LIGHT-WEIGHT SECURITY AND DATA PROVENANCE FOR MULTI-HOP INTERNET OF THINGS

Seminar Report

Submitted in partial fulfillment of the requirements for the award of degree of

BACHELOR OF TECHNOLOGY

In

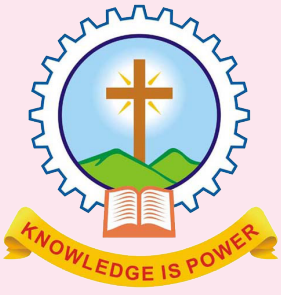
COMPUTER SCIENCE AND ENGINEERING

of

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

Submitted By

NITHIN BENNY



Department of Computer Science & Engineering
Mar Athanasius College of Engineering Kothamangalam

LIGHT-WEIGHT SECURITY AND DATA PROVENANCE FOR MULTI-HOP INTERNET OF THINGS

Seminar Report

Submitted in partial fulfillment of the requirements for the award of degree of

BACHELOR OF TECHNOLOGY

In

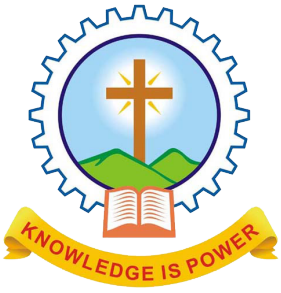
COMPUTER SCIENCE AND ENGINEERING

of

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

Submitted By

NITHIN BENNY



Department of Computer Science & Engineering
Mar Athanasius College of Engineering Kothamangalam

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
MAR ATHANASIUS COLLEGE OF ENGINEERING
KOTHAMANGALAM

CERTIFICATE

This is to certify that the report entitled **Light-Weight Security and Data Provenance for Multi-Hop Internet of Things** *submitted by* **Mr. NITHIN BENNY, Reg. No.MAC15CS044** *towards partial fulfillment of the requirement for the award of Degree of Bachelor of Technology in Computer science and Engineering from APJ Abdul Kalam Technological University for December 2018 is a bonafide record of the seminar carried out by him under our supervision and guidance.*

.....	.....	.....
Prof. Joby George	**Prof. Neethu Subash**	**Dr. Surekha Mariam Varghese**
Faculty Guide	***Faculty Guide***	***Head Of Department***

Date:

Dept. Seal

ACKNOWLEDGEMENT

First and foremost, I sincerely thank the 'God Almighty' for his grace for the successful and timely completion of the seminar.

I express my sincere gratitude and thanks to Dr. Solly George, Principal and Dr. Surekha Mariam Varghese, Head Of the Department for providing the necessary facilities and their encouragement and support.

I owe special thanks to the staff-in-charge Prof. Joby george, Prof. Neethu Subash and Prof. Joby Anu Mathew for their corrections, suggestions and sincere efforts to co-ordinate the seminar under a tight schedule.

I express my sincere thanks to staff members in the Department of Computer Science and Engineering who have taken sincere efforts in helping me to conduct this seminar.

Finally, I would like to acknowledge the heartfelt efforts, comments, criticisms, co-operation and tremendous support given to me by my dear friends during the preparation of the seminar and also during the presentation without whose support this work would have been all the more difficult to accomplish.

ABSTRACT

The security protocols for the Internet of Things need to be light weighted due to the limited resources and scalability. The cryptographic solutions are not feasible to apply on small and low-energy devices of Iot because of their energy and space limitations. A light-weight protocol to secure the data and achieving data provenance is presented for the multi-hop Iot network. The Received Signal Strength Indicator (RSSI) of communicating Iot nodes are used to generate the link fingerprints. The link fingerprints are matched at the server to compute the correlation coefficient. Higher the value of correlation coefficient, higher the percentage of the secured data transfers. Lower value gives the detection of adversarial node in between a specific link. Data provenance has been achieved by comparison of packet header with all the available link fingerprints at the server. The time complexity is computed at the node and server level. The energy dissipation is calculated for the IoT nodes and overall network. The RSSI values are taken in real time from MICAz motes.

Contents

List of Figures

List of Abbreviations

IoT	Internet of Things
RSSI	Received Signal Strength Indicator
EVM	Error Vector Magnitude
ToA	Time of Arrival

Introduction

Internet of Things (IoT) comprises a complex network of smart devices, which frequently exchange data through the Internet [1]. IoT has become the necessity for the future communication. It is estimated that 50 billion smart devices will be connected through IoT by 2020 [2]. The information of a patient to a medical staff, automobile's performance and statistics, home automation, transportation domain, smart grids and smart meters will be based on IoT. The data acquired from sensors or IoT nodes is propagated to Internet cloud where it is received by the concerned body. The acquired data needs to be accurate and should have the information about its origin.

As the number of nodes are large in number, small in size and mostly accessible, the measures should be taken to make sure that the data is secured and effciently received at the receiving end. Data security and provenance act as backbone in order to implement IoT network because the IoT nodes are not physically protected [3]. The data can easily be forged or tampered if proper security primitives are not taken. Security primitives include detection of certain attacks, masking channel state, intrusion detection, location distinction and data provenance. Provenance is to find the origin of the data. A single change in data might cause big problems e.g., in terms of medical health report generated by an IoT node sent to a doctor, meter reading sent to the company for billing according to the consumption and change in transportation system information [1]. Therefore, the traditional cryptographic techniques are not the viable solution in IoT because of the energy limitations of the IoT nodes [4]. Less space acquiring and energy effcient security primitives with less computational complexities are key building blocks for enabling end-to-end content protection, user authentication, and consumer confidentiality in the IoT world [2].

