Developing an Integrated Cloud-based Framework for Securing Dataflow of Wireless Sensors

Habibah AL-Harbi[1], Khalil H. A. Al-Shqeerat[2], Ibrahim S. Alsukayti[3]
Computer Science Department, Qassim University, Qassim, Saudi Arabia

***Abstract*—Cloud computing environment has been developed rapidly and becomes a popular trend in recent years. It provides on-demand services to several applications with access to an unlimited number of resources such as servers, storage, networks. Wireless Sensor Network, on the other hand, has been enormously progressing in various applications and producing a considerable amount of sensor data. Sensor networks are based on a group of interconnected small size sensor nodes that can be distributed over different geographical areas to observe environmental and physical phenomena. Nevertheless, it has limitations concerning power, storage, and scalability that need to be addressed adequately. Integrating wireless sensor networks with cloud computing can overcome these problems. Cloud computing provides a more secure and high available platform for effective management of sensor data. This paper proposes a framework to secure the dataflow of sensor devices from wireless sensor networks to cloud computing using an integrated environment. The framework presents an authentication scheme to validate the identity of sensor devices connected to the cloud environment. Furthermore, it provides secure environments with high availability and data integrity.**

Keywords*—Framework; security; wireless sensor; cloud computing; data integrity; availability

I. Introduction

Cloud computing is a flourishing technology that has appeared in the commercial sector of information technology [1]. Its paradigm can make computer software more attractive as a service. It eliminates the need for setting up a large number of physical devices or operating the infrastructure while requiring experts and technical support [2]. Cloud Computing is Internet-based computing where resources allocations are shared. It provides software and information to the computer and all other devices on demand as requested. IBM declared that cloud computing is a novel model for using and delivering several IT-based services [3]. It allows transparent access to cloud services without the need for recognizing the underlying technologies or implementation.

Some common issues, such as security, pricing models, scheduling, and integration of different applications, are resolved using cloud computing [4]. Moreover, the accessibility of essential resources such as memory, bandwidth, storage, servers, and networks are supported by cloud computing. Furthermore, the "pay as you use" services are considered. Fig. 1 shows the general architecture of cloud computing.

Security and privacy are the most critical challenges in cloud computing [5]. Security relies on a group of techniques that protect sensitive data from the vulnerable attacks and ensure data integrity, authentication, and confidentiality [6]. Privacy ensures that users can control their sensitive data.

The security issues cover several areas, including operating systems, networks, virtualization, databases, resource scheduling, load balancing, and memory management [7]. For example, the network that links systems or applications must be secure.

On the other hand, Wireless Sensor Networks (WSNs) are self-organizing networks that implemented vastly in various applications. WSNs consist of a group of spatial distributed multifunctional sensors [8]. These sensors have the capabilities of transmitting or monitoring significant environmental or physical situations such as temperature, humidity, pressure, and sound. Interconnected sensors sense the surrounding environment and transfer sensed data to master or sink nodes [9]. Fig. 2 shows the general structure of WSNs.

WSNs become a significant trend in different domains in industrial, commercial, governmental, entertainment, medical, military, transportation, city management, smart spaces, and environmental applications [10]. However, there are still several security issues due to limitations in regards to communication and interconnectivity. These issues include confidentiality, integrity, authentication, availability, and freshness of sensor data [11]. Besides, WSNs face many challenges as the resources of sensor nodes suffer from low power, cost, storage capacity, and bandwidth availability.

WSNs are exposed to various threats and attacks where the attacker can access the sensor node. WSNs security is associated with some main requirements, including integrity, authentication, freshness, confidentiality, and availability [12]. Therefore, an integrated platform is needed to secure data sent from WSNs and to ensure security requirements for cloud computing.

This paper presents a secure framework that provides a secure dataflow from sensor nodes to cloud computing in an integrated environment. An effective authentication scheme is developed to validate the identity of sensor devices using their sensor serial number and geographic location. Moreover, efficient encryption techniques are used to secure data sent from sensors to cloud computing, such as public key encryption (RSA) and symmetric key encryption (AES). In addition, data integrity is addressed in the framework using the hashing technique. Furthermore, Scyther analyzer protocol tool is used to analyze and validate the proposed security solutions.

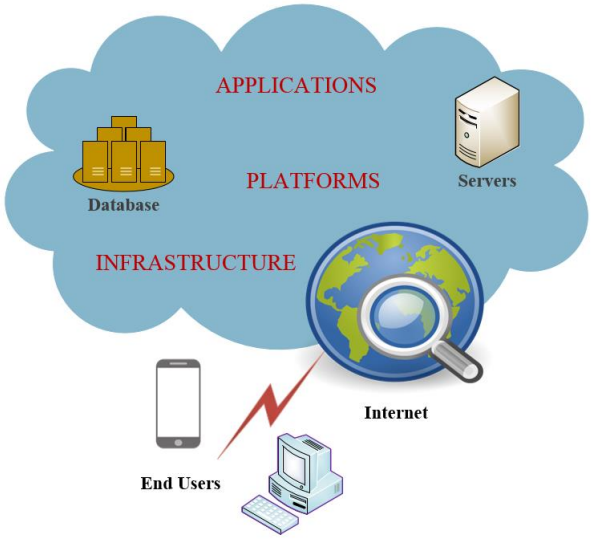

Fig. 1. General Architecture of Cloud Computing.

