Guangdong University Scientific Innovation Project and built in Dongguan University of Technology (DGUT), and a private smart home cloud platform authorized by Canbo CO., LTD, China. As shown in Table 1, the two private platforms are constructed by employing completely different cloud architectures, concretely, the former is constructed using some open-source solutions, and the latter is constructed using VMware solutions. In addition, in the first private platform, the deployed home devices and appliances using different network access technologies are provided by different vendors; in the second private platform, the deployed kitchen and bathroom devices and appliances are the independent productions of Canbo CO., LTD, but the other kinds of deployed home devices and appliances are provided by different vendors and use different network access technologies. For effective and seamless interactions/interoperations in the same associated platform or across the heterogeneous platforms in smart home environment, the response time defined as the maximum execution time taken by systems tasks is significant and crucial for real-time home manipulation applications. Accordingly, the performance of the proposed layered cloud architectural model is measured with respect to the response times of home manipulations.

Table 1: Configurations of private smart home cloud platforms

Private platform	Configuration parameters			
	Hardware configuration	OS	Virtualization software	Management software
DGUT	CPU: Intel Xeon E3-1231v3 RAM: 16GB Storage: 1TB	Ubuntu Server 12.04 LTS	KVM	OpenStack
Canbo	CPU: Intel Xeon E7-4850v3 RAM: 128GB Storage: 15TB	Ubuntu Server 16.04 LTS	VMware vSphere	VMware vCenter

In the prototype, considering two different network situations, e.g., without any loads and with loads (512kbps), the response times in the two scenarios that the consumers and the target devices are associated to the same private platform (shown in Figure 2) and the consumers and the target devices are associated to different private platforms (shown in Figure 3) are show in Figure 14. In the two scenarios, total of 500 testing samples of home manipulations are performed respectively. The further analysis results of the response times are shown in Table 1. From Table 2, we can clearly see that, within the proposed layered cloud architectural model, the test values of average response time are justified for the requirements of home manipulation applications, especially for the interactions/interoperations across heterogeneous platforms.



(a) Scenario shown in Figure.2 (b) Scenario shown in Figure.3

Figure 14: The response times in two different network situations.

5.2. *Proofs of Security & Privacy Requirements*

In this section, the proposed ontology-based security service framework is analyzed with respect to the critical security and privacy preservation requirements.

1) In the proposed layered cloud platform, only the registered entities (e.g., home devices and *Apps*) from different vendors can generate valid certificates including *ID*s and public keys, consequently, other

Table 2: Comparisons of response times.

Performance evaluation		Scenario shown in Figure.2	Scenario shown in Figure.3
Average response time (ms)	Without any loads	23.68	52.36
	With loads (512kbps)	35.06	66.38
Standard deviation (ms)	Without any loads	6.93	8.68
	With loads (512kbps)	11.58	15.63

illegal entities cannot eavesdrop using different IDs and public keys. Therefore, we can conclude that the verification requirement could be satisfied.

2) For mutual authentication between the entities before the process of interactions/interoperations, with the instances of signature algorithms contained by *Signature* class, the legitimate entities participating into the interactions/interoperations could generate valid signatures based on the generated keys, with which, the participant entities could mutually determine the identities and proceed with normal interaction/interoperation sessions consequently. Therefore, the proposed ontology-based security service framework also satisfies the requirement of mutual authentication.

3) With the defined subclasses and contained instances in the *KeyCarrier* class, session keys in the process of interactions/interoperations could be only shared between the authorized entities within the whole expiry period. Thus, data confidentiality and integrity in the sessions can be ensured.

4) With the contained instances in the defined *DigitalSignature*, *Encryption* and *SecurityToken* subclasses, the private information of participant entities could be effectively protected during the process of ongoing authentications or interaction/interoperation sessions. Moreover, because the participant entities will trigger the next interactions/interoperations by applying for new session keys, eavesdroppers or adversaries are also unable to correlate the sessions and derive previous or subsequent interrogations. Therefore, strong anonymity and untractability of the participant entities within the proposed ontology-based security service framework could be efficiently ensured.