To ensure the trust of users, the IoT-based network should be secured enough. The security mechanism involved should be light-weighted because of the low energy requirements for IoT nodes [5]. The mutual authentication between IoT nodes with the server should also be secured and authentic [3]. Accurate and secure data provenance in the IoT are used for improving the level of trust. The data provenance is useful for determining and describing

the derivation history of data starting from the original resource. The records can be used to protect intellectual property and its relevance from the perspective of regulatory mechanisms. However, the data provenance integrity is a big question. The data provenance can be forged or tampered by an unauthorized party if the provenance is not properly protected by implementing inefficient security protocols. In order to establish the trust of IoT, a solution to security should be designed which is light-weight and highly secured [6]. Most of the security algorithms and cryptography techniques used today contain high computational complexities with high energy consumption.

The solution proposed in this paper incorporates light-weight security algorithms for secured IoT-based information exchange without using extra hardware. Adversarial node is detected effectively by correlating the link fingerprints generated by the adjacent IoT nodes. The correlation coefficient is computed at the server. Data provenance is also achieved using the same link fingerprints generated to find the intrusion detection in the IoT network. Hence, fingerprints are used to authenticate the integrity of data and in the detection of intrusion. The proposed solution has less time complexity compared to other state-of-the-art available solutions. The energy calculations are presented as well showing very desirable results when compared to the previously work done in [7].

Literature review

Due to scalability of IoT devices, it is difficult to protect them. That is why they are very prone to attacks [3]. The taxonomy of attacks in IoT are spoofing, altering, replaying routing information, Sybil attack [6], Denial of Service (DoS) attacks [7], attacks based on node property, attacks based on access level, attacks based on adversary location and attacks based on information damage level [1] etc. In order to tackle these attacks, a required solution needs to be light-weighted and secured enough to gain the trust of IoT users [10].Acryptographic solution to secure the IoT network is provided using Advanced Encryption Standard (AES)-128 Algorithm and Inverse AES-128 Algorithm. These solutions deal with intense cryptography and computational complexities. That is why AES-128 algorithm is not suitable for IoT considering a large number of IoT nodes.

Working on the mutual authentication between RFID tags in IoT, researchers introduced a light-weight protocol by encryption method based on XOR manipulation, instead of complex encryption such as using the hash function, for anti-counterfeiting and privacy protection. In unsecured RFID the attacker can clone the Electronic Product Key (EPC) of the target tag and program it to another tag. Physical Unclonable Functions (PUFs) are used at the node end to protect it from the attacker to get access to the information stored in the node memory. PUFs may be used to provide security in IoT systems without the need to store secrets in the nodes. For communication purposes, a light-weight messaging protocol called MQ Telemetry Transport (MQTT) can be used. A centralized "broker" is used to communicate with terminals. MQTT broker controls the type of information shared among terminals, which helps to protect the privacy. Elliptic Curve Cryptography (ECC) is also preferred because it provides an equal amount of security with less computation power and bandwidth than its Rivest, Shamir, and Adelman (RSA) counterpart. In some papers, the concept of mutual trust between security systems on IoT objects through the establishment of a framework for access control at the node level is discussed. According to the researchers, trust is established from the creation phase to the operation phase in IoT. This trust arises through two mechanisms; the creation of key and the token key created by the manufacturer. Based on the new Lightweight Label-Based

Access Control Scheme (LACS), the authentication of authorized fog nodes is achieved to ensure protection. Specifically, LACS authenticates fog node by checking the integrity of the value of the shared file embedded label, where only the authorized fog node has access to the caching service. Both the physical and social layer information are combined for realizing rapid content dissemination in device-to-device vehicle-to-vehicle (D2D-V2V)-based IoT networks.

In [7] paper, securing the data provenance is achieved by using the RSSI values received by a static base station and a mobile body-worn device. Performed experiments show that highly correlated fingerprints are acquired. After every 10 to 15 minutes, a link fingerprint of 128 bits is generated by using RSSI at base station and body worn device. The storing and accessing of data provenance are also important to be a secured process. The proposed trust model is described for cloud computing in [6]. High trust can be achieved using the same model in IoT environment. Improved energy efficiency is achieved by using Gale-Shapley algorithm which matches D2D pair with cellular user equipments (UEs). Correlation among UEs are analyzed using a game-theoretic approach.

Existing system

3.1 Device limitations

Why is it difficult to secure and apply security features to IoT as those used in traditional Internet? Trappe et al. [5] presented the issue of IoT constraints, and their effects on using current cryptographic tools as the ones utilized in traditional Internet. The two main limitations are the battery capacity and computing power.

Battery Life Extension: Because some IoT devices are deployed in environments where charging is not available, they only have a limited energy to execute the designed functionality and heavy security instructions can drain the devices' resources. Three possible approaches can be used to mitigate this issue. The first is to use the minimum security requirements on the device, which is not recommended especially when dealing with sensitive data. The second approach is to increase the battery capacity. However, most IoT devices are designed to be lightweight and in small size. There is no extra room for a larger battery. The final approach is to harvest energy from natural resources (e.g., light, heat, vibration, wind), but this type of approach would require an upgrade to the hardware and significantly increase the monetary cost.