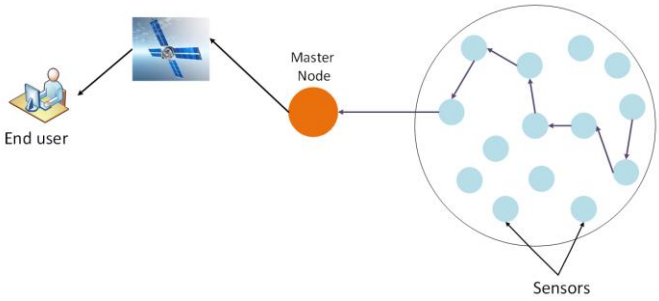

Fig. 2. General Structure of WSNs.

The rest of the paper is organized as follows: In Section II, a literature review is presented. Section III introduces the architecture of the proposed integrated framework. The design and analysis of a secure framework are presented in Section IV. An implementation of the proposed framework is presented in Section V. Finally, Section VI discusses and analyzes the implementation results of the proposed framework.

II. Literature Review

This section presents an overview of some related approaches that have been proposed for integrating WSNs with cloud computing.

In [3], the authors proposed a new framework for wireless sensor networks integration with a Cloud computing model. This framework shows how the data is shifted from WSN to cloud computing. Reliability and availability of wireless sensor networks will be promoted by applying cloud computing. The proposed system has its useful applications and significant role in the medical sciences field.

A method for processing the sensory data in wireless sensor networks and mobile cloud computing (WSN–MCC) integration is introduced in [13]. It addresses the critical issues concerning WSN–MCC integration. Also, the authors proposed a framework for a novel sensory data processing, for transmitting available sensory data to the users in a fast, reliable, and secure manner. Analytical and experimental results are presented to explain that the proposed framework can improve the network lifetime, the storage requirement, the security and monitoring performance of WSNs, and the security of the transmitted sensory data.

Authors in [14] proposed a flexible, secure scheme for data-centric applications in cloud computing based scenario. They have taken healthcare application as a case study to show the performance analysis of the proposed security method. They have tried to eliminate potential security threats and guarantee fast and flexible security solutions to the fundamental security requirements.

In [15], the authors proposed a Sensor-cloud infrastructure to provide a flexible platform which shares a vast amount of sensor data from various applications. They focus on the processing of sensor data with the collaboration of WSN and cloud securely to access the sensor cloud resources using authentication and access control. The authors propose a new data processing framework to integrate wireless sensor networks with cloud computing. Furthermore, in sensor-cloud infrastructure, the identity-based cryptography is proposed to facilitate key distribution and authentication.

A unique based framework is introduced in [16] for integrating body area network with cloud computing. This framework uses the concept of publish/subscribe (pub/sub) broker. The methodology of this framework is implemented by transmitting the gathered sensitive data from the patient to the web application on cloud computing. The simulation result indicated that internal attack detection works well.

The authors in [17] proposed an integrated architecture of wireless sensor networks and cloud computing in an agricultural environment. This paper aims to simplify the shifting of data from wireless sensor networks to cloud computing. The integrated architecture consists of three levels including Sensing Data Level, Cloud Service Level (SAAS, PAAS, and IAAS) and Control Level. Furthermore, the development of the agricultural environment management (AEMS) system is based on the architecture of WSNs and cloud computing. The testing results of temperature and humidity data that are collected from wireless sensors and AEMS are presented.

Since the embedded systems are limited in resources, storage, and computing, a framework in [18] is presented to extend the local resources of these embedded systems. Scalability and high availability are provided in this framework. In order to manage the use of cloud computing to minimize computation cost and execution time, they implement a scheduling algorithm. If the task requires high computation, it is applied in cloud computing, while in case of medium and low computations, they are implemented in the local servers and embedded systems. Moreover, windows azure services are used to implement this framework. They used response time and throughput metrics in order to evaluate the performance of the implemented framework.

In order to provide confidentiality, authenticity, integrity, and privacy of wireless sensor networks, the authors in [19] have proposed a framework to secure data delivery in WSNs.

They use private cloud computing to increase computation resources and storage. Also, to simplify data retrieval, Elliptic curve cryptography is used in this framework for encryption and decryption of collected data. To prevent some attacks and secure the framework, the authors suggest some detection and security measurement. Moreover, this framework is implemented in the medical field. Finally, they validate the performance and robustness of the framework by using performance and security analysis.

III. Proposed Framework Architecture

The proposed framework securely integrates wireless sensor networks environment with cloud computing. Fig. 3 shows the general architecture of the proposed framework to secure dataflow from WSNs environment to cloud computing platforms. The framework consists of the following several components: sensor nodes, sensor gateway, integration unit, Management unit, access policy unit, and cloud platforms (databases, servers, and processing unit).

The sensor nodes are distributed in different fixed geographical locations, and each sensor is identified by a unique identifier. A combination ID consists of serial number and current geographical location where the sensor node is located is used to identify each sensor node. Latitude and longitude values of sensor's location are concatenated to the sensor id as salt, and then the salted id is hashed to make the dictionary attack more difficult.

The sensed data is passed from sensor nodes to the sensor gateway directly. The gateway collects sensed data from sensors, computes the hash code of collected data, and then encrypts both data and hash using a symmetric-key encryption method.

