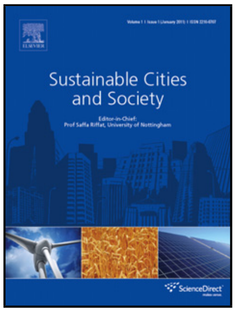


A Lightweight Cyber Security Framework with Context-Awareness for Pervasive Computing Environments

Jalal Al-Muhtadi, Kashif Saleem, Sumayah Al-Rabiaah, Muhammad Imran, Amjad Gawanmeh, Joel J.P.C. Rodrigues

PII: S2210-6707(20)30827-1

DOI: https://doi.org/10.1016/j.scs.2020.102610

Reference: SCS 102610

To appear in: *Sustainable Cities and Society*

Received Date: 10 June 2020

Revised Date: 11 November 2020

Accepted Date: 12 November 2020

A Lightweight Cyber Security Framework with Context-Awareness for Pervasive Computing Environments

Jalal Al-Muhtadi, Kashif Saleem*, Sumayah Al-Rabiaah,
Muhammad Imran, Amjad Gawanmeh, Joel J. P. C. Rodrigues

Jalal Al-Muhtadi, and Sumayah Al-Rabiaah, are with College of Computer and Information Sciences (CCIS), King Saud University, Riyadh, 11653, KSA ({jalal, salrabiaah}@ksu.edu.sa).

Jalal Al-Muhtadi, Kashif Saleem, and Joel J. P. C. Rodrigues, *Fellow, IEEE,* are with Center of Excellence in Information Assurance (CoEIA), King Saud University, Riyadh, 12372, KSA (ksaleem@ksu.edu.sa).

Muhammad Imran is with College of Applied Computer Science, King Saud University, Saudi Arabia (cimran@ksu.edu.sa).

Amjad Gawanmeh is with University of Dubai, College of Engineering and IT, Dubai, UAE (agawanmeh@ud.ac.ae).

Joel J. P. C. Rodrigues, *Fellow, IEEE*, is with Federal University of Piauí (UFPI), Teresina-PI, Brazil and Instituto de Telecomunicações, Portugal (e-mail: joeljr@ieee.org).

ACKNOWLEDGEMENT

The authors extend their appreciation to the Deanship of Scientific Research at the King Saud University for funding this work through research group no (RG-1439-022).
This work is partially supported by the FCT/MCTES through national funds and when applicable co-funded EU funds under the Project UIDB/EEA/50008/2020; and by Brazilian National Council for Research and Development (CNPq) [via Grant No. 309335/2017-5].
The authors thank the Deanship of Scientific Research and researchers support and services unit (RSSU) at King Saud University for their technical support.

Highlights

- Smart environments should protect sensitive information from exposure or monitoring.
- A lightweight cyber security framework with context-awareness for pervasive computing environment is proposed.
- The proposed framework employs a symmetric key encryption to leverage data confidentiality and sigital signature-based hash function to ensure integrity.
- Implementation of the proposed framework is performed in a real cloud environment.
- Experimental results demonstrate the effectiveness of the proposed framework.

***Abstract*—** Internet of things (IoT) plays a key role in enabling smart sustainable cities. Pervasive computing over the IoT platform makes life more convenient by embedding sensors based on context-aware computing devices in the physical environment for the ubiquitous availability of computing resources. The sensors gather contextual information from the physical world and transmit it to receivers as per requirements or in case of environmental changes, such as temperature and humidity. However, the combination of dynamic operation and the need to handle sensitive and private data make the pervasive computing environment and IoT devices vulnerable to numerous attacks. Smart environments require a maximum level of safety assurance, such as trusted context producers and customers, which should protect sensitive information from exposure or monitoring. This paper discusses the major cyber threats in smart environments and proposes a novel lightweight security framework that authenticates and maintains the context providers and receivers. The cloud environment is adopted for user authentication at the user layer to implement access control and role assignment. Finally, the proposed security framework is implemented in the IBM cloud platform with six devices to evaluate its efficiency, sustainability, and secure communication.

Keywords**—**Context-aware, Cyber Security, Data Privacy, Internet of Things, Pervasive Computing, Smart Environment, Sustainability.

1. Introduction

Humans are increasingly dependent on computer systems, mobile devices, and software, with a strong demand for access to computing resources on a 24/7 basis regardless of location. In addition, many consumer devices are now designed to be connectable and programmable, providing value-added services and novel automation possibilities via the internet of things (IoT) for enabling smart sustainable cities (Bibri, 2018). The value of IoT is increasing at a growth rate of 39%, with its prediction to reach about $520 billion in 2021, and by 2023 the expenditure should go beyond US$22 billion annually (Lee, 2019). However, managing and using all these devices is becoming more complex, and may require education. These demands have led to the development of a new computing paradigm known as pervasive computing, which provides targeted access to computing resources ubiquitously. The pervasive computing environment (or smart environment) requires computing devices embedded throughout the physical world, which makes them transparent, seamless, and convenient for communication (Weiser, 1999; Yu, Ma, Cao, & Lu, 2013).

The key difference between pervasive computing and conventional desktop computing is the ability of the pervasive computing to facilitate spontaneous interaction. The main features of the pervasive computing (Bettini et al., 2010) include universality—a large number of different types of computing devices are arranged and embedded into the computing environment (Gao, Zhu, Gong, Tan, & Zhou, 2016)—dynamism—most users are mobile, thereby resulting in a dynamic change in the structure of the computing system (Lu & Liu, 2012)—transparency—the user is unaware of the underlying process involved in providing services (Shankar, Camp, Connelly, & Huber, 2012)—adaptability—the environment can sense and infer requirements, as well as provide users with the necessary information services automatically (Li, Nastic, & Dustdar, 2012)—diversity—different devices are linked together to form a service without a fixed computing environment, hence devices have different computing powers and communication bandwidths (Hansmann, Merk, Nicklous, & Stober, 2013). The characteristics of the pervasive computing require that computing devices are proactive and provide intelligent computing services without relying wholly on service requests.

In addition to the active input from each user, the context-awareness technique is the most direct basis for the system to determine its own behavior. Context refers to the computing environment, which records and perceives information that may affect its interactions and thus plays an active supporting role. It helps the system to handle tasks intelligently and automatically without requiring user attention, thereby preventing adverse effects of attention interference on the interaction process. There are three context categories in a pervasive computing environment: the environment context—representing factors such as location, speed, and time (Perera, Zaslavsky, Christen, & Georgakopoulos, 2014)—the device context—representing factors such as network bandwidth, and other device features (Bahl, Han, Li, & Satyanarayanan, 2012) —the user context—encompassing personal preferences and requirements (Rahimi, Ren, Liu, Vasilakos, & Venkatasubramanian, 2014). The three contexts communicate with each other, automatically adjust the interaction mode and user interface, and process the necessary tasks without disturbing the user.

Pervasive computing often features a smart environment, where many different types of computing equipment and context-aware modules are embedded and hidden in the physical world but cooperate with each other and offer services actively. In a smart environment, the sensors and applications interact to make life easier for users by exploiting information about the current environment. This information may be private or sensitive, e.g., the identities of the users in a given physical space (Tang & Li, 2012). Context-awareness is closely associated with smart environments (Cook & Das, 2012), where sensors can detect and respond to surrounding features such as temperature and humidity. Context-aware means that an application automatically adapts to discovered context by changing its behavior (G. Chen & Kotz, 2000). The large amount of context data captured by sensors causes the sensors to modify themselves and the actions of their applications to move from the current state to another state.

Current context information is provided by sensors, i.e., a small device that perceives the current context in the smart environment and transforms it into a signal that can be understood by computing devices (Al-Muhtadi et al., 2018). The sensor may convey personal information such as a user's name and address in the pervasive environment, thereby making the user vulnerable to attacks (Bouachir, Aloqaily, Tseng, & Boukerche, 2020; K. Kim, Kim, & Lim, 2017; Yaseen et al., 2018). To ensure security, the context information should be known by authorized applications only. The security system should be able to authenticate devices and applications that want to participate in the smart environment, i.e., the authentication is not restricted to humans but may include entities, such as devices, applications, libraries, and sensors (Laufs, Borrion, & Bradford, 2020). Thus, the smart environment only reacts according to the context information that is produced by trusted sensors. The security system should be pervasive and transparent (Al-Muhtadi, Ranganathan, Campbell, & Mickunas, 2003).

Pervasive computing based on a smart environment allows new kinds of applications and functions to be provided and widely implemented, thereby yielding user convenience (Bettini & Riboni, 2015). In one example, a context-aware application can monitor active badges representing particular users to track their location, thus allowing incoming calls to be forwarded to the nearest telephone (G. Chen & Kotz, 2000). Another example is an aware home application that operates home appliances and services for residents based on the current context (Covington, Fogla, Zhan, & Ahamad, 2002). A secure middleware was designed to support context-aware applications based on sensors, but the smart environment and its applications often experience security limitations due to the environment is dynamic and associated with the physical world (Tang & Li, 2012). Besides, the smart environment depends heavily on contextual information from multiple sources. The conventional mechanisms do not guarantee security; hence, it is more difficult to ensure confidentiality, integrity, and authentication, thereby exposing users to a wide range of attacks (Farivar, Haghighi, Jolfaei, & Alazab, 2020; S. Kim, Lee, Kim, & Yoon, 2017).