Lightweight Computation: The paper [5] mentioned that conventional cryptography cannot work on IoT systems, since the devices have limited memory space which can't handle the computing and storage requirements of advanced cryptography algorithms. To support security mechanisms for the constrained devices, the authors suggested reusing existing functions. An example is to use physical layer authentication by applying signal processing at the receiver side to verify whether a transmission came from the expected transmitter in the expected location. Alternatively, a specific analog characteristics of a transmitter can be used to effectively encode analog information. These analog nuances can't be predicted or controlled in manufacturing, and can serve as a unique key. This way of authentication has little or no energy overhead because it takes advantage of radio signals.

Shafagh et al. [4] proposed an Encrypted Query Processing algorithm for IoT. The approach allows to securely store encrypted IoT information on the cloud, and supports efficient database query processing over encrypted data. Specifically, they utilize alternative lightweight cryptographic algorithms that replace additive homomorphic encryption and orderpreserving encryption with Elliptic Curve ElGamal and mutable order preserving encoding algorithms, where they made some changes to suit the computation limitations of IoT devices. The system scheme replaces the web application communication with an End-to-End system that stores encrypted data from personal devices on Cloud database, and data encryption/decryption is performed at the client-side. The keying material will only reside in the personal device, and the need of a trusted proxy which has access to all the secret keys is eliminated. The system architecture includes three main parties: IoT devices, users, and the Cloud. The application data can be stored in the Cloud by directly uploading it by the smart device or via a gateway like a wearable device. The paper addressed only some encryption schemes that support the most used queries in IoT data processing. However, the design can be extended to cover more schemes. The experiment results showed an improvement in the time performance compared to existing schemes.

Kotamsetty et al proposed an approach to reduce latency for IoT when performing query processing over encrypted data by applying latency hiding technique, which consists of breaking down the query results of large size into small sized data sets. This allows computational work to be performed on a set of data while fetching the remaining encrypted information. To decide the appropriate data size to be requested in each iteration in order to minimize the latency, the study proposed an algorithm that starts with an initial data size and adoptively adjust the size to minimize the gap between computation and communication latencies in each iteration. The algorithm has two variants: the first starts with a size that is a fraction of the large query size. In the second variant, the starting size is fixed. The experiment results demonstrated that the proposed approach outperforms existing solutions in terms of latency for queries with larger data size. Salami et al proposed a lightweight encryption scheme for smart homes based on stateful Identity-based Encryption, in which the public keys are merely identity strings without the need for a digital certificate. This method is known as Phong, Matsuka and Ogata

(PMO)'s stateful IBE scheme. It is the combination of IBE and stateful Diffie-Hellman (DH) encryption scheme. To add more efficiency to the proposed scheme and reduce the communication cost, the research study divides the encryption process into key encryption and data encryption, with the focus on the second one, because the size of ciphertexts produced by key encryption is larger than the one resulted from the data encryption. This division led to two-sub algorithms: KEYEncrypt and DATAEncrypt. The first is for encrypting a session key, and the second is for data encryption. The resulted ciphertext from the sub algorithms is transmitted separately in a way that data ciphertexts are transmitted many times without attaching the key ciphertext. The evaluation results showed that the proposed scheme is secure against plaintext attacks. Also, the performance analysis showed that it outperforms the regular IBE scheme in terms of speeding up the encryption operations, and reducing approximately one-third of communication overhead.

3.2 Attack classification

Andrea et al come up with a new classification of IoT devices attacks presented in four distinct types: physical, network, software, and encryption attacks. Each one covers a layer of the IoT structure (physical, network, and application), in addition to the IoT protocols for data encryption. The physical attack is performed when the attacker is in a close distance of the device. The network attacks consist of manipulating the IoT network system to cause damage. The software attacks happen when the IoT applications present some security vulnerabilities that allow the attacker to seize the opportunity and harm the system. Encryption attacks consist of breaking the system encryption. This kind of attacks can be done by side channel, cryptanalysis, and man-in-the-middle attacks. They also presented a multi-layered security approaches to address the IoT structure layers and encryption system vulnerabilities and security issues. Based on the study, to countermeasure the security problems at the physical layer, the device has to use secure booting by applying a cryptographic hash algorithms and digital signature to verify its authentication and the integrity of the software. Also, a new device must authenticate itself to the network before any transmission or reception of data. In addition to that, a device should carry an error detection system, and all of its information has to be encrypted

to maintain data integrity and confidentiality. At the network layer, authentication mechanisms and point-to-point encryption can be used to ensure data privacy and rooting security. The application layer can also provide security by means of authentication, encryption, and integrity verification, which allows only the authorized users to access data through control lists and firewalls, in addition to the use of anti-virus software. Ronen et al introduced a new taxonomy classification for IoT attacks based on how the attacker features deviates from the legitimate IoT devices. The categories are presented in: ignoring, reducing, misusing, and extending the system functionality. The study focused on the functionality extension attacks on smart lights. The paper presented two attacks: the first one consisted of creating a covert channel to capture confidential information from an organization building that implemented smart lights which are connected to the internal sensitive network. The work is done by using an optical receiver that could read the data from a distance of over 100 meters by measuring the exact duration and frequency of the small changes in the lights intensity. The second attack showed that an attacker can use those lights to create strobes in the sensitive light frequencies, which can lead to a risk of epileptic seizures. The experiments showed that it is necessary to focus on security issues during the different phases of designing, implementing and integrating of the IoT devices.

3.3 Authentication and access control

A. IoT Authentication Scheme : Salman et al proposed a new IoT heterogeneous identity-based authentication scheme by applying the concept of Software Defined Networking (SDN) on IoT devices. SDN can be deployed using fog-distributed nodes. Each set of devices is communicating with a gateway that can support authentication for the things. These gateways are also connected to a central controller which has access to the central data. The authentication process has to go through the gateway and then the controller in order to give access to the things. The message flow between the three levels: things, gateway, and the controller, happens in three phases.