The integration unit is used as a temporary buffer and to forward the encrypted data from the sensor network environment to cloud computing. After the data is passed to cloud computing, the cloud controller will rely on the policy management unit to authenticate the identity of each sensor node. If the sensor device is identified, the sensed data will be forwarded to the application server in cloud computing for decrypting received data and applying data integrity technique.

The load balancer distributes the workload stream across multiple application and database servers to improve overall availability and achieve high performance.

On the right side of Fig. 3, the rectangle components show the security requirements in each phase during transmitting data from sensor devices to cloud computing. The sensor data is protected by authentication, confidentiality, and integrity techniques at the sensor gateway. During the transmission of sensor data from a wireless sensor network to cloud computing, a secure connection is provided.

The authentication of sensor nodes is performed by the application server on the cloud side to check their identity. Furthermore, availability, confidentiality, and integrity of data will be provided in cloud computing.

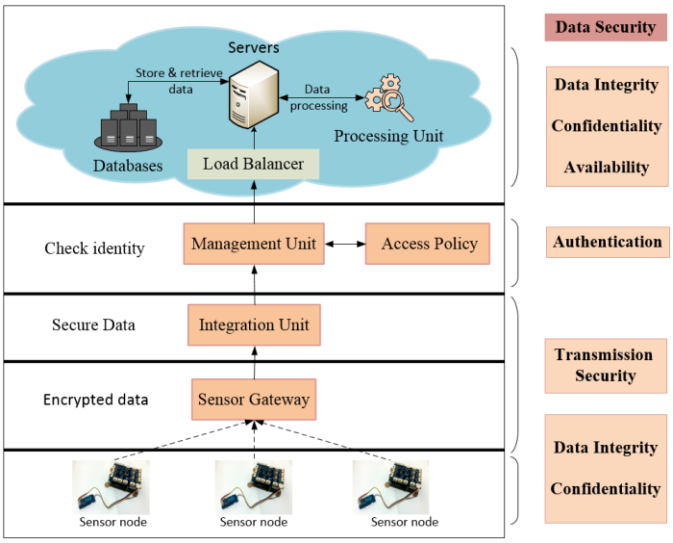

Fig. 3. An Integrated Framework Architecture.

In the suggested framework, security requirements are provided as follows:

- Authentication is provided for each sensor device using its serial number and current geographical location.
- Confidentiality is the encryption and decryption of data transmitted from sensors to cloud computing.
- Data Integrity is provided by using a hashing technique to ensure that the transmitted data have not been modified during transmission.
- Availability is guaranteed by cloud computing, which provides access to an unlimited number of resources.

IV. Framework Design and Analysis

A. Operations Design

This section demonstrates the primary operations of the proposed framework. The serial number and geographical location of the sensor device are used to validate the identity of each device using the authentication scheme explained in the next section.

Data encryption is implemented using symmetric key cryptography. In addition, a hashing technique was implemented for data integrity.

Fig. 4 shows a flowchart for the process of sending data at the sensor side. It illustrates how the sensed data moved toward cloud computing.

In the beginning, each master node (gateway) reads its serial number and current geographical location and then attempts to connect to the cloud controller. The sensor device is logged in to cloud computing if it is already registered. Otherwise, it carries out the registration process first. After that the master node collects sensed data from other sensor nodes, and then hashes the sensed data. Symmetric-key encryption algorithm encrypts both hashed and sensed data before transmitting them to cloud computing.

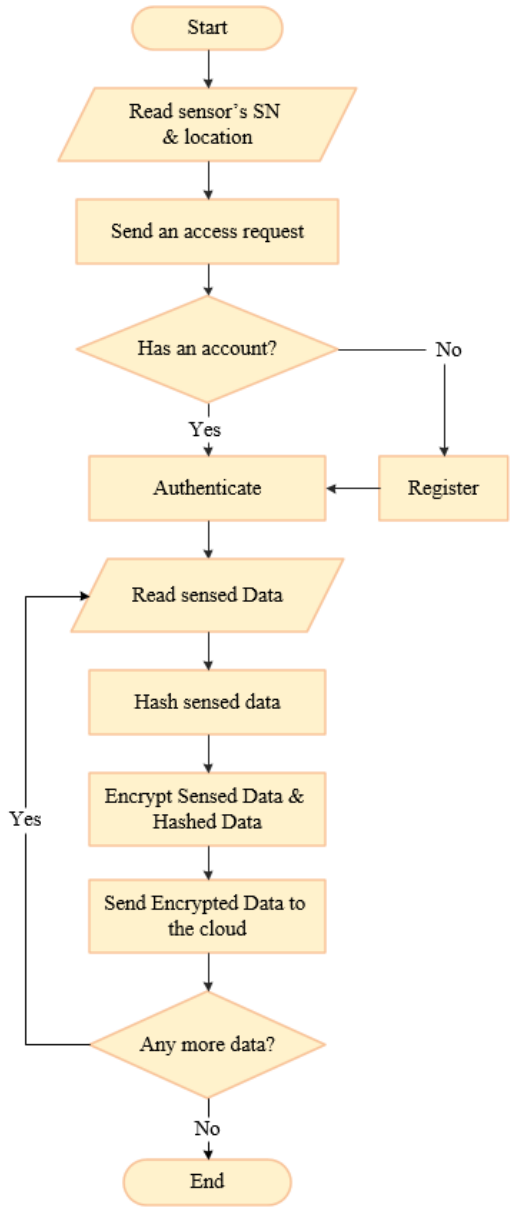

Fig. 4. Flowchart for Sending Data at the Sensor Side.