Recent studies on the threats posed by IoT, such as ownership, functional restrictions, physical security, and infrastructure, have resulted in several recommended countermeasures (Chaudhry, Ibrahim, & Bashir, 2016; Khurshid et al., 2019). Firstly, to distribute trust, connected devices should be produced by different manufacturers or run by different service providers, which is the fundamental principle of IoT initiatives. IoT devices generally have resource limitations, so the manufacturers cannot use data encryption for protection, hence allowing attackers to eavesdrop easily on IoT traffic. Secondly, the auto-configuration of IoT devices is a significant shortcoming because security issues are inherited from the TCP/IP suite, thereby leading to a growth in common threats like address resolution protocol spoofing, address table-based attacks, watering hole attacks, and de-auth attacks. Therefore, it is necessary to customize an intrusion detection systems, but this is not widely applied in pervasive environments. Finally, issues such as IoT device ownership, business model, device life cycle, personal data security, and reliable information delivery are still threats to IoT (Al Ridhawi, Otoum, Aloqaily, Jararweh, & Baker, 2020).

Most studies on smart environment have targeted security problems from the perspective of access control to services or context-based authentication principles, but the trust of context providers and prevention of data manipulation by unauthorized receivers have not been extensively studied. Therefore,

- this work presents the design of a cybersecurity framework to guarantee that the contextual information is only accessed by authorized applications, thereby ensuring the completeness and confidentiality.

Cerberus (Al-Muhtadi et al., 2003) is taken because it can tackle many security problems in a context-aware pervasive computing environment, such as offering customers adjustable services in terms of non-intrusive authentication and access control in pace with the modification of contextual information.

- Cerberus is a mechanism extended by implementing the authentication of sensors and context information receivers.

The safe channels are provided to convey the contextual information to authorized receivers using encryption to guarantee its integrity and confidentiality. Consequently, the proposed system accommodates the addition of new sensors.

- The Cerberus security services are enhanced in a manner to achieve a dynamic and safe context-aware pervasive computing environment.
- The work done in the preliminary report (Al-Rabiaah & Al-Muhtadi, 2012) is improved by additional mathematical details with the adoption of a cloud environment for user authentication in the cloud user layer to implement access control and role assignment. By doing so, the proposed system can be

used to overcome the current cyber threats in the IoT and pervasive environment mentioned by (Chaudhry et al., 2016).

- Moreover, the novel security framework has been analyzed in the real environment by implementing it in the IBM cloud platform with six devices that are configured to send the secure data in parallel over the secure channels in real time.

Hence, the proposed framework achieves efficient and secure data communication in pervasive environments. The rest of this work is organized as follows. Section 2 presents the related work; Section 3 presents the proposed IoT threat model; Section 4 set out the system design; Section 5 presents the implementation; finally, Section 6 presents the conclusion and future work.

2. Related Work

ZigBee technology is a wireless local area network protocol with a low data rate, short time delay, low power consumption, and high security. It is mainly used in short-range wireless sensor networks (WSNs), consumer electronics, automatic control, logistics management, and smart home areas (Al-Muhtadi et al., 2018; K. Saleem, Fisal, & Al-Muhtadi, 2014). In 2010, the Bluetooth Alliance developed Bluetooth v4, which has low power consumption. It has been applied to IoT sensors, medical equipment, and smart wearable devices (Gomez, Oller, & Paradells, 2012). Radio-frequency (RF) communication technology and RF identification (RFID) technology are the most common RF technologies. The former offers long transmission distances, strong wall crossing, low energy consumption, and low cost (Dobkin, 2012). It is widely used in smart home areas, and the product development cycle is relatively short but has some vulnerabilities (Good & Benaissa, 2013; Yaseen et al., 2018). In smart environment networks, data is mainly based on a small number of bursts, and the transmission rate is low. However, it has higher requirements for transmission distance, security, network capacity, low power consumption, and delay. Table 1 compares several wireless network schemes (Kamal, Parvin, Saleem, Al-Hamadi, & Gawanmeh, 2017; Yang, 2016). The security and privacy challenges of IoT are highlighted in (Yaqoob et al., 2017) and for smart cities in (Braun, Fung, Iqbal,

& Shah, 2018). The authors in (Azab, Alazab, & Aiash, 2016), (Alazab & Tang, 2019), (Vinayakumar et al., 2020) and (Amanullah et al., 2020) studied how machine learning, deep learning and big data technologies can be adopted for industrial IoT security. The security requirements of IoT and its key components were described in (Khattak, Shah, Khan, Ali, & Imran, 2019). The focus of this study is the security of two key enabling IoT technologies—i.e., RFID and sensor network—at the perception layer.

Most studies on context-awareness in smart environments focused on functionality rather than security issues, but several reports have considered various aspects of security in smart environments. The role-based access control (RBAC) model uses an access control mechanism based on the subject role to classify subjects according to their properties. For example, two context-aware solutions for privacy protection in smart space have been proposed for dynamic scenarios, thereby reducing communication between the user and system and exploring the communication between different users (Pallapa, Das, Di Francesco, & Aura, 2014). The middleware architecture, named context-aware security architecture, offers services for receivers, such as security management, authorization, and context management, and this can provide authorization and authentication techniques that are intuitive and non-intrusive (Covington et al., 2001; Covington, Moyer, & Ahamad, 2000). A generalized RABC (GRABC) algorithm, which expands the RBAC model, was developed to contain objects and system states (Covington et al., 2000). The access control policies are flexible and understandable because they adopt subject, object, and environment roles while using reasonable names, and there is no encryption. The security aspects of GRABC were enhanced by showing how changing the context affects security, where a context-aware access control model was developed to issue permissions to sets of roles, including both subject and environment aspects (Covington et al., 2001). The system gathers context information from the environmental sensors and then plays a safeguard role at home against attackers. For healthcare applications' protection, a dynamic context-aware security mechanism was proposed; the trust level for each tool was provided by the hospital administrator to ensure the reliability of the tool providers (Hu & Weaver, 2004). The access to a specific resource in a system depends on its trust level. The authors implemented their idea by using web services as context providers. The communication in their system was text based, using extensible markup language. They secured the communication using a secure sockets layer.

Table 1: Comparison of properties of wireless networks

	ZigBee	Wi-Fi 3	Bluetooth	RFID 433MHz
Tran. Distance (m)	<100	<100	<10	<250
Tran. Speed (bps)	20k/40k/250k	11-54M	1M/3M	500k
Power	Low	High	Middle	Low
Frequency (Hz)	2.4G/868M/915M	2.4G	2.4G	433M
Cost	Middle	High	Middle	Low
Penetrating power	Low	Middle	Low	High
Network Capacity (number of connections)	<65,000	<30	<7	<65,000
Response delay	Low	High	High	Low
Security	Middle	Low	High	Middle

Another context-aware framework for data association in healthcare systems offers an authenticated framework to guarantee that crucial data collected by sensors is allocated to the correct patient (Chowdhury & Light, 2009). Patients are authenticated via their physiological and personal information, using stable biometrics data like face and voice rather than variable data like heart rate. When the system gathers the contextual data from sensors, it will generate and analyze the current context of the patient. For listed wireless attacks' prevention, a secured method for physiological context data acquisition and transmission was proposed to allow dynamic operations (Chowdhury & Light, 2009).

An authentication framework has also been proposed to extend the Kerberos protocol (Al-Muhtadi, Ranganathan, Campbell, & Mickunas, 2002). Only a single sign is allowed, and an entity is authenticated via wearable/embedded devices containing a confidential level for user authentication. System flexibility is achieved through the support of multiple authentication levels and by allowing the dynamic addition of new authentication techniques. Errors in physical devices can also be predicted to make IoT services more trustable (K. Kim et al., 2017). The security recommendations include joint information security efforts among governments, academia, and the information protection industry to provide a more beneficial cyber environment for the society.

In pervasive computing, trustable computing services are required for transactions. To this end, communication between devices should be context-aware to address dynamic environments without consuming large amounts of energy, thereby extending the lifetime of battery-powered devices. A context-adaptive and energy-efficient transaction management mechanism has been proposed for the dynamic revision of transactions (Tang & Li, 2012). Such a mechanism could significantly reduce the number of failed transactions based on simulations. The context information can be applied to analyze user behavior, although it is difficult to extract special semantics, whereas forecasting multidimensional relationships can suggest context recognition, but the variety of context information will be chaotic. An algorithm named MR Tensor Cube has been proposed to handle big data based on a MapReduce framework, thus facilitating effective context recognition (S. Kim et al., 2017). The core of this algorithm reduces continuous data using a partial filter and slice and can be used for multi forecasting analysis in recommendation systems. The relative gain is proportional to the volume of available context information when compared to other methods.

The threats to information and services in a pervasive computing environment can be countered by an anonymous authentication and access control scheme to mitigate threats like eavesdropping (Djellali, Lorenz, Belarbi, & Chouarfia, 2014). However, this is limited to mobile users and services in the pervasive environment. Although the authors added an extra layer of biometric authentication, it is limited to the communication between mobile users and services and does not consider the challenges of using the proposed model for the exchange of advanced context-aware information, not only between mobile users but also between sensors, brokers, and receivers. Although there is a service connection delay in the process of authenticating users in a pervasive computing environment, this does not cause much delay overall (S.-K. Kim, Kim, & Min, 2015). The tasks carried out during authentication include proxy preparation, certification of identity, and mutual authentication; therefore, a method was proposed to mitigate the service connection delay in proportion to the message size (S.-K. Kim et al., 2015). However, the method was based on Jini IoT devices and did not apply to all heterogeneous IoT devices.