The first phase consists of obtaining an authentication certificate for the gateway from a controller. Phase two consists of things registration to the gateway. The final phase is the authentication request which is sent from the IoT device to the gateway.

The experimental evaluation shows that the proposed scheme is immune to masquerade attack, man-in-the-middle attack, and replay attack. Porambage et al proposed and designed a pervasive authentication protocol and a key establishment scheme for the resource constrained wireless sensor networks (WSNs) in distributed IoT application, called PAuthKey. The proposed PAuthKey protocol comprises two phases: registration phase for obtaining cryptographic credentials to the edge devices and end users; authentication phase for authentication and key establishment in mutual communication. With PAuthKey protocol, end-users can authenticate themselves to the sensor nodes directly and acquire sensed data and services. The protocol supports the distributed IoT applications, since the certificates are lightweight and can be handled by the high resource constrained devices, irrespective of their originality. Ho et al studied the security vulnerabilities of smart locks by observing five types of locks: August, Danalock, Kevo, Okidokeys, and Lockitron. The paper focused on the consequence of the door's automatic unlocking system. Some locks have the capability to unlock the door if the owner is located in a certain distance from the door. This feature allows to open the door even if the owner doesn't have the intent for the action to occur, especially when the person is inside the home. This can create an insecure feeling for the resident and allows the attacker to seize the opportunity and enter the home when the owner is around without his/her permission. To countermeasure this vulnerability, the study proposed a touch-based intent communication solution that prevents locks to unlock the door without the owner intent to do it. In this solution, the authorized user has to wear a special wearable device that communicate with the lock via an ear bone conduction microphone. A hand-held vibrator is used to transmit the intent signal. The wearable device will detect the vibration and send an unlock command. The results showed that the system unlocks only when it detects the person's action, and it didn't react to the vibration caused by any of daily activities such as computer tone and phone vibration; however, the solution presented some limitations like the addition of hardware to the smart lock, and the wearable device to be able to transmit the vibrations. Also, the vibration sensor may not detect the intent action if the wearable device is loosen, or the user is touching the door with the hand which is not wearing the device. Sharaf et al proposed a new approach for authentication process using the device's unique fingerprint. According to the study, each

device has a unique fingerprint which consists of multiple features such as location, physical state of object, or transmitter state. A group of IoT objects may have different types of fingerprinting features. For that reason, conventional device fingerprinting techniques can't be used for the IoT object's authentication. The paper proposed the use of transfer learning, to authenticate devices that have different feature spaces. To apply the new idea, the research study followed two-fold approach. First, it verifies if the message is sent by a single object. Then, it validates the legitimacy of the sending device. To realize the first phase, the paper adopted the Infinite Gaussian Mixture Mode (IGMM) as a generative model assuming that the fingerprints for each object follow a multivariate Gaussian distribution. The second phase was done by comparing the clustering results from the IGMM with the expected cluster shape for the device. This was done by applying Bhattacharyya distance. However, the environment can cause changes in some devices' fingerprint features. To solve this issue, the study applied transfer learning techniques to differentiate between normal changes due to the environment effects, from the malicious changes produced by attackers. This is done under two assumptions. The first is that the changes can affect more than an object at the same time, and the second is that an attacker cannot target all objects affected by the environment. The test results of the proposed authentication approach showed an increase in the authentication performance compared to conventional authentication techniques. Zhang et al proposed an algorithm to defend against DDoS attacks by considering a network composed of four groups of nodes: working node, monitoring node, legitimate user node, and the attacker node. The algorithm proposed consists of addressing each node's DDoS security issues in the network. The working nodes are considered as the devices that collect information and execute simple tasks. They have memory computation, storage, and energy limitations. To countermeasure the DDoS attack, the working node has to differentiate between malicious requests and legitimate ones. A sender that sends the same content messages will be flagged and saved in a list of served requests to check for further attacker requests. The list has to be of small size due to the devices' space constraint. A legitimate user node has to send request with lower frequency and reasonable content. A monitoring node is included in the scheme for future work implementation. The node will be responsible for storing the old records of attackers in order to prevent the working

nodes from serving the malicious attacks. In the proposed algorithm, an attacker's request has only one chance to be served. After the second attempt, the attacker is put in the attacking list, and its packets will be dropped. The study simulation results showed that the algorithm is effective for detecting and preventing DDoS. Bouij et al proposed an authorization access control model called SmartOrBAC that extends the OrBAC (Organization-based Access Control) model to fit the IoT network requirements by including collaboration-related and context aware concepts, and dividing the IoT network structure into four abstraction layers: constrained, less constrained, organization layer, and collaboration layer across domain access control, with a central authorization engine for each separate group of components within a specific layer. The constrained layer, as its name says, contains devices with constrained capabilities. A less constrained device is associated to a group of the first layer components to take in charge the intensive computation tasks within the same security domain. This central element of the less constrained layer is referred as Client Authorization Engine (CAE), on the client side, and Resource Authorization Engine (RAE), on the source side. The Organization layer specifies the security access policies for each group of the client and the resource organization. It also structures them into different security domains. The fourth layer comes to enhance the OrBAC access model with the addition of collaboration related concepts. This added layer is responsible for establishing agreements and rules cross the domain access control. The evaluation of the presented model showed that it is less complex that the capabilities based models. It also ameliorates the security policy management cost, and reduces the risks of errors.