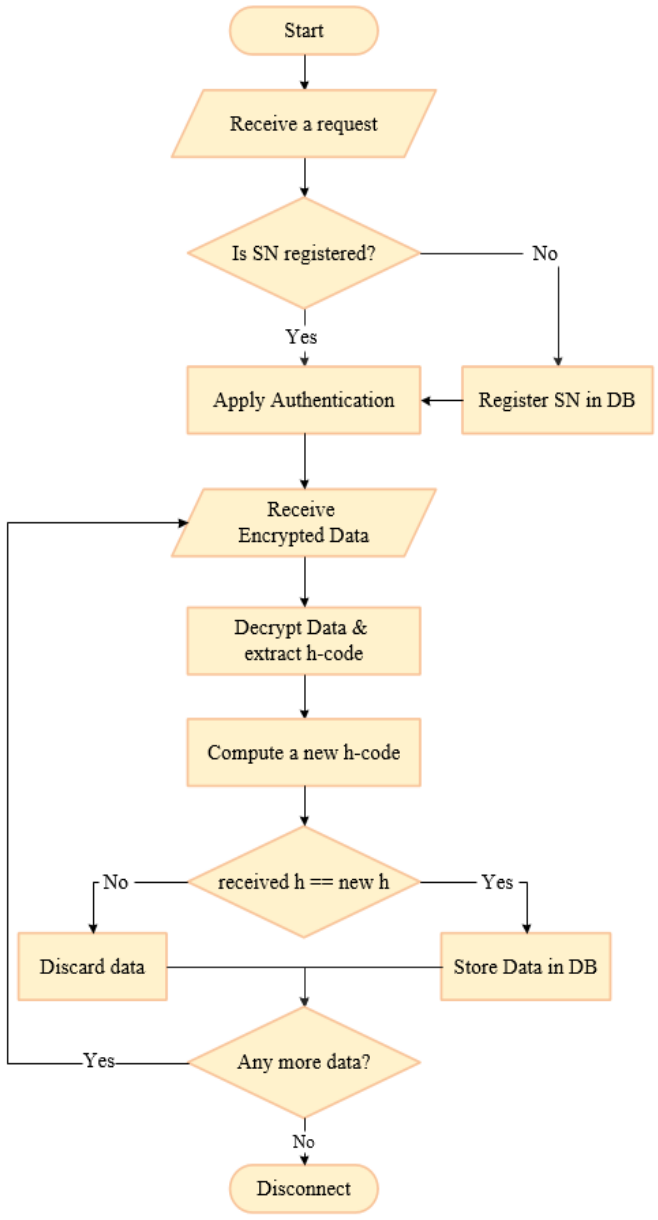

Fig. 5. Flowchart for Receiving Data on the Cloud-Side.

Fig. 5 shows a flowchart for the process of receiving data on the cloud-computing side. Upon the reception of the encrypted data at cloud computing, the data is decrypted using the same symmetric encryption. The cloud controller computes a new hash value of the sensed data and compares it with that sent with sensed data. If they match, it stores data to the database, otherwise, discards it.

B. Authentication Scheme Design

The authentication scheme suggested in [20] has been developed in the framework to authenticate the identity of master sensor nodes (SN) before transmitting data towards the cloud controller (CC).

This proposed scheme includes registration and authentication phases. Fig. 6, and 7 show the sequence diagrams of both registration and authentication phases.

In the beginning, the master node generates a varying time nonce *ni* and sends a connection request to the cloud controller after encrypting it using public-key cryptography method. The encrypted request involves the sensor's serial number *sn* and generated nonce *ni*. Upon receiving a request, the cloud controller checks its log-file, if the sensor node is not registered, the registration phase starts when the cloud controller creates an account for the sensor device as shown in Fig. 6.

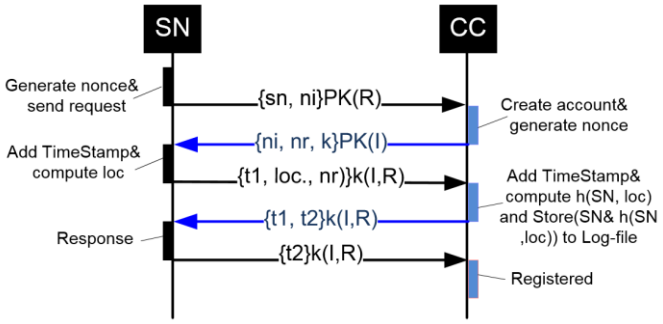

Fig. 6. The Registration Phase.

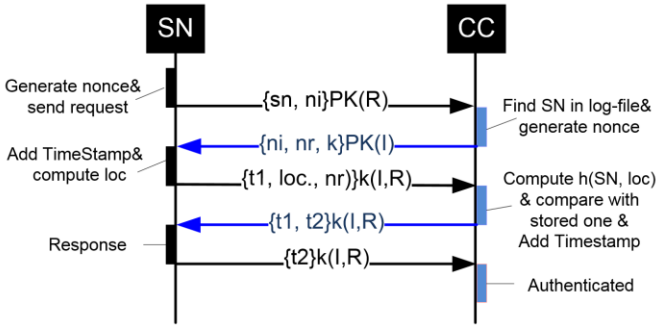

Fig. 7. The Authentication Phase.