Other recent reports have provided potential sustainable solutions within the IoT environment. For example, WSNs fast authentication algorithm, which enhances the sensor nodes' cooperation process by speeding up the verification of vBNN-IBS signature, has been introduced (Benzaid, Lounis, Al-Nemrat, Badache, & Alazab, 2016). The authors of (Rahman, Hossain, Hassanain, & Muhammad, 2018) proposed fog cloud hybrid architecture to support an ad-hoc crowd consisting of a massive social network and distributed IoT nodes around a smart city environment. Although the communication architecture in this framework allowed connections between mobile users and fog nodes, as well as between fog nodes and the cloud, the framework did not address the issue of security. The opportunity to augment the informational landscape of smart sustainable cities with big data applications to realize the required level of environmental sustainability has also been addressed (Bibri, 2018). The framework lacks security support, but the work presented will help such sustainable solutions to be implemented with security support. In (Jararweh, Otoum, & Ridhawi, 2020), a trustworthy smart city service delivery solution is introduced for only the edge network side. The given solution provides the trustworthy and sustainble sevices on the basis of an intrusion detection system at every edge server powered by artificial neural network (ANN).

Most of the recent reports addressing security threats from the perspective of services access control or context information-based authentication principles included mechanistic defects that cannot guarantee the safety of sensors (context producers) or protect the data from malicious interference. Therefore, a mechanism, which aims to extend Cerberus by authenticating the sensors, context receivers (applications), and users, is proposed. Multiple secure channels, which convey the context data to authorized receivers and users in an encrypted manner within the cloud environment to ensure integrity and confidentiality, are employed. Section 4 presents a detailed interpretation of the novel framework.

3. Threat Model

In a pervasive computing environment, sensors generate information, including personal and sensitive information like the identities and locations of people in an environment. An attacker could break the integrity and gain access to contextual information by eavesdropping communications between sensors, system components, and applications. To prevent this, the contextual information should be encrypted. The mechanism is required to guarantee the authentication of receivers for contextual information. Therefore, the Cerberus scheme is extended by addressing the threats, as given in Figure 1, which also generalizes the framework for any type of platform.

- **Sensor Compromise**

The context provider is a small device that senses and measures quantities in its current physical environment, e.g., temperature or humidity, and converts them into signals that can be read by computing devices. In the proposed system, the sensors provide information that is needed by applications to run correctly. Some of this information is personal and sensitive, e.g., the identities and locations of people in an environment. Sensor compromise is an identity authentication problem (Figure 1, Threat 1). The threat can be executed when an attacker adds illegitimate sensors to the system or replaces trusted sensors with fake ones. To address these issues, a mechanism that authenticates the sensors needs to be defined to guarantee that contextual information is generated only by trusted sensors.

- **Communication Compromise**

A sensor (context provider) needs to communicate with computing devices that are interested in its acquired information. In the proposed system, the sensors communicate with the system components and applications over a network to transfer contextual information to their destination. Communication compromise is a problem concerning the integrity and confidentiality of transferred information (Figure 1, Threat 2). The threat can be executed when an attacker intercepts transferred contextual information, alters it, and sends the modified information to system components or applications, i.e., the attacker scams the recipients of contextual information. In addition, the threat can be executed when an attacker intercepts context information and benefits from it, particularly when the information is sensitive and private. To address these issues, a mechanism can be injected to ensure the integrity and confidentiality of the conveyed contextual information by offering safe communication between system components.

- **Recipient Compromise**

Sensors in smart environments generate contextual information, which is used by external applications to achieve their functionality, and if these recipients are attacked, the information can be misused. Recipient compromise is a problem concerning identity authentication (Figure 1, Threat 3). The threat can be executed when an unauthorized application receives sensitive or private contextual information, and benefits from it or exploits it. To address these issues, an authentication mechanism that guarantees the recipients of contextual information are trusted is required.

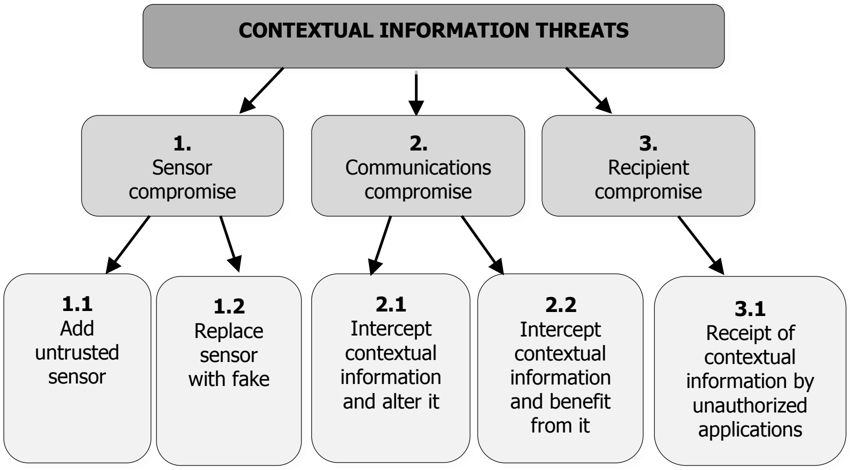

CONTEXTUAL INFORMATION THREATS
1. Sensor compromise
2. Communications compromise
3. Recipient compromise
1.1 Add untrusted sensor
1.2 Replace sensor with fake
2.1 Intercept contextual information and alter it
2.2 Intercept contextual information and benefit from it
3.1 Receipt of contextual information by unauthorized applications

Figure 1. Threats involving attacks on contextual information.

4. System Design

The proposed system design contains three layers: the WSN, application, and cloud user layers. This structure decomposes the security task into different levels, thereby making it easier to address the issues in each layer. In particular, the WSN layer is analyzed in detail; it uses different WSNs in different devices based on their features to enhance the security. Further, the design of the processes is summarized to show the authentication process that guarantees security, and the advantages of the system are briefly listed.

4.1.Overview of the System Design

The proposed system with multiple sensors aims to provide safe services for context-aware pervasive computing environments. The contextual information generated by these sensors is used by applications to ensure correct functionality. Figure 2 shows the focus of the framework. The system permits a context recipient to select the types of context information in its field of interest. For example, if a context recipient requires temperature and humidity information, then only this information will be issued by the system. The system model of the smart environment can be set as an event for users to obtain information services, and should thus include the following elements: sensor security, application service security, communication security, and user identity security. With the help of third parties (clouds), authentication and privacy protection can be implemented between both the sensor and recipient sides simultaneously.

In a smart environment, cloud user authentication is adopted to apply access control and to assign roles. Compared to the previous server-based intelligent system, the cloud solution has greater flexibility and stability and reduces costs as well. This solution could apply big data mining methods to sensor data, which is conducive to the feedback regulation of intelligent systems, thereby providing users with a more intelligent service. In terms of user identification, biometric technologies such as fingerprints and irises can be applied with high security. Figure 3 shows the system design and the mechanism of context information transfer.

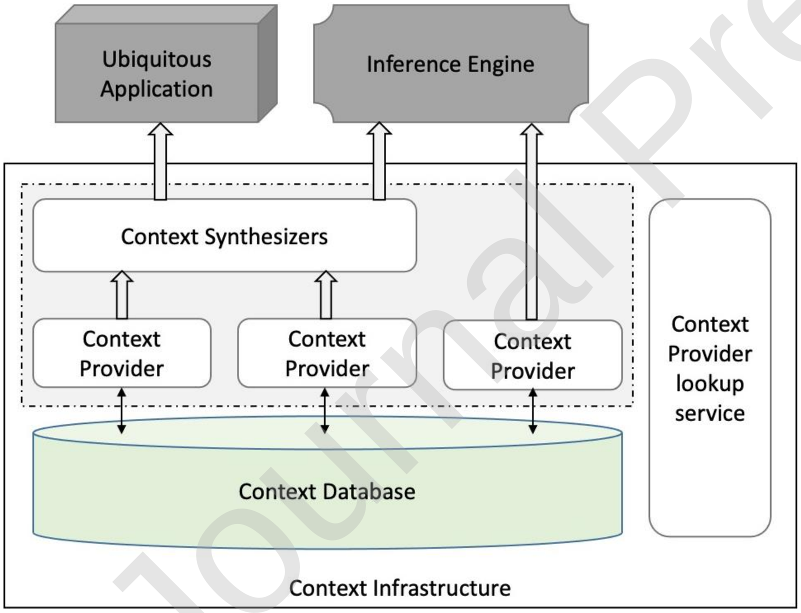


Figure 2. Framework of Cerberus and its context-aware security framework

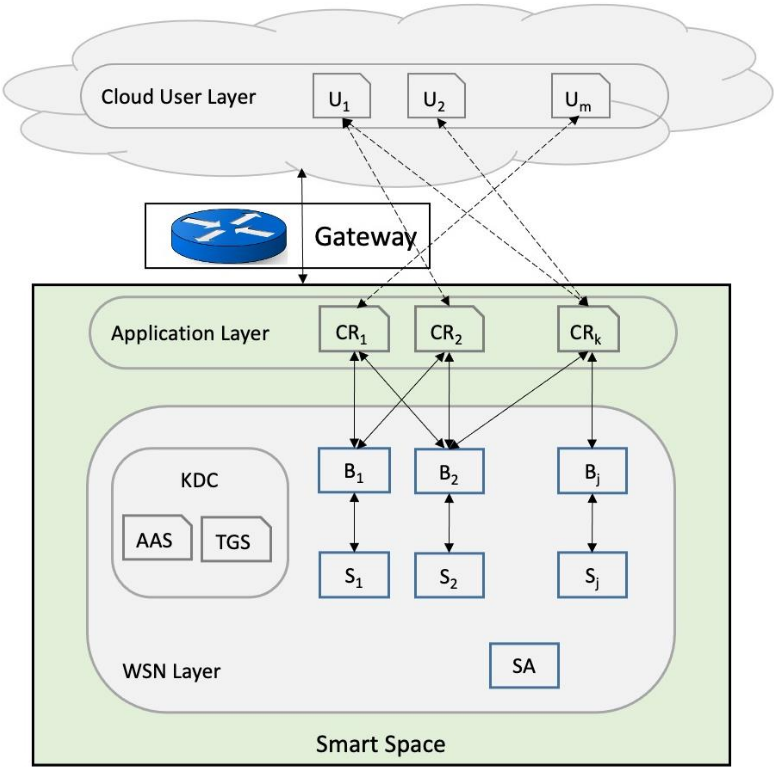


Figure 3. The system design process

The novel system has been designed as follows:

- The cloud environment is adopted for user authentication at the cloud user layer to implement access control and role assignment.
- The proposed system provides authentication services for sensors and applications using the Cerberus authentication protocol (Neuman & Ts'o, 1994; Kashif Saleem, Derhab, Al-Muhtadi, & Shahzad, 2015) with some extensions as described in Section *C*.
- Applying a symmetric key encryption method to the contextual information helps the proposed system to provide confidential services.
- Incorporating and utilizing hash functions (Aziz et al.) and digital signature mechanisms guarantee data integrity.