B. IoT Authentication Architecture : Lessa et al proposed an architecture for secure communication between constrained IoT devices using Datagram Transport Layer Security (DTLS) based on certificates with mutual authentication. The communication is done by introducing a new device called IoT Security Provider (IoTSSP), which is responsible for managing and analyzing the devices' certificates along with authentication and session establishment between the devices. The infrastructure could be composed by one or more IoTSSPs.

Each one is responsible for a set of constrained devices. Optional Handshaking Delegation, and Transfer of Session are the two new main mechanisms that are introduced in the study. The first mechanism consists of delegating the handshaking process to the IoTSSP upon

the reception of a client request for authentication to communicate with a constrained device. The Handshaking Execution Module (HEM) in IPv6 over low power wireless personal area networks border router (6LBR) redirect the message to the IoTSSP, which replies to the Internet device to verify its request. It then communicates the message to the constrained device and check for its availability. This process also prevents DoS attacks. After the authentication process is finished, the second mechanism will take place by using a DTLS extension called Session Transfer Ticket that transfer a secure communication session to the constrained device, which will receive all the parameters of the active session defined in the IoTSSP. The proposed solution earlier is based on a lightweight key agreement protocol, the Identity Based Encryption (IBE), and Pseudonym Based Encryption (PBE) to ensure anonymity, data secrecy, and trust between IoT or WSN nodes in the network. Their architecture consists of a Base Station BS, a sink node SN, and a set of nodes N. the BS contains the PKG server where the nodes' IDs are stored. Their solution requires that all the messages to be transmitted to the SN which then send them to their final destination, and each transmission is acknowledged by an ACK message. The encrypted data will incur a Message Authentication Code function before sending the message. Also, in order to obscure a sent message with an ACK message, the study proposed that both messages will have the same length. Another requirement is that a shared session key should be established between N node and SN, and between SN and BS. Each node N should use a virtual ID and apply PBC technique. Four phases need to be followed to establish the proposed system model. The first step is the network setup, which is also divided in three steps to setup the system's security parameters. These steps consists of configuring the PKG in the BS node, and the SN and N nodes parameters. The second phase highlights the mechanisms that ensures both SN and N nodes are legitimate devices in the network. The third and fourth phase is the establishment of session keys between N node and SN, and between SN and BS. The proposed solution was shown to be resistant to most known attacks in the WSN and IoT.

The results also showed an improvement in security and privacy preservative performance. Yoshigoe et al proposed a way to hide real network traffic with synthetic packet-injection framework, thus making traffic analysis difficult for hackers. The framework consists

of a Synthetic Packet Engine (SPE) that generates and inject additional packets to the network whenever needed. These false packets mimic the behavior of real actions, like opening a door, which is followed by the action of locking the door after a few seconds. The SPE can be incorporated with the use of a VPN, which can encrypt the data and hide the packets sequence number that can distinguish between real traffic and the injected ones. The SPE can also be integrated as a part of both the client and the server process. This combination can be applied to application that does require immediate response from the server, which is not supported when using the SPE with the VPN

Proposed system

When two IoT nodes communicate, then various metrics like RSSI, Time of Arrival (ToA), phasor information and Error Vector Magnitude (EVM) are used to generate link fingerprint. In terms of RSSI, there is a linear relation between the RSSI variations of any connected nodes. This information is helpful in generating the link fingerprints which are highly correlated for two connected nodes by computing the Pearson correlation coefficient. We can use this information to develop link fingerprints. The RSSI values are recorded in real time by using MICAz motes.

The duration of recording RSSI values at each IoT node can be increased or decreased depending on the availability of power to the nodes. As the IoT nodes are power limited, realistic approach is to take the recording time large but acceptable in a manner that the results are not affected.