While if the sensor node is already registered, the authentication phase starts as shown in Fig. 7.

In the next steps of both phases, the cloud controller generates its nonce *nr*, and then encrypts it with a new session key *k* and sensor's *ni* using the public key of SN before sending the response back to the node. The session key will be used for encrypting any data sent from the SN toward CC. When the SN receives the message sent from the CC, it will decrypt it using its private key and will compare its generated nonce with the nonce comes back from CC. If they are not matched, it will interrupt the connection. The timestamp *t1*, *nr*, and sensor's geographical location *loc* are sent to the CC after encrypting them by symmetric encryption method using the session key *k*. On the cloud side, nonces are compared as same as in the sensor side. If they are not matched, the controller will close the connection.

After that, the CC computes the hash value of concatenated *sn* and *loc* of the SN, store hashed value to the sensor's account and then sends encrypted *t2* with *t1* to the SN. After receiving a response from SN, the CC accomplishes the registration. In case the SN already has an account, the CC compares the computed hashed value with that value stored in log-file to grant or deny the connection. Otherwise, the sensor device is authenticated and ready to send data securely.

C. Authentication Scheme Evaluation Results

In this section, Scyther analyzer tool [21] is used to ensure and validate the security of the authentication phase. In this analysis, the authentication scheme is examined against a set of potential attacks. Each row shows the analysis at each process and whether there are attacks or not. Fig. 8 shows the analysis results of the authentication scheme.

Claim				Status		Comments
Authentication	I	Authentication,i2	Secret ni	Ok	Verified	No attacks.
		Authentication,i3	Secret nr	Ok	Verified	No attacks.
		Authentication,i4	Secret k	Ok	Verified	No attacks.
		Authentication,i5	Secret t1	Ok	Verified	No attacks.
		Authentication,i6	Secret t2	Ok	Verified	No attacks.
		Authentication,i7	Secret loc	Ok	Verified	No attacks.
		Authentication,i8	Secret SN	Ok	Verified	No attacks.
		Authentication,i9	Alive	Ok	Verified	No attacks.
		Authentication,i10	Weakagree	Ok	Verified	No attacks.
		Authentication,i11	Niagree	Ok	Verified	No attacks.
		Authentication,i12	Nisynch	Ok	Verified	No attacks.
		Authentication,i13	Commit R,ni,nr	Ok	Verified	No attacks.
	R	Authentication,r2	Secret ni	Ok	Verified	No attacks.
		Authentication,r3	Secret nr	Ok	Verified	No attacks.
		Authentication,r4	Secret k	Ok	Verified	No attacks.
		Authentication,r5	Secret t1	Ok		No attacks within bounds.
		Authentication,r6	Secret t2	Ok		No attacks within bounds.
		Authentication,r7	Secret loc	Ok	Verified	No attacks.
		Authentication,r8	Secret SN	Ok	Verified	No attacks.
		Authentication,r9	Secret h(SN,loc)	Ok	Verified	No attacks.
		Authentication,r10	Alive	Ok	Verified	No attacks.
		Authentication,r11	Weakagree	Ok	Verified	No attacks.
		Authentication,r12	Niagree	Ok	Verified	No attacks.
		Authentication,r13	Nisynch	Ok	Verified	No attacks.
		Authentication,r14	Commit I,nr,ni	Ok	Verified	No attacks.

Fig. 8. Authentication Scheme Analysis.

The initiator and receiver roles represent both sensor and cloud controller sides. The sensor node is represented by 'I' while the cloud controller is represented by 'R'.

During the authentication phase, all sensitive sensor data will be selected as claims. The claim statements ensure secret data. Different types of claims are defined as follows:

Secret: It means the term must be secret.

Alive: It checks the aliveness of all roles. In other words, it checks that the roles are communicating with each other.

Weakagree: To guarantee complete running all the claims in the role during the weak agreement.

Niagree: To guarantee complete running all the claims in the role during the non-injective agreement.

Nisynch: Means that the synchronization is non-injective.

Commit: It represents effective claims.

Under the status, there are two columns show whether the claim is correct and verified or not. The last column shows the comment, which explains the status of the result and whether there are some attacks or not.

Scyther analyzer results, as shown in Fig. 8 prove that all sensitive information exchanged between the sensor node and cloud controller during the authentication phase is secure and protected against any potential attack.

V. Framework Implementation

This section describes the implementation of the suggested integrated framework between cloud computing and WSNs. The wireless network consists of two master nodes (raspberry pi) distributed in different distant geographical locations connected with three sensors. The first master node connected with temperature and humidity sensors, and the second one placed in another location and connected with an air quality sensor. Sensors sense the surrounding environment and transmit sensed data every 5 seconds to the cloud computing for storing in the database. All sensors run simultaneously in different and discrete times. They were connected to the cloud environment for three days continuously, and sometimes for a few hours or even for a few minutes.