5) A majority of well-known security attacks could be efficiently prevented. For example, with the contained instances of digest and signature algorithms, the generated shared secret authentication and session keys can efficiently defeat *impersonation* attacks and *repudiation* attacks. Similarly, with the contained instances of key agreement algorithms, adversaries could not decrypt the encrypted message with the private key owned only by the certified entities in the process of interactions/interoperations, $Man-In-The-Middle(MITM)$ attacks can also be defeated; because the shared secret keys for ongoing interaction/interoperation sessions are different and will be regenerated for newly initiated sessions, the well-known key attacks can be prevented as well. Finally, with the contained instances of transformation algorithms and key transport algorithms, *redirection*, *replay* and *injection* attacks all could be efficiently resisted.

6. Challenges and Future Work

New architectures and platforms for smart home management, such as the cloud- and IoT-based one proposed in this work must be provably efficient, scalable, secure and reliable before starting their large-scale deployment. Existing mechanisms and approaches, however, are not yet fully satisfactory in meeting all these requirements at the same time. There are still some serious challenges described as follows.

i) *Global standards for architecture, device interconnection, service integration*

Since there are a number of stakeholders such as device and service vendors involved in smart home clouds, and there are complex dependencies among these stakeholders as well, global standards are essential to avoid incompatibilities and conflicts between privately developed platforms and solutions. However, establishing global standards to lower the complexity and make smart home clouds more compatible and cost effective, remains a challenge. Further efforts on standardization should be conducted to coordinate various resources for implementing more effective smart home clouds and reducing the number of adaptations and mediation stages.

ii) *Scalability, performance and technology integration*

The effectiveness of smart home clouds depends on their scalability in handling a dynamically growing number of homes. Apart from handling regular operations of home devices, smart home clouds must be able to face the ever-growing demands for home entertainment and some other applications, and provide the interactions/interoperations among the heterogeneous devices and services from different vendors, such that further and more advanced developments aimed at optimizing the utilizations of computing, storage and network resources are needed. Meanwhile, the realization of optimization algorithms that coordinate the private platforms/clouds with the public one to achieve real-time cross-layer data synchronization and minimize the traffic overhead between layers is necessary as well. In addition, with the launch of new home devices and technologies each year, developing effective IoT middleware that supports integration of these new technologies and devices with the existing ones will be challenging.

iii) *Security and privacy*

Security and privacy are other fundamental concerns within smart home clouds. A low security level is clearly unacceptable for home services regarding operations safety and people health. For example, when we go out for some time, unnecessary services such as air conditioning, lights, gas and other appliances will be put in standby mode or turned off to save energy and protect house safety, however, the attackers may maliciously send from the outside many fake requests to some specific device or cloud service, by bringing serious threat to house safety. Therefore, reliable and well-balanced security frameworks preventing unauthorized access or disclosure of home privacy, are needed to enhance the security and trust of cloud services without limiting the overall system flexibility. Additionally, reasonable efforts in law and regulations are also needed to effectively provide security guarantees in the smart home sector.

7. Conclusion

Undoubtedly, smart home technologies are changing and improving people's life experience. However, the increasing heterogeneity issues seem to restrict their widespread application. Starting from these considerations, in this work, a novel multi-layer cloud architectural model is developed for IoT-based smart home, which provides a substantially improved degree of interactions/interoperations between heterogeneous home devices and services provided by different vendors. In addition, in the developed layered cloud architectural model, ontology is used to discuss how new homehold services can be constructed in order to make the smart home platforms more useful and better solve the heterogeneity problems introduced by the use of different devices/solutions to implement effective and security home services.

Such IoT- and cloud-based platforms are expected to be the backbone of the future smart home with the ultimate goal of making home living experience more comfortable and enjoyable. However, research on integrating IoT and cloud computing within the smart home scenario is still in its infancy and the existing studies on this topic are still insufficient. To make IoT and cloud enabled smart home platforms be more useful, new advanced home services, e.g., home device remote monitoring and control, multimedia entertainment, etc., need to be developed and reasonably deployed, and business intelligence should be massively introduced in the smart home ecosystem. Additionally, there are still a number of challenges to be faced when developing future integrated smart home scenarios, such as lack of global standards, scalability, performance as well as security and privacy. Because of the complexity involved in addressing these challenges, the collaboration among academia, home device companies, law enforcement organizations, government authorities, standardization groups and cloud service providers, as well as a systematic approach in engineering new architectures and operating schemes, are definitely needed. Although the problems that are still open are very severe, IoT and cloud computing provide tremendous opportunities for technology innovation in the smart home industry, and will serve as enabling infrastructures for developing a new generation of network-centric home services where the participating home entities are distributed on a metropolitan area scale and cooperated in a federated way within the future smart cities.