4.1 System model

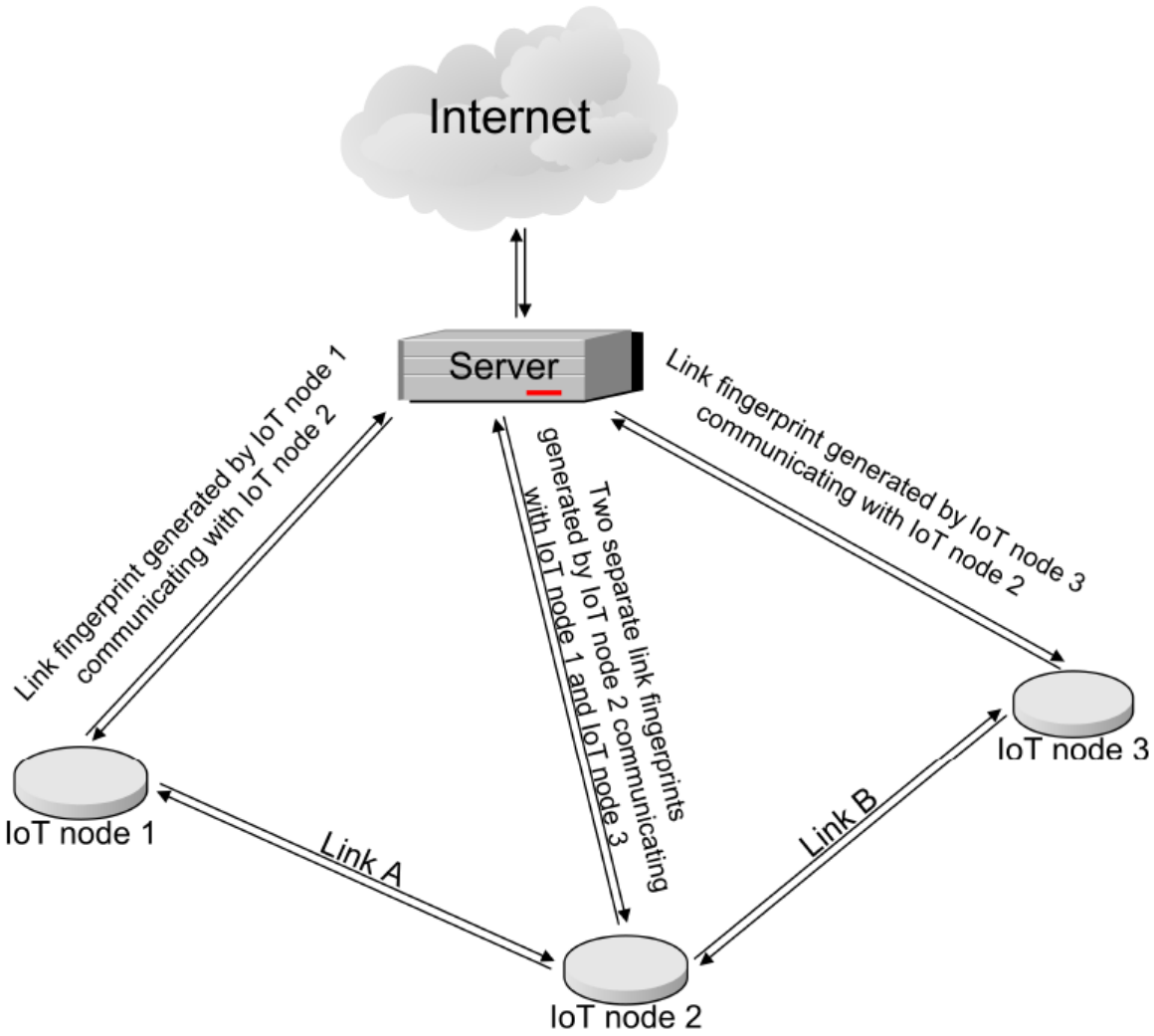


Fig. 4.1: System model

The scheme presented in this paper ensures security of IoT network for all the scenarios mentioned above consuming less energy. It uses real-time experimental values. MICAz motes are used as IoT nodes.

4.2 Adversarial node detection

4.2.1 Received signal strength indicator

In our experiment, each IoT node records its respective RSSI values after every 20 seconds. The RSSI values received are in dBm ranging from -48 dBm to 20 dBm. The signal strength is calculated using Friis transmission equation which states that

$$P_r = \frac{P_t G_t G_r}{L_p} \qquad \text{(Equ: 4.1)}$$

where, Pr is the received power, Pt represents the transmitted power, Gr and Gt are the receiving and transmitting antennas gains, respectively and Lp is the path loss. More the path loss, less will be the received power and hence low value of RSSI. Path loss is expressed as:

$$L_p = (\frac{4\pi d}{\lambda})^2 \qquad \text{(Equ: 4.2)}$$

where d is the distance between two communicating IoT nodes. Lambda is the wavelength which is approximately 416 micro meter because the operating frequency of MICAz motes is 2.4 GHz. A gain of 50 is given to make all the values positive.

4.2.2 Quantization

The resulting RSSI values are quantized using word-length of 8 bit providing 256 levels (L). The amplitude values are mapped onto a finite set of known values. This is achieved by dividing the distance between minimum and maximum RSSI values into L zones, each of height 1, which is given as,

$$\Delta = \frac{P_{r(max)} - P_{r(min)}}{L} \quad \text{(Equ: 4.3)}$$

Pr(max) and Pr(min) are the maximum and minimum received powers, respectively. The midpoint of each zone is assigned a value from 0 to L -1. Each sample falling in a zone is approximated to the value of the midpoint. Each zone is then assigned an 8 bit of word-length. This 8-bit word-length is representing the link fingerprint (LF).

4.2.3 Vernam cipher

The link fingerprint (each 8-bit binary stream representing RSSI value) is then encoded with an 8-bit secret key i.e., K1 for IoT node 1, K2 for IoT node 2 and K3 for IoT node 3.

$$LF_{encoded(1\rightarrow n)} = LF_{1\rightarrow n} \oplus K_i \quad \text{(Equ: 4.4)}$$

$\oplus$ represents logical exclusive-OR operation, whereas LFencoded is the encoded link fingerprint. Each IoT node sends LFencoded to the server and keeps a copy of the same with itself. The link fingerprint and the secret key will not be shared with any other IoT node. The server is assumed as highly secured and the data is stored after the authentication is successful. Though in one case, it is considered that adversarial node can send its data to the server by replacing IoT node. K1, K2 and K3 are present at the server, which are assumed to be fully protected. The server decodes all the received encoded link fingerprints of each IoT node using key associated to the concerned IoT node as,

$$LF_{1\rightarrow n} = K_i \oplus LF_{encoded(1\rightarrow n)} \quad \text{(Equ: 4.5)}$$

4.2.4 Pearson correlation coefficient

The binary coded link fingerprints are converted to the respective decimal values in dBm and correlation process is performed by computing the Pearson correlation coefficient (p). If the value is between 0.8 and 1 then it is considered as highly correlated in a multi-hop network. Mathematically,

$$\rho_{X,Y} = \frac{cov(X,Y)}{\sigma_X \sigma_Y} \quad \text{(Equ: 4.6)}$$

where, cov is the covariance and sigma represents the standard deviation. A simplified equation can be written as;

$$r = \frac{\sum_{i=1}^{n}(X_i - \overline{X})\sum_{i=1}^{n}(Y_i - \overline{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \overline{X})^2}\sqrt{\sum_{i=1}^{n}(Y_i - \overline{Y})^2}} \quad \text{(Equ: 4.7)}$$

where Xi and Yi are the RSSI values of the ith packet received at communicating IoT nodes and XN and YN are the respective mean RSSI values of a sequence of n packets. The correlation coefficient r returns a value in [-1:1] where 1 indicates perfect correlation, 0 indicates no correlation, and -1 indicates anti-correlation.