The private cloud of Qassim University has been used to implement the framework using Apache Cloud Stack platform. Two application servers and one database server have created in the cloud. Sensor nodes connect to the cloud computing through a load balancer, which works as a gateway to forward the workloads to the most appropriate application server. The application server decrypts the sent data and ensures data integrity before transmitting it to the MySQL database server. The client and cloud servers' codes have been written using Python programming language.

VI. Implementation Results and Analysis

A. Data Integrity Testing

The accuracy of data has been tested every time it is transmitted or processed and stored in the DB server.

The integrity test is applied to 77167 records of data stored in the database. When the cloud server receives sensed data and its hash value, it computes a new hash value and then compares it with that value sent from the sensor node to provide integrity. Checking these two hashed value proves if data being modified or missed after transmitting or during processing.

Received data and hashed values are stored in DB. If values are same, the data status in DB assigned to '0', i.e., data is not modified as shown in Fig. 9; otherwise, data status will be '1'.

As a result, 77029 records with status '0' and 138 records only with status '1'. As noted during the experiment, the status becomes '1' only at the beginning of connection for the first packet sent from the temperature and humidity sensors, and then, the "0" state continues until the end of the connection, which leads to stability during sensor data transmission. On the other hand, sometimes the status becomes '1' when intentionally connection is interrupted during transmission.

SensorValues	dataType	HsensorValue	newHASH	RecordTime	status	RSerialNo
22.0	T	cb30211d8e…	cb30211d8e9…	2019-02-28 15:58:28	0	000000009db266b5
54	A	2fca346db6…	2fca346db656…	2019-02-28 15:58:31	0	00000000d22c7fa0
20.0	H	585348dbd2…	585348dbd28…	2019-02-28 15:58:33	0	000000009db266b5
53	A	2858dcd105…	2858dcd1057…	2019-02-28 15:58:36	0	00000000d22c7fa0
22.0	T	cb30211d8e…	cb30211d8e9…	2019-02-28 15:58:38	0	000000009db266b5
53	A	2858dcd105…	2858dcd1057…	2019-02-28 15:58:41	0	00000000d22c7fa0
20.0	H	585348dbd2…	585348dbd28…	2019-02-28 15:58:43	0	000000009db266b5
53	A	2858dcd105…	2858dcd1057…	2019-02-28 15:58:46	0	00000000d22c7fa0
22.0	T	cb30211d8e…	cb30211d8e9…	2019-02-28 15:58:48	0	000000009db266b5
54	A	2fca346db6…	2fca346db656…	2019-02-28 15:58:51	0	00000000d22c7fa0
20.0	H	585348dbd2…	585348dbd28…	2019-02-28 15:58:53	0	000000009db266b5
52	A	41cfc0d1f2d…	41cfc0d1f2d1…	2019-02-28 15:58:56	0	00000000d22c7fa0
22.0	T	cb30211d8e…	cb30211d8e9…	2019-02-28 15:58:58	0	000000009db266b5
53	A	2858dcd105…	2858dcd1057…	2019-02-28 15:59:01	0	00000000d22c7fa0
20.0	H	585348dbd2…	585348dbd28…	2019-02-28 15:59:03	0	000000009db266b5
52	A	41cfc0d1f2d…	41cfc0d1f2d1…	2019-02-28 15:59:07	0	00000000d22c7fa0
22.0	T	cb30211d8e…	cb30211d8e9…	2019-02-28 15:59:08	0	000000009db266b5
54	A	2fca346db6…	2fca346db656…	2019-02-28 15:59:12	0	00000000d22c7fa0
20.0	H	585348dbd2…	585348dbd28…	2019-02-28 15:59:13	0	000000009db266b5
59	A	3e1e967e9b…	3e1e967e9b7…	2019-02-28 15:59:17	0	00000000d22c7fa0
22.0	T	cb30211d8e…	cb30211d8e9…	2019-02-28 15:59:18	0	000000009db266b5
54	A	2fca346db6…	2fca346db656…	2019-02-28 15:59:22	0	00000000d22c7fa0
20.0	H	585348dbd2…	585348dbd28…	2019-02-28 15:59:23	0	000000009db266b5
52	A	41cfc0d1f2d…	41cfc0d1f2d1…	2019-02-28 15:59:27	0	00000000d22c7fa0
22.0	T	cb30211d8e…	cb30211d8e9…	2019-02-28 15:59:28	0	000000009db266b5
55	A	02d20bbd7e…	02d20bbd7e3…	2019-02-28 15:59:32	0	00000000d22c7fa0
20.0	H	585348dbd2…	585348dbd28…	2019-02-28 15:59:33	0	000000009db266b5
53	A	2858dcd105…	2858dcd1057…	2019-02-28 15:59:37	0	00000000d22c7fa0

Fig. 9. Sample of Sensed Data Stored in DB.

According to the counted zeros and ones, the following evaluation metrics are considered:

-True Positive (TP): 77029 records with equal hashes and status '0';

-False Positive (FP): 0 record with not equal hashes and status '0';

-False Negative (FN): 0 record with equal hashes and status '1';

-True Negative (TN): 138 records with not equal hashes and status '1'.

Furthermore, in order to test the correctness of integration procedures, the received data is first injected with random synthesis data, and the computed hashed value is manually modified to ensure that the system can detect any changes on security procedures. These two measurements were repeated several times with the same results.