- A publish or subscribe model to execute communications in the system is adopted, where the sensors play the publishers role, the application components play the subscribers role, and the brokers are used for communications between the sensors and applications.
- A special broker is selected dynamically when a new sensor is introduced into the system.
- Multiple network types are adopted to configure the IoT and WSN layer to enhance the security of sensors and context information (Figure 4).

The following assumptions are made:

- The sensors and the applications cannot communicate with the system until they are authenticated by submitting their digital certificates (each participant in the system requires a digital certificate that contains its information).
- The system components are trusted based on certificates issued by a trusted third party.

To live in the system, a complete lifetime is given to every context recipient.

4.2. Framework Components

The components in the system can be defined as follows:

- Sensor (*S#*) is a context producer, which provides safe data concerning the current context (including sensitive data).
- Context Recipient (*CR#*) is an application, which requires contextual information from the sensors.
- Broker (*B#*) is an agent between a sensor and a corresponding context recipient; it conveys the information from the sensor to the recipient in an encrypted fashion.
- Sensor Authenticator (*SA*) authenticates new sensors before allowing them into the system. It also offers a key that is shared for data encryption between the sensor and the corresponding broker.

- Key Distribution Center authenticates and provides keys to context recipients that wish to join the system. It includes two parts: Application Authentication Service (*AAS*) and Ticket Granting Service (*TGS*).
- *AAS* authenticates context recipients that wish to join the system.
- *TGS* provides the keys that are required by authenticated recipients, thereby allowing them to communicate with the corresponding brokers.
- Gateway is a protocol converter and provides connectivity with the internet, and also connects two heterogeneous networks to achieve high-level communication (M. Chen, Wan, González-Valenzuela, Liao, & Leung, 2014).
- User (*U#*) refers to users in the cloud, who gain access and assigned roles.

In a smart environment, users may have many different devices. To enhance the security of sensors and contextual information, it is a good idea to adopt different types of communication protocol-based devices to configure the complete scenario (Figure 4). In the proposed solution, from the physical perspective, the untrusted or fake sensors find it difficult to communicate with the transponder or signal converter because of using access controls and encryption methods. Additionally, the noise sensors can be introduced to confuse and mislead attackers.

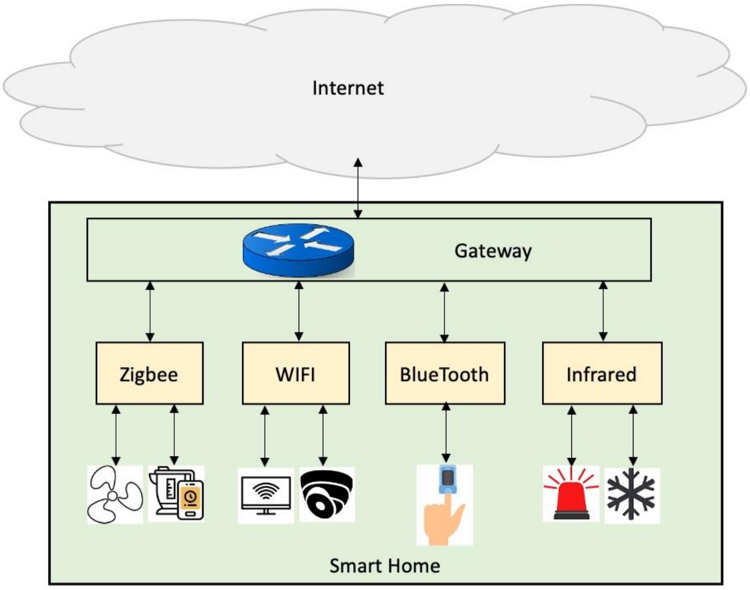


Figure 4. A smart environment featuring heterogeneous networks

4.3.System Processes Design

The system relies on three major types of keys for encrypted data transfer between its components:

(a) The key is shared between every sensor and its broker—this is utilized to encrypt the contextual information transfers from the sensor to the broker.

(b) Every corresponding context recipient and broker interested in a process is given a key—this is utilized to distribute the broker keys to recipients (Section 4.3.4).

(c) Every corresponding context recipient and broker interested in a process is assigned a temporary key—this is the same for each group utilized for encryption and is given to all context recipients interested in the same encrypted information from a broker.

Here, the expressions used in the system are defined.

h(m): hash the message (m)

$\mathbf{K_{AB}}$: A and B shared secret key

$\boldsymbol{d_A}$: Private key used by A

{, #187}$\boldsymbol{e_A}$: Encrypted message (m) using A's public key

$\boldsymbol{e_A}$: Public key used by A

{, #187}$\boldsymbol{d_A}$: Decrypted message (m) using A's private key

4.3.1. Authenticating the Context Provider

The sensor (S1) should be authenticated before it starts communicating with the system. The Cerberus authentication protocol (Pantsar-Syväniemi, Simula, & Ovaska, 2010) is used and is further enhanced to perform the following process and graphically by Figure 5:

i. **S1** assigns its credential to the **SA**.

ii. **S1** is then verified by **SA**, further **B1** is created as a broker for **S1**, and $\mathbf{K_{S1B1}}$ is created by **SA** and sent to **S1.**

Ticket issued to **S1**: {{B1_id, K_{S1B1}, TimeStamp}$\boldsymbol{e_{S1}}$}$\boldsymbol{d_{SA}}$, where K_{S1B1} is a key between **S1** and **B1.**

Ticket issued to **B1**: {{TB1=S1_id, K_{S1B1}, life}$\boldsymbol{e_{B1}}$}$\boldsymbol{d_{SA}}$.

iii. **S1** sends $\mathbf{K_{S1B1}}$ to **B1** by the checking and decryption of its ticket.

{{TB1=S1_id, K_{S1B1}, life}$\boldsymbol{e_{B1}}$}$\boldsymbol{d_{SA}}$ is the ticket of **B1** received from **SA**.

{auth=TimeStamp}K_{S1B1}: Authenticator.

iv. $\mathbf{K_{S1B1}}$ is derived by the decryption and verification of **B1's** ticket and responds with {auth=TimeStamp+1}K_{S1B1}, following the decryption of the authenticator.

The decryption is performed {auth=TimeStamp+1}K_{S1B1} by **S1** that shows the **S1** has interacted with **B1** as shown in Section 4.*C*.3.

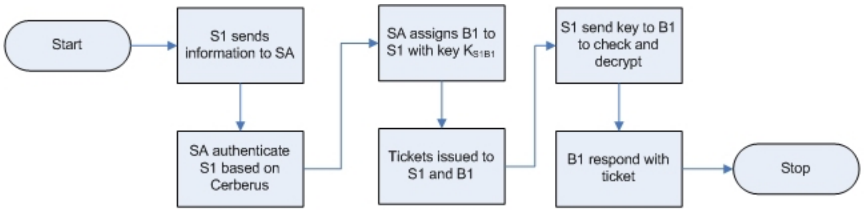

Start
S1 sends information to SA
SA authenticate S1 based on Cerberus
SA assigns B1 to S1 with key K$_{S1B1}$
Tickets issued to S1 and B1
S1 send key to B1 to check and decrypt
B1 respond with ticket
Stop

Figure 5. Process Flow of Authenticating the Context Provider

4.3.2. Authenticating the Context Recipient

An application, defined as a context recipient (CR1), should be authenticated when it requires certain contextual information, and the type of information should be checked so that only that specific information is shared. The Cerberus authentication protocol (Pantsar-Syväniemi et al., 2010) with enhancements is used to perform the following process and is elaborated by Figure 6:

1. **CR1** sends its digital certificate and intends to communicate with **TGS** through **AAS**.
2. **AAS** creates $\mathbf{K_{CR1, TGS}}$ upon authentication of the certificate sent by **CR1,** then $\mathbf{K_{CR1, TGS}}$ is a common key between **TGS** and **CR1** in {{ $K_{CR1, TGS}$, nonce}e_{CR1} }d_{AAS} $\mathbf{K_{TGS}}$ is a private key of **TGS** in Ticket: {TGT = Context Receiver (CR)1_id, $K_{CR1, TGS}$, LIFE}K_{TGS}
3. **CR1** sends **CR1_id** and **B1** to only **TGS** after verifying and decrypting its ticket, where **CR1** is interested in broker **B1**. **AAS** sends Ticket and Authenticator: {auth=TimeStamp}$K_{CR1,TGS}$ to **TGS**.
4. Authenticator is decrypted by **TGS** after **TGS** decrypts its own ticket, then **TGS** creates $\mathbf{K_{CR1, B1}}$ and transmits it to **CR1,** as shown below: {B1, $K_{CR1, B1}$, nonce2}$K_{CR1, TGS}$, where $K_{CR1, B1}$ is a common key used by **CR1** and **B1**. **B1's** Ticket: **{{**Tb1=CR1_id, $K_{CR1,B1}$, life **}**$\boldsymbol{e_{B1}}$**}**$\boldsymbol{d_{TGS}}$
5. **B1** receives a ticket from **CR1** to perform decryption, as shown below: {{Tb1=CR1_id, $K_{CR1,B1}$, life }e_{B1}}d_{TGS}, this is from **TGS**. {auth=TimeStamp}$K_{CR1,B1}$ as Authenticator
6. After decryption of **B1's** Ticket, **B1** responds with the following:

$\{K_{B1group}$, TimeStamp+1$\}K_{CR1,B1}$. This is the current group key of B1, and all the context recipients are associated with it. Finally, **CR1** decrypts $\{K_{B1group}$, TimeStamp+1$\}K_{CR1, B1}$ to deduce the current group key of B1. Then, communication can start between **B1** and **CR1**. If **CR1** wants to communicate with broker **B2,** then it will start the above process from Step 3.