The server correlates the LFs of adjacent IoT nodes. They are highly correlated if there is no involvement of any adversarial node in the IoT network. If any adversarial node comes between IoT node 1 and IoT node 2 then the link fingerprint received by IoT node 1 is different than link fingerprint received by IoT node 2. A highly uncorrelated Pearson correlation coefficient is computed. The decoding is done at the server using the keys already present at the server. Algorithm 1 and 2 represent the detection of adversarial node's presence in IoT network.

4.2.5 Algorithm

Algorithm 1 : Link Fingerprint Generation and Encoding at IoT Node. i = 1 → n and j = 1 → Number of IoT Nodes

Initialize the IoT node

Read the RSSI values from adjacent IoT node

$RSSI_{new}[i]$ = RSSI[i]+ gain

Quantize $RSSI_{new}[i]$

LinkFingerprint[i] = Assign binary code-word to Quantized RSSInew[i]

$RSSI_{en}[i]$ = XOR(LinkFingerprint[i],$Key_{node(j)}$)

$RSSI_{en}[i]$ bundled up with session identifiers

Keep a copy at the IoT Node

Send a copy to the server

Algorithm 2 : Adversarial Node's Detection at the Server. ρ Is the Pearson Correlation Coefficient Having Values Between -1 and 1

LinkFingerprint[i] → XOR$RSSI_{en}[i], Key_{node(a))}$

$RSSI_{new[i]}$ → = bin-dec conversion(LinkFingerprint[i])

LinkFingerprint[j] → XOR($RSSI_{en[j]}, Key_{node(b)}$)

$RSSI_{new[j]}$ → bin-dec conversion(LinkFingerprint[j])

$rho(RSSI_{new[i]}, RSSI_{new[j]})$

if rho between 0.9 and 1 then

return No adversarial node is present

else if ρ=-1 to 0.9 then

return Adversarial node is present

else

return The RSSI values are not correctly measured

end if

4.3 Data provenance

For data provenance, header information is used to reach the origin from which the data is originated. As discussed earlier, each IoT node sends the copy of the link fingerprints to the server, so all the header information will already be present at the server. If the information is received at IoT node 3 from IoT node 1 via IoT node 2, the link fingerprints of header are compared at the server in sequence with copies of link fingerprints previously sent by the IoT nodes. From whichever IoT node the last header information matches, the data is originated from that IoT node. Size of header depends on the selection of packet size. In our case, the header size is 16 bytes. Algorithm 3 describes the data provenance in which the IoT nodes are connected to each other. Each IoT node attaches the encoded link fingerprint as header to the packet it receives and forwards it to the next IoT node. At the end, the concerned node upon receiving the packet adds its own link fingerprint as header and just like any other IoT node, it sends it to the server. The server knows the size of header that each IoT node attaches and the adjacent IoT nodes of each IoT node. In order to check the origin from which the data is originated, server decodes the header with the keys present at the server and correlates the link fingerprint with the already present link fingerprints received from that node. If the link fingerprints match, the same process is repeated for the adjacent IoT node(s).The process continues until:

1) Highly matched link fingerprints are observed and all the header data is exhausted. The origin is the last IoT node from which the header data is matched.

2) Mismatch occurs in link fingerprints showing that the data has been tempered at that node.

While finding the origin of data, if adversarial node is present between any two IoT nodes and the packet flows through adversarial node then the server will still get high correlated result by comparing the link fingerprints. The link fingerprints will match the link fingerprints present at the server received from the IoT node. The reason is that if we consider the situation, the adversarial node is between IoT node 1 and IoT node 2, the IoT node 1 adds the link fingerprint at the header which is of the link between IoT node 1 and adversarial node. Similarly, IoT node

2 adds the link finger print of the link between adversarial node and IoT node 2 to the packet header received from adversarial node and forwards it. The last IoT node on receiving it, adds its link fingerprint. The server checks the header for the origin and gets high correlated value after decoding the header inserted by IoT node 2. The origin can still be measured even if the adversarial node is present in between. Though the link fingerprints of IoT node 1 and IoT node 2 will be highly uncorrelated. The intrusion detection is already performed in section A.

4.3.1 Cases

Case 1 (No Forging of Data): The first case is when the packet is transferred from IoT node 1 to IoT node 3 via IoT node 2, IoT node 1 attaches the encoded link fingerprint to the header and sends it to IoT node 2. IoT node 2 attaches two encoded link fingerprints to the header. One of link A and other of link B. IoT node 3 upon receiving the packet adds its encoded link fingerprint to the packet. When data provenance has to be performed, the packet header is decoded in sequence at the server. Firstly, the last inserted packet is decoded with the key associated with IoT node 3 and link fingerprints are compared with all the available link fingerprints received from IoT node 3. The simulations have shown that the match is 100percent with a part of all the available link fingerprints of IoT node 3. Then the adjacent nodes are checked. As the adjacent node is IoT node 2, so the next sequence of packet is decoded with K2 and 100percent match is detected at some part of all available link fingerprints from IoT node 2. Now the adjacent nodes are checked again. IoT node 2 connected with IoT node 1 and IoT node 3 connected with IoT node 2 are in the adjacency list. Both are checked and 100 percent match is found with a part of all link fingerprints present at the server received from IoT node 2 linked with IoT node 1. Now the same process is done for the next in sequence of header. A 100 percent match in link fingerprints from the header with part of IoT node 1's link fingerprints is achieved. By now, all the header sequences are checked and no header data is left to find a match for. The last header is the first inserted header from ioT node 1 which is received at IoT node 3 in the end.