B. Availability Testing

In order to assess the availability of the proposed system, load testing is applied in the cloud environment. System availability is checked to make sure that the system is running regularly even if a portion of network or hardware fails. In order to achieve this goal, Cloud-based loading tests are accomplished to handle a large number of requests; in addition, to managing an unlimited number of resources in the cloud.

Apache JMeter tool has been used to generate a high load test. Three VMs were created (one VM as a master and the other two machines as slaves) to produce a large number of virtual users. The master-VM is responsible for distributing the load of a large number of requests among the slaves-VMs. While the slaves-VMs perform the commands of the master by sending requests to the target system, and then they send the results back to the master. The target is an application server, where the load test will be executed.

Response time and throughput are measured to assess the performance of the cloud system. Response time is a total time the system takes to respond to a service request, while the throughput is the number of successful requests sent over a communication channel.

Initially, three virtual-users have been started and executed ten times for each slave, i.e., each slave executes 30 requests. The number of users is increasing exponentially. Both local and remote machines have been tested separately.

Fig. 10 shows the system performance of executing a different number of requests by evaluating local and distributed remote machines.

In local testing, the test is conducted using a single machine. The result shows that the system has high performance if the number of requests is less than 2000 requests. After that, the average response time increased. When the number of requests reaches 10000 requests, the average response time is 1890 ms. With rise up the number of requests to 20000 requests, the average response time extremely increases to 19941ms. Through this massive number of requests, some requests fail while other requests take a long time to be sent. The result of this simulation shows that higher performance is achieved when the number of requests is low. While when the number of requests increases, the response time becomes unacceptable.

In order to obtain higher performance when sending a large number of requests, the system is expanded using cloud capabilities. Remote testing is applied to simulate the cloud environment. Fig. 10 shows how the response time improves when using remote distributed servers. Note that when the number of requests reaches 20000 requests, the average response time is less (14533 ms) than the time spent for the single local server.

Fig. 11 shows the performance of the system using the throughput metric for local and remote testing.

As shown in Fig. 11, the throughput increases when the number of requests increases using both local and remote testing. When the number of requests reaches 10000 requests, the throughput peaks to the highest point. However, the throughput using remote machines is higher than the local machine.

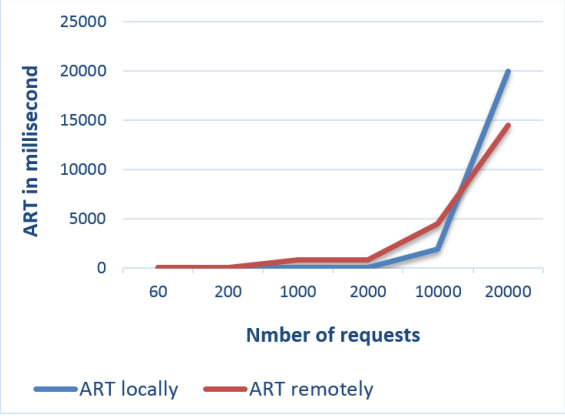

Fig. 10. Average Response Time of Local and Remote Machines.

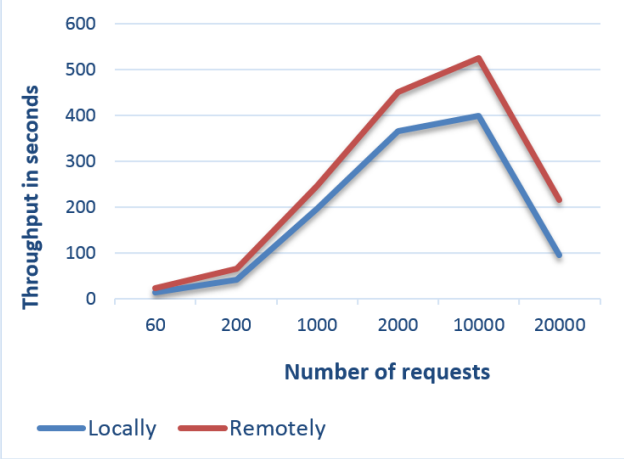

Fig. 11. Throughput Results for Local and Remote Machines.

VII. Conclusion

An integrated cloud-based framework is proposed to improve the security of data transmitted from wireless sensors networks. This paper aims to provide a secure environment for the sensed data. Therefore, the authentication scheme has implemented to ensure and validate sensors identity. In order to provide confidentiality and data integrity, encryption, and hashing mechanisms were used. The experimental results prove that the system performance is improved when sensor networks were integrated with the cloud-computing environment compared to local servers.

The framework proposed in this paper assumes that sensors in WSNs are fixed and their position coordinates are constant. In the future, features of portable sensors must be considered and investigated to enable the framework to protect the dataflow of the mobile sensors as well as static.

Further research is required to develop the framework in public and open source cloud services, and then measure network performance in terms of privacy, scalability, availability, and integrity. Furthermore, there is a need to intensify research to provide diverse methods and techniques for managing cloud resources and linking them to wireless sensor networks. Moreover, the proposed framework must be extended to supplement intrusion detection to improve the security level.

References

[1] A. Agrawal, “A Study on Integration of Wireless Sensor Network and Cloud Computing : Requirements , Challenges and Solutions,” In Proceedings of the Sixth International Conference on Computer and Communication Technology 2015, pp. 152-157.

[2] R. Buyya, C. Yeo, S. Broberg, I. Brandic, and R. Buyyaa, “Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility,” Future Generation Computer Systems, vol. 25, no. 6, 2009, pp. 599-616.