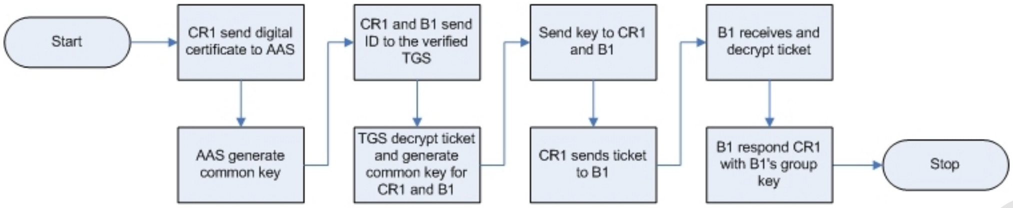

Start
CR1 send digital certificate to AAS
AAS generate common key
CR1 and B1 send ID to the verified TGS
TGS decrypt ticket and generate common key for CR1 and B1
Send key to CR1 and B1
CR1 sends ticket to B1
B1 receives and decrypt ticket
B1 respond CR1 with B1's group key
Stop

Figure 6. Process Flow of Authenticating the Context Recipient

4.3.3. Secure Link Between Context Provider and Context Recipient

The confidentiality services are applied to the contextual information by using symmetric key cryptography. In addition, to guarantee the integrity of contextual information, a hash function and digital signature are used as graphically presented in Figure 7.

Given that **S1** attracts **CR1** and **CR2** (all have been authenticated) when **S1** creates context information (CI), then it is transferred as follows:

1. **B1 receives from S1**:
   i. **C ={context information}**$\boldsymbol{K_{S1B1}}$
   ii. **I =h (C,** $\boldsymbol{K_{S1B1}}$**)**
2. **B1** implements hash function toward $\boldsymbol{K_{S1B1}}$ and **C** with CI**,** further with **I** the outcome is compared, if they are equal, the **C** will be decrypted by **B1** and sent to **CR1** and **CR2** as follows:
   i. **C′ = { CI }**$\mathbf{K_{B1group}}$
   ii. **I′ = {h (C')}**$\boldsymbol{d_{B1}}$

Finally, if I′ is verified by **CR1** and **CR2**, the hash function will be utilized toward **C'**, in that $\mathbf{K_{B1group}}$ will be used to decrypt **C′**.

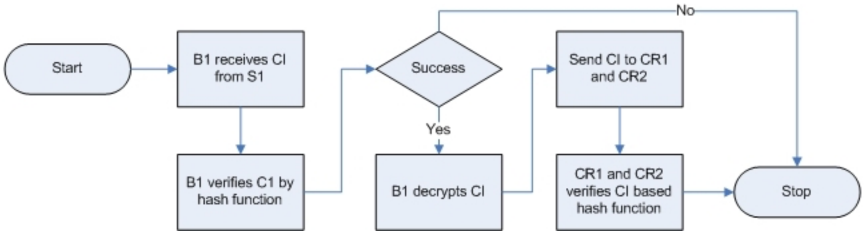


Figure 7. Process Flow of maintaining Secure Link Between Context Provider and Context Recipient

4.3.4. Group Key Update

Suppose context recipients **CR1, CR2** and **CR3** receive information from broker **B1**. That is, all the context recipients have the group key **B1**. If any recipient becomes nonfunctional, the group key remains easily retrievable. To avoid this issue, the broker keeps track of all the recipients in a group. If any recipient becomes nonfunctional, the broker regenerates the key for that particular group, encrypts it using a distinct shared key for each context recipient, and separately distributes the new group key among the other recipients, as shown in detail below:

B1 sends the CI to **CR1, CR2,** and **CR3**. The group key **B1** is changed from $\mathbf{K_{B1group}}$ to $\mathbf{K_{B1groupNEW}}$ only when **B1** finds out that **CR2** is no longer functional. After that, **CR1** and **CR3** will receive new keys from **B1** as follows, then they will decrypt and use them.

1) Key to CR1: $\{\mathbf{K_{B1groupNEW}}\}\mathbf{K_{CR1,B1}}$

2) Key to CR3: $\{\mathbf{K_{B1groupNEW}}\}\mathbf{K_{CR3,B1}}$

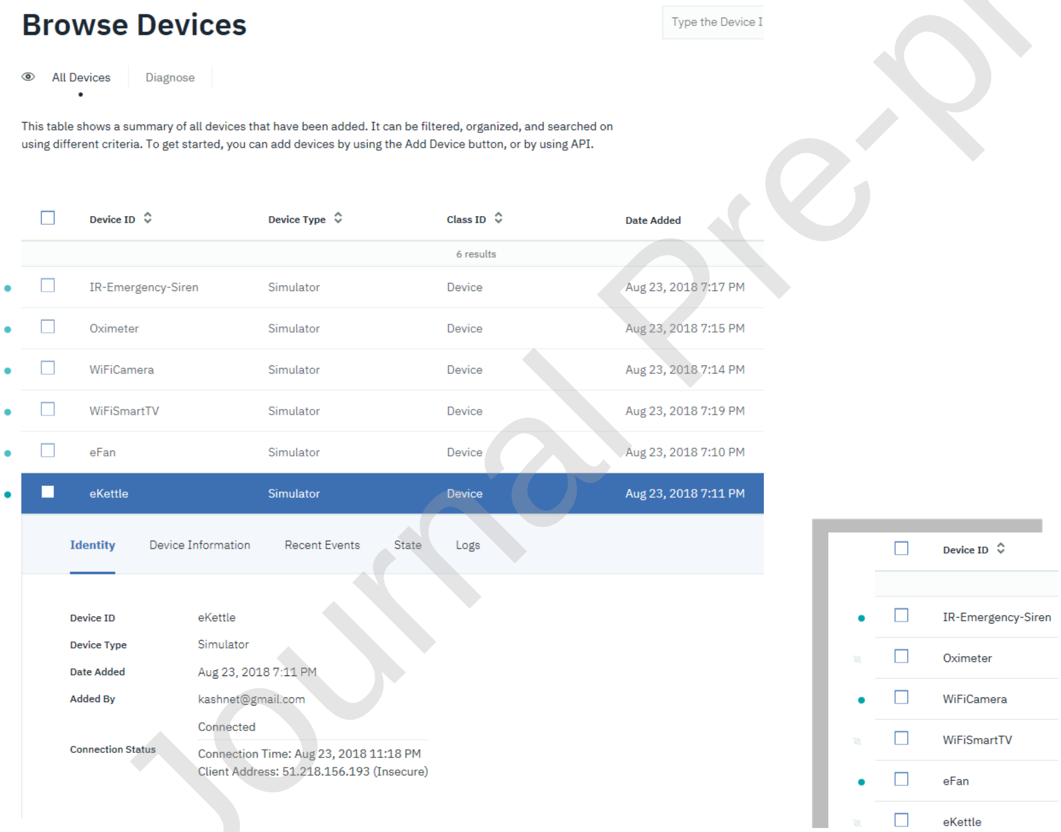
4.4. Characteristics of the Proposed System

- **Flexible association:** The publisher and subscriber model ensures a flexible association between the publisher and subscriber. The publisher (sensor) posts information while ignoring the recipients, and the subscriber (context recipient) shows its interest in the specific type of information (context) while ignoring the publisher.
- **Scalable:** The system achieves scalability by executing parallel operations to avoid tiring the publishers.
- **Dynamic:** Sensors can be authenticated and allowed to enter the system dynamically.
- **Integrity:** Digital signature and hash function encryption are utilized to make sure the contextual data experiences no loss of integrity.
- **Validation:** The powerful Cerberus (M. Chen et al., 2014) algorithm with extensions is used for authentication.
- **Confidentiality:** Symmetric key encryption is used to guarantee the confidentiality and rapid communication of contextual information.
- **Lightweight:** The protocol can be implemented using primitive security operations that are supported by most existing devices used for IoT support.

5. Implementation

An implementation that demonstrates all the security features in a pervasive computing environment is required to evaluate the proposed platform. The implementation should consider the unique characteristics of pervasive environments, such as dynamic operation, and data sensitivity and privacy. It should also provide support for smart environments that require extra safety assurance, including trusted context producers and customers. In addition, the implementation should support the authentication of context providers and recipients. Consequently, the framework can ensure the privacy of data flow between components, as demonstrated below. The cloud environment is adopted for user authentication in the cloud user layer to implement access control and role assignment. Six devices are enabled in the platform, and how these devices can be configured and used within the secure pervasive environment are shown for illustration purposes.

To implement the platform correctly, data from all devices, including data concerning device movement and location, are considered. For experimental purposes, the testbed is deployed, as shown in Figure 4 and Figure 8 with six connected devices. If the devices are active and can send real-time data packets (Figure 9) then they are labeled in green; otherwise, they are labeled in gray, and the status is shown as *disconnected*. Figure 10 demonstrates how secure connectivity is configured according to the policy.



Browse Devices

Type the Device I

All Devices | Diagnose

This table shows a summary of all devices that have been added. It can be filtered, organized, and searched on using different criteria. To get started, you can add devices by using the Add Device button, or by using API.

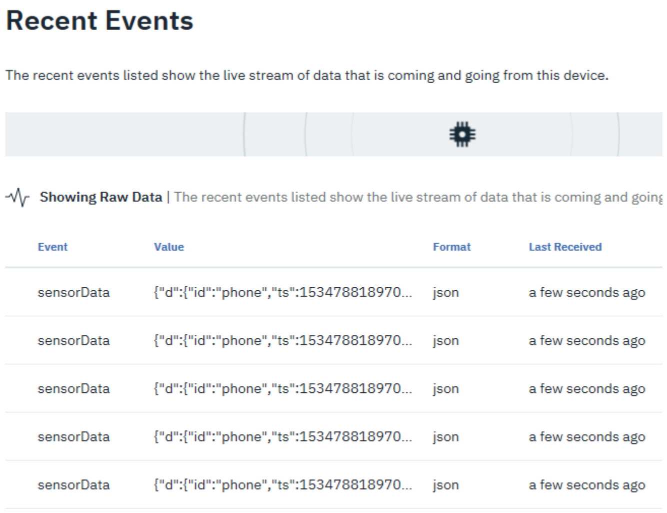
	□	Device ID	Device Type	Class ID	Date Added
				6 results	
●	□	IR-Emergency-Siren	Simulator	Device	Aug 23, 2018 7:17 PM
●	□	Oximeter	Simulator	Device	Aug 23, 2018 7:15 PM
●	□	WiFiCamera	Simulator	Device	Aug 23, 2018 7:14 PM
●	□	WiFiSmartTV	Simulator	Device	Aug 23, 2018 7:19 PM
●	□	eFan	Simulator	Device	Aug 23, 2018 7:10 PM
●	■	eKettle	Simulator	Device	Aug 23, 2018 7:11 PM

Identity | Device Information | Recent Events | State | Logs

Device ID	eKettle
Device Type	Simulator
Date Added	Aug 23, 2018 7:11 PM
Added By	kashnet@gmail.com
Connection Status	Connected Connection Time: Aug 23, 2018 11:18 PM Client Address: 51.218.156.193 (Insecure)

	□	Device ID
●	□	IR-Emergency-Siren
	□	Oximeter
●	□	WiFiCamera
	□	WiFiSmartTV
●	□	eFan
	□	eKettle

Figure 8. The status of six connected devices with green as connected and gray as offline



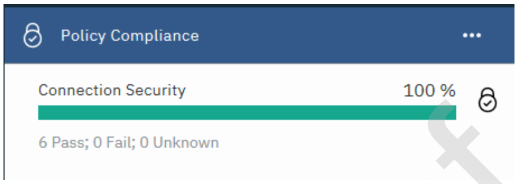
Recent Events

The recent events listed show the live stream of data that is coming and going from this device.

Showing Raw Data | The recent events listed show the live stream of data that is coming and going

Event	Value	Format	Last Received
sensorData	{"d":{"id":"phone","ts":153478818970...	json	a few seconds ago
sensorData	{"d":{"id":"phone","ts":153478818970...	json	a few seconds ago
sensorData	{"d":{"id":"phone","ts":153478818970...	json	a few seconds ago
sensorData	{"d":{"id":"phone","ts":153478818970...	json	a few seconds ago
sensorData	{"d":{"id":"phone","ts":153478818970...	json	a few seconds ago

Figure 9. Events received by every device in real time



Policy Compliance

Connection Security 100 %

6 Pass; 0 Fail; 0 Unknown

Figure 10. The policy compliance passed by all six connected devices

The devices are configured and placed according to the scenario for which the details are as follows:

Device Type: iPhone X Model MQC22J/A

Simulator Device ID: IR-Emergency-Siren

Authentication Token: IRSirenAuth

This device is placed near to roof to act as a siren, and its movement is still with higher Y-axis value due to height, as shown in Figure 11a.

Device Type: iPad model MLYJ2LL/A

Simulator Device ID: eKettle

Security Token: eKettleauthtoken

Just like a kettle, the iPad is moved and then put back in the original location as shown in Figure 11d over Y-axis based on the device longitude value.

Device Type: iPad model MH0W2AE/A

Simulator Device ID: eFan

Security Token: eFanauthtoken

This mobile device is placed very near to the roof, and its longitude value is shown over Y-axis in Figure 11b.

Device Type: iPad model MNV72LL/A

Simulator Device ID: WiFiCamera

Security Token: WiFiCamera

This device is placed in a cupboard near the roof, and its longitude value is shown over Y-axis in Figure 11e. The outputs in the graph Figure 11b and Figure 11e generated from device IDs eFan and WiFiCamera are similar because of the same models.

Device Type: Mobile Samsung Model SM-G950FD

Simulator Device ID: Oximeter

Authentication Token: Oximeter

The mobile device as an oximeter is placed on the table and moved a little, as shown in Figure 11f.

Device Type: SmartTV Sony Bravia KDL-48W600B

Simulator Device ID: WiFiSmartTV

Authentication Token: SmartTVAuthToken

Several aspects of the devices were considered to achieve the correct configuration. For example, the SmartTV does not contain sensors such as an accelerometer, gyroscope, or compass. Therefore, from the device type SmartTV, the recipient receives a data packet that includes the same value of longitude continuously, as shown in Figure 11c. The framework allows the user to browse a multitude of custom recipes to configure and connect the devices based on the Watson IBM IoT Platform (IoT, 2018) and IBM Cloud Developer Tools (Cloud, 2018). This allows the user to expand on the basic service and consume the device IoT data flow within the applications. Figure 11 shows the sustainable and secure framework output in real time, which provides efficient and secure communication between the cloud and multiple IoT devices.

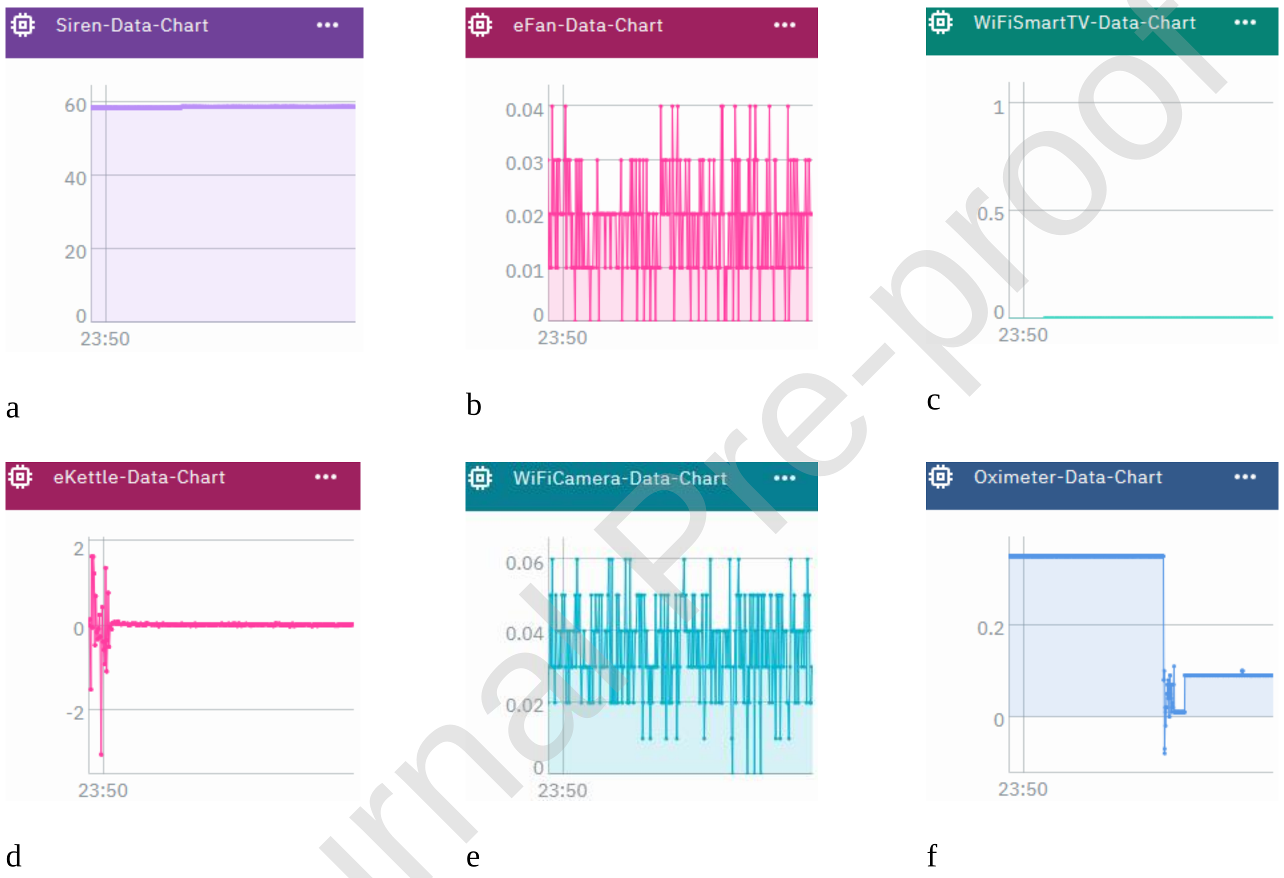
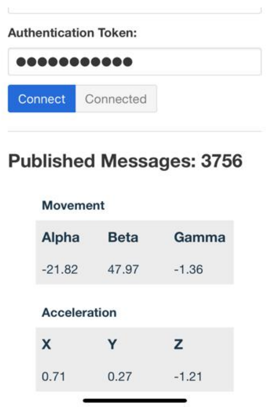


Figure 11. Connected devices in real time

Figure 12 shows an application interface functions on every mobile device with an authentication token. Where Alpha, Beta, and Gamma denotes the geographical position over X-, Y-, and Z-axis. The variation in energy consumption can easily be noticed in Figure 13, it is due to the data transfer is paused, as shown in Figure 14, by the green line that represents the data transfer/upload. The implementation shows that the proposed security framework is practical and can provide a convenient method to support the security of IoT smart devices given their context. One of the key features of the framework is its capability to support multiple devices. Consequently, it can be used within different contexts, such as smart home, industrial, and institutional environments, without redesigning or updating. Another key feature is its lightweight security support, which enables the support of devices with mobility, wearable devices, and even implanted devices with low battery power. Hence, the novel cybersecurity framework provides the necessary infrastructure for security support, which is often neglected.