[3] S. H. Shah, F. K. Khan, W. Ali and J. Khan, "A new framework to integrate wireless sensor networks with cloud computing," 2013 IEEE Aerospace Conference, Big Sky, MT, 2013, pp. 1-6. DOI: 10.1109/AERO.2013.6497359.

[4] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud computing: state-of-the-art and research challenges", ", Journal of Internet Services and Applications, vol. 1, no. 1, 2010, pp. 7-18, DOI: 10.1007/s13174-010-0007-6.

[5] S. Sahmim, H. Gharsellaoui, “Privacy and Security in Internet-based Computing: Cloud Computing, Internet of Things, Cloud of Things: a review,” Procedia Computer Science, vol. 112, 2017, pp. 1516-1522, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2017.08.050.

[6] H. Takabi, J. B. D. Joshi, and G. Ahn, "Security and Privacy Challenges in Cloud Computing Environments," IEEE Security & Privacy, vol. 8, no. 6, pp. 24-31, Nov.-Dec. 2010. DOI: 10.1109/MSP.2010.186.

[7] J. Sen, I. Sengupta, “Autonomous Agent Based Distributed Fault-Tolerant Intrusion Detection System,” In: Chakraborty G. (eds) Distributed Computing and Internet Technology. ICDCIT 2005, vol. 3816. pp. 125-131, Springer, Berlin, Heidelberg.

[8] J. Yick, B. Mukherjee, and D. Ghosal, “Wireless sensor network survey,” Computer Networks, vol. 52, no. 12, 2008, pp. 2292-2330, DOI: 10.1016/j.comnet.2008.04.002.

[9] F. Lewis, “Wireless sensor networks”, In: Smart Environments: Technologies, Protocols, and Applications, Ch. 2, John Wiley, New York, 2004, pp. 11-46.

[10] K. N. SunilKumar and Shivashankar, “A Review on Security and Privacy Issues in Wireless Sensor Networks,” 2017 2nd IEEE International Conference on Recent Trends in Electronic, Information & Communication Technology (RTEICT), 2017.

[11] H. Kobo, A. Abo-Mahfouz, and G. HANCKE, “A Survey on Software-Defined Wireless Sensor Networks: Challenges and Design Requirements,” IEEE Access, vol. 5, 2017, pp. 1872-1899.

[12] K. Islam, W. Shen, and X. Wang, "Security and privacy considerations for Wireless Sensor Networks in smart home environments," Proceedings of the 2012 IEEE 16th International Conference on Computer Supported Cooperative Work in Design (CSCWD), Wuhan, China, pp. 626-633. 2012, DOI: 10.1109/CSCWD.2012.6221884.

[13] C. Zhu, H. Wang, X. Liu, L. Shu, L. T. Yang, and V. C. M. Leung, “A Novel Sensory Data Processing Framework to Integrate Sensor Networks with Mobile Cloud,”. IEEE Systems Journal, vol. 10, no. 3, Sep. 2016, pp. 1125–1136.

[14] S. Saha, R. Das, S. Datta, and S. Neogy, (2016). “A cloud security framework for a data centric WSN application,” In Proceedings of the 17th International Conference on Distributed Computing and Networking - ICDCN ’16, 2016, pp. 1–6.

[15] F. Banaie, and S. Seno, “A cloud-based architecture for secure and reliable service provisioning in wireless sensor network,” 2014 4th International Conference on Computer and Knowledge Engineering (ICCKE), October 2014, pp. 96-101, Mashhad, Iran.

[16] M. Aseeri, M. R. Ahmed, S. N. Sakib, and M. S. Kaisert, “A unique framework to integrate secured BAN and could computing to monitor patient,” In 2016 5th International Conference on Informatics, Electronics and Vision, ICIEV 2016, pp. 813–818, Institute of Electrical and Electronics Engineers Inc., Dhaka, Bangladesh, 2016, DOI: 10.1109/ICIEV.2016.7760114.

[17] M. Kassim, and A. Harun, “Wireless sensor networks and cloud computing integrated architecture for agricultural environment applications,” 2017 Eleventh International Conference on Sensing Technology (ICST), 2017.

[18] M. Jassas, J. Mathew, A. Azim, and Q. H. Mahmoud, "A framework for extending resources of embedded systems using the Cloud," 2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE), Windsor, ON, Canada, 2017, DOI: 10.1109/CCECE.2017.7946662.

[19] K. Zkik, G. Orhanou, and S. ELHajji, "A new secure framework in WSNs using ECC: Medical application," 2017 International Conference on Engineering and Technology (ICET), 2017, pp. 1-7, DOI: 10.1109/ICEngTechnol.2017.8308144.

[20] K. H. A. Al-Shqeerat, M. A. A. Hammoudeh, M. I. Abbasi, “Design and Analysis of an Effective Secure Cloud System at Qassim University,” International Journal of Computer Science and Information Security, vol. 14, no. 8, 2016.

[21] C. Cremers, “The Scyther Tool: Verification, Falsification, and Analysis of Security Protocols,” In: Gupta A., Malik S. (eds) Computer Aided Verification. CAV 2008. Lecture Notes in Computer Science, vol. 5123, pp 414-418. Springer, Berlin, Heidelberg, 2008.