Authentication Token:

●●●●●●●●●●●

Published Messages: 3756

Movement

Alpha	Beta	Gamma
-21.82	47.97	-1.36

Acceleration

X	Y	Z
0.71	0.27	-1.21

Figure 12. Mobile interface with authentication token

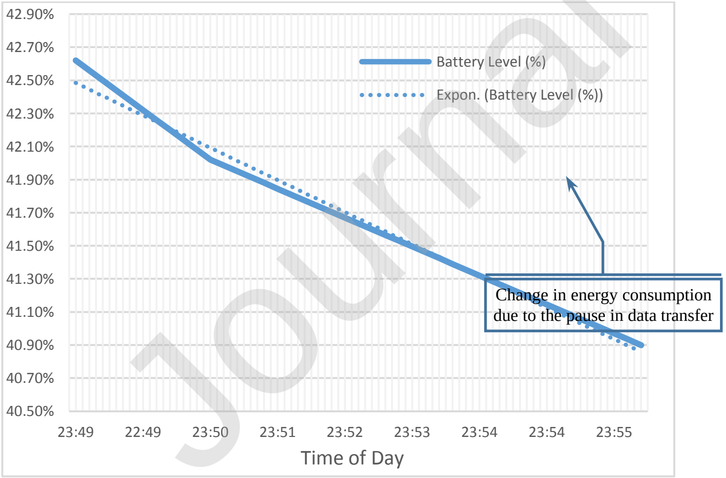


Figure 13. Variation in Energy Consumption

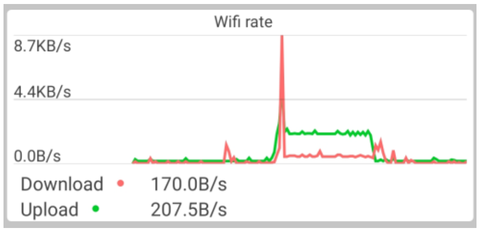


Figure 14. Data Rate vs Time (sec)

6. Conclusion and Future Work

This paper reviews the literature on the smart sustainable cities and pervasive environments' security and the service access control or the context information-based authentication. The major cyber threats which are faced by context-aware pervasive computing environments are also discussed. In this regard, a novel cybersecurity framework that offers authentication for context providers and recipients by using the enhanced Cerberus mechanism is proposed, so as to ensure the integrity of contextual information and its smooth transfer between devices. The symmetric key encryption method is applied to leverage the data confidentiality, and the data integrity is guaranteed by incorporating digital signature-based hash function. The novel system allows the dynamic inclusion of new context providers and recipients after authentication. Moreover, a cloud environment with heterogeneous devices is adopted for user authentication to improve the security of sensors and contextual information. Additionally, the proposed cybersecurity framework is enabled with simple security operations and is therefore lightweight. Finally, the novel security framework is implemented and analyzed in a real cloud environment with six real-time devices that show the efficiency, communication security, minimal energy consumption, scalability of multiple device handling in real time, and policy compliance passed the data privacy of the system in a pervasive IoT environment. Specifically, the minimal energy consumption of about 0.35 % is noted with and without data transfer or while uploading 207.5 B/s to the cloud.

In the future, the proposed framework will be evaluated in an extended scenario, by employing other security techniques and in the presence of malicious or non-self devices. Further experiments will be conducted on the use of this framework within a smart home environment by integrating more features. For instance, considering the power consumption of mobile devices within a smart home and providing scheduling methods for power optimization is an appealing issue in demand-side managed systems.

Declaration of interests

THE AUTHORS DECLARE THAT THEY HAVE NO KNOWN COMPETING FINANCIAL INTERESTS OR PERSONAL RELATIONSHIPS THAT COULD HAVE APPEARED TO INFLUENCE THE WORK REPORTED IN THIS PAPER.

Acknowledgements

The authors extend their appreciation to the Deanship of Scientific Research at the King Saud University for funding this work through research group no (RG-1439-022).

This work is partially supported by the FCT/MCTES through national funds and when applicable co-funded EU funds under the Project UIDB/50008/2020; and by Brazilian National Council for Research and Development - CNPq, via Grant No. 309335/2017-5.

The authors thank the Deanship of Scientific Research and researchers support and services unit (RSSU) at King Saud University for their technical support.

References

Al Ridhawi, I., Otoum, S., Aloqaily, M., Jararweh, Y., & Baker, T. (2020). Providing secure and reliable communication for next generation networks in smart cities. *Sustainable Cities and Society, 56*, 102080. doi:https://doi.org/10.1016/j.scs.2020.102080

Al-Muhtadi, J., Qiang, M., Zeb, K., Chaudhry, J., Saleem, K., Derhab, A., . . . Pasha, M. (2018). A Critical Analysis of Mobility Management Related Issues of Wireless Sensor Networks in Cyber Physical Systems. *IEEE Access, 6*, 16363-16376. doi:10.1109/ACCESS.2018.2812741

Al-Muhtadi, J., Ranganathan, A., Campbell, R., & Mickunas, M. D. (2002). *A flexible, privacy-preserving authentication framework for ubiquitous computing environments.* Paper presented at the Distributed Computing Systems Workshops, 2002. Proceedings. 22nd International Conference on.

Al-Muhtadi, J., Ranganathan, A., Campbell, R., & Mickunas, M. D. (2003). *Cerberus: a context-aware security scheme for smart spaces.* Paper presented at the Pervasive Computing and Communications, 2003.(PerCom 2003). Proceedings of the First IEEE International Conference on.

Al-Rabiaah, S., & Al-Muhtadi, J. (2012). *Consec: Context-aware security framework for smart spaces.* Paper presented at the Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2012 Sixth International Conference on.

Alazab, M., & Tang, M. (2019). *Deep Learning Applications for Cyber Security*: Springer.

Amanullah, M. A., Habeeb, R. A. A., Nasaruddin, F. H., Gani, A., Ahmed, E., Nainar, A. S. M., Imran, M. (2020). Deep learning and big data technologies for IoT security. *Computer Communications, 151*, 495-517. doi:https://doi.org/10.1016/j.comcom.2020.01.016

Azab, A., Alazab, M., & Aiash, M. (2016, 23-26 Aug. 2016). *Machine Learning Based Botnet Identification Traffic.* Paper presented at the 2016 IEEE Trustcom/BigDataSE/ISPA.

Aziz, M. F., Khan, A. N., Shuja, J., Khan, I. A., Khan, F. G., & Khan, A. u. R. A lightweight and compromise-resilient authentication scheme for IoTs. *Transactions on Emerging Telecommunications Technologies, n/a*(n/a), e3813. doi:10.1002/ett.3813

Bahl, P., Han, R. Y., Li, L. E., & Satyanarayanan, M. (2012). *Advancing the state of mobile cloud computing.* Paper presented at the Proceedings of the third ACM workshop on Mobile cloud computing and services.

Benzaid, C., Lounis, K., Al-Nemrat, A., Badache, N., & Alazab, M. (2016). Fast authentication in wireless sensor networks. *Future Generation Computer Systems, 55*, 362-375. doi:https://doi.org/10.1016/j.future.2014.07.006

Bettini, C., Brdiczka, O., Henricksen, K., Indulska, J., Nicklas, D., Ranganathan, A., & Riboni, D. (2010). A survey of context modelling and reasoning techniques. *Pervasive and Mobile Computing, 6*(2), 161-180.

Bettini, C., & Riboni, D. (2015). Privacy protection in pervasive systems: State of the art and technical challenges. *Pervasive and Mobile Computing, 17*, 159-174.

Bibri, S. E. (2018). The IoT for smart sustainable cities of the future: An analytical framework for sensor-based big data applications for environmental sustainability. *Sustainable Cities and Society, 38*, 230-253. doi:https://doi.org/10.1016/j.scs.2017.12.034

Bouachir, O., Aloqaily, M., Tseng, L., & Boukerche, A. (2020). Blockchain and Fog Computing for Cyberphysical Systems: The Case of Smart Industry. *Computer, 53*(9), 36-45. doi:10.1109/MC.2020.2996212

Braun, T., Fung, B. C. M., Iqbal, F., & Shah, B. (2018). Security and privacy challenges in smart cities. *Sustainable Cities and Society, 39*, 499-507. doi:https://doi.org/10.1016/j.scs.2018.02.039

Chaudhry, J., Ibrahim, A., & Bashir, A. K. (2016). Internet of Threats and Context Aware Security: Part Two. *Newsletter, 2016*.

Chen, G., & Kotz, D. (2000). *A survey of context-aware mobile computing research*. Retrieved from

Chen, M., Wan, J., González-Valenzuela, S., Liao, X., & Leung, V. C. (2014). A Survey of Recent Developments in Home M2M Networks. *IEEE Communications Surveys and Tutorials, 16*(1), 98-114.

Chowdhury, M. A., & Light, J. (2009). *Context-Aware Data Association and Authenticity in Pervasive Healthcare*. Paper presented at the Privacy, Security, Trust and the Management of e-Business, 2009. CONGRESS'09. World Congress on.

Cloud, I. (2018). IBM Cloud Developer Tools. Retrieved from https://github.com/IBM-Cloud/ibm-cloud-developer-tools

Cook, D. J., & Das, S. K. (2012). Pervasive computing at scale: Transforming the state of the art. *Pervasive and Mobile Computing, 8*(1), 22-35.

Covington, M. J., Fogla, P., Zhan, Z., & Ahamad, M. (2002). *A context-aware security architecture for emerging applications.* Paper presented at the Computer Security Applications Conference, 2002. Proceedings. 18th Annual.

Covington, M. J., Long, W., Srinivasan, S., Dev, A. K., Ahamad, M., & Abowd, G. D. (2001). *Securing context-aware applications using environment roles.* Paper presented at the Proceedings of the sixth ACM symposium on Access control models and technologies.

Covington, M. J., Moyer, M. J., & Ahamad, M. (2000). *Generalized role-based access control for securing future applications*. Retrieved from

Djellali, B., Lorenz, P., Belarbi, K., & Chouarfia, A. (2014). Security Model for Pervasive Multimedia Environment. *Journal of Multimedia Information System, 1*(1), 23-43. Retrieved from https://hal.archives-ouvertes.fr/hal-01167505

Dobkin, D. M. (2012). *The rf in RFID: uhf RFID in practice*: Newnes.

Farivar, F., Haghighi, M. S., Jolfaei, A., & Alazab, M. (2020). Artificial Intelligence for Detection, Estimation, and Compensation of Malicious Attacks in Nonlinear Cyber-Physical Systems and Industrial IoT. *IEEE Transactions on Industrial Informatics, 16*(4), 2716-2725. doi:10.1109/TII.2019.2956474

Gao, K., Zhu, Y., Gong, S., Tan, H., & Zhou, G. (2016). Research on social network discovery algorithm in pervasive sensing environment. *Concurrency and Computation: Practice and Experience, 28*(15), 4093-4106.

Gomez, C., Oller, J., & Paradells, J. (2012). Overview and evaluation of bluetooth low energy: An emerging low-power wireless technology. *Sensors, 12*(9), 11734-11753.

Good, T., & Benaissa, M. (2013). A holistic approach examining RFID design for security and privacy. *The Journal of Supercomputing, 64*(3), 664-684. doi:10.1007/s11227-010-0497-9

Hansmann, U., Merk, L., Nicklous, M. S., & Stober, T. (2013). *Pervasive computing handbook*: Springer Science & Business Media.

Hu, J., & Weaver, A. C. (2004). *A dynamic, context-aware security infrastructure for distributed healthcare applications.* Paper presented at the Proceedings of the first workshop on pervasive privacy security, privacy, and trust.

IoT, W. (2018, April 11, 2018). Device Identity provisioning in Watson IoT with WISeKey's Managed PKI API. Retrieved from https://developer.ibm.com/recipes/tutorials/device-identity-provisioning-with-wisekeys-managed-pki-api/

Jararweh, Y., Otoum, S., & Ridhawi, I. A. (2020). Trustworthy and sustainable smart city services at the edge. *Sustainable Cities and Society, 62*, 102394. doi:https://doi.org/10.1016/j.scs.2020.102394

Kamal, M. S., Parvin, S., Saleem, K., Al-Hamadi, H., & Gawanmeh, A. (2017, 2017). *Efficient low cost supervisory system for Internet of Things enabled smart home.* Paper presented at the 2017 IEEE International Conference on Communications Workshops, ICC Workshops 2017.

Khattak, H. A., Shah, M. A., Khan, S., Ali, I., & Imran, M. (2019). Perception layer security in Internet of Things. *Future Generation Computer Systems, 100*, 144-164. doi:https://doi.org/10.1016/j.future.2019.04.038

Khurshid, A., Khan, A. N., Khan, F. G., Ali, M., Shuja, J., & Khan, A. u. R. (2019). Secure-CamFlow: A device-oriented security model to assist information flow control systems in cloud environments for IoTs. *Concurrency and Computation: Practice and Experience, 31*(8), e4729. doi:10.1002/cpe.4729

Kim, K., Kim, I., & Lim, J. (2017). National cyber security enhancement scheme for intelligent surveillance capacity with public IoT environment. *The Journal of Supercomputing, 73*(3), 1140-1151. doi:10.1007/s11227-016-1855-z

Kim, S., Lee, S., Kim, J., & Yoon, Y.-I. (2017). MRTensorCube: tensor factorization with data reduction for context-aware recommendations. *The Journal of Supercomputing*. doi:10.1007/s11227-017-2002-1

Kim, S.-K., Kim, B.-G., & Min, B.-J. (2015). Reducing Security Overhead to Enhance Service Delivery in Jini IoT. *International Journal of Distributed Sensor Networks, 11*(11), 205793. doi:10.1155/2015/205793

Laufs, J., Borrion, H., & Bradford, B. (2020). Security and the smart city: A systematic review. *Sustainable Cities and Society, 55*, 102023. doi:https://doi.org/10.1016/j.scs.2020.102023

Lee, I. (2019). The Internet of Things for enterprises: An ecosystem, architecture, and IoT service business model. *Internet of Things, 7*, 100078. doi:https://doi.org/10.1016/j.iot.2019.100078

Li, F., Nastic, S., & Dustdar, S. (2012). *Data quality observation in pervasive environments.* Paper presented at the Computational Science and Engineering (CSE), 2012 IEEE 15th International Conference on.

Lu, Y., & Liu, Y. (2012). Pervasive location acquisition technologies: Opportunities and challenges for geospatial studies. *Computers, Environment and Urban Systems, 36*(2), 105-108.

Neuman, B. C., & Ts'o, T. (1994). Kerberos: An authentication service for computer networks. *IEEE Communications Magazine, 32*(9), 33-38.

Pallapa, G., Das, S. K., Di Francesco, M., & Aura, T. (2014). Adaptive and context-aware privacy preservation exploiting user interactions in smart environments. *Pervasive and Mobile Computing, 12*, 232-243.

Pantsar-Syväniemi, S., Simula, K., & Ovaska, E. (2010). *Context-awareness in smart spaces.* Paper presented at the Computers and Communications (ISCC), 2010 IEEE Symposium on.

Perera, C., Zaslavsky, A., Christen, P., & Georgakopoulos, D. (2014). Context aware computing for the internet of things: A survey. *IEEE Communications Surveys & Tutorials, 16*(1), 414-454.

Rahimi, M. R., Ren, J., Liu, C. H., Vasilakos, A. V., & Venkatasubramanian, N. (2014). Mobile cloud computing: A survey, state of art and future directions. *Mobile Networks and Applications, 19*(2), 133-143.

Rahman, M. A., Hossain, M. S., Hassanain, E., & Muhammad, G. (2018). Semantic Multimedia Fog Computing and IoT Environment: Sustainability Perspective. *IEEE Communications Magazine, 56*(5), 80-87. doi:10.1109/MCOM.2018.1700907

Saleem, K., Derhab, A., Al-Muhtadi, J., & Shahzad, B. (2015). Human-oriented design of secure Machine-to-Machine communication system for e-Healthcare society. *Computers in Human Behavior, 2015*(51), 977–985. doi:10.1016/j.chb.2014.10.010

Saleem, K., Fisal, N., & Al-Muhtadi, J. (2014). Empirical studies of bio-inspired self-organized secure autonomous routing protocol. *IEEE Sensors Journal, 14*(7), 2232-2239. doi:10.1109/JSEN.2014.2308725

Shankar, K., Camp, L. J., Connelly, K., & Huber, L. (2012). Aging, privacy, and home-based computing: Developing a design framework. *IEEE Pervasive Computing, 11*(4), 46-54.

Tang, F., & Li, M. (2012). Context-adaptive and energy-efficient mobile transaction management in pervasive environments. *The Journal of Supercomputing, 60*(1), 62-86. doi:10.1007/s11227-009-0277-6

Vinayakumar, R., Alazab, M., Srinivasan, S., Pham, Q., Padannayil, S. K., & Simran, K. (2020). A Visualized Botnet Detection System Based Deep Learning for the Internet of Things Networks of Smart Cities. *IEEE Transactions on Industry Applications, 56*(4), 4436-4456. doi:10.1109/TIA.2020.2971952

Weiser, M. (1999). The computer for the 21st century. *Mobile Computing and Communications Review, 3*(3), 3-11.

Yang, G. (2016). *Research and Design of Smart Home System Based on 433MHz RF Communication.* (Master Thesis). CNKI.

Yaqoob, I., Ahmed, E., Rehman, M. H. u., Ahmed, A. I. A., Al-garadi, M. A., Imran, M., & Guizani, M. (2017). The rise of ransomware and emerging security challenges in the Internet of Things. *Computer Networks, 129*, 444-458. doi:https://doi.org/10.1016/j.comnet.2017.09.003

Yaseen, M., Saleem, K., Orgun, M. A., Derhab, A., Abbas, H., Al-Muhtadi, J., . . . Rashid, I. (2018). Secure Sensors Data Acquisition and Communication Protection in eHealthcare: Review on State of the Art. *Telematics and Informatics, 35*(4), 702-726. doi:https://doi.org/10.1016/j.tele.2017.08.005

Yu, P., Ma, X., Cao, J., & Lu, J. (2013). Application mobility in pervasive computing: A survey. *Pervasive and Mobile Computing, 9*(1), 2-17.