Case 2 (Packet Is Forged at the Node Level): This case represents a situation when packet is forged at IoT node 1 and is received at IoT node 3 via IoT node 2. Decode the

header in sequence with the key of that IoT node and comparing it with all the available link fingerprints of that IoT node present at the server followed by checking in the table for adjacent IoT node. The results show that when the packet is checked for IoT node 1, the match is not 100 percent rather a very low percentage of match is observed. This shows that the packet data is forged at IoT node 1.

4.3.2 Algorithm

Algorithm 3 Data Provenance

for Header,i, i = n $\rightarrow$ 1 do

// n is the last IoT node the packet is received at

$LinkFingerprintHeader_i$ = XOR($Header_i, Key_i$)

Correlate $LinkFingerprintHeader_i$ with copy of link fingerprints

received from IoTnode[i]

if Correlation greater than 95 percent then

return i $\rightarrow$ i -1

else

Data forged between IoTnode[i] and IoTnode[i-1]

end if

end for

The origin of the packet is IoTnode[i]

4.4 Motes

The RSSI values are taken in real time using MICAz motes shown in Fig 2. The MICAz is a 2.4 GHz, IEEE 802.15.4 compliant mote used for enabling low-power wireless sensor networks. It features a IEEE 802.15.4/ZigBee compliant radio which transceivers use in the 2400 MHz to 2483.5 MHz band, offering both high speed (250 kbps) and hardware security (AES-128). The range of the radio is 75 m to 100 m outdoors and 20 m to 30 m indoors.

The MICAz MPR2400CA platform provides 4 KB of RAM, 128 KB of program ash memory and 512 KB measurement (serial) ash memory. It is very energy efficient with current draw of 8 mA in active mode and less than 15 A in sleep mode. The user interface consists of 3 LEDs - red, green and yellow [6]. The MICAz is capable of running TinyOS , which we use to program the MICAz motes to get the desired RSSI values. The experiment is performed in an indoor environment. The base station is positioned at the lobby to generate log files having RSSI values in dBm of each MICAz mote. Three MICAz motes move randomly in the lobby, halls and labs to generate RSSI values and sends their respective RSSI values to the static base station. The MICAz motes do not cross each other. The orientation of the MICAz motes are kept.

Fig. 4.2: (a) Base station (b) Motes

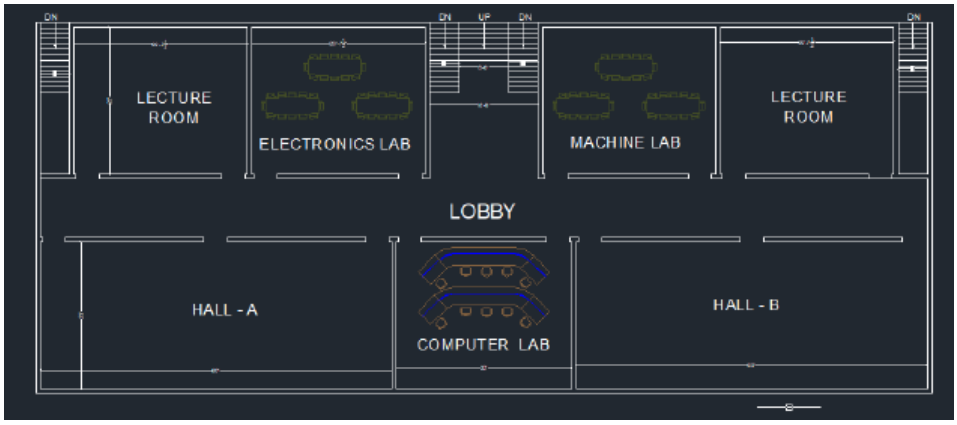


Fig. 4.3: Layout of premises

4.5 Considered cases

The RSSI values acquired from MICAz motes are simulated on MATLAB R2017a. The results have been achieved for various scenarios. Each scenario is presented below. Various cases are implemented and the simulation results are presented for adversarial node detection. The results are achieved by using two methods:

1) Finding Pearson correlation coefficient without using any filter

2) Finding Pearson correlation coefficient by applying Savitzky-Golay filter

The following scenarios are taken in account when performing the experiments and simulations:

1) No adversarial node is present in the IoT network

2) Adversarial node is present in between two communicating IoT nodes

3) The packet is forged or tempered at any IoT node

4) The IoT node is replaced by adversarial node

5) The server is not secured in a way that adversarial node can send its data to the server but cannot access the data present at the server

6) Finding the intrusion in later data using provenance algorithm

Conclusion

The fingerprints generated between any two connected IoT nodes are highly correlated. Introducing an adversarial node gives very low correlation coefficient. It means that the detection of any adversarial node in an IoT network can be done for low power nodes. The data forensics can also be applied by looking at the header of the last received data. The origin of data is computed by extracting the header. The server is considered as highly protected because it contains the keys associated with all the IoT nodes. We get the light-weight solution for the security and data provenance in IoT environment. The energy calculations show that less energy is consumed by applying the link fingerprint generation protocol, sending the packet to the server and to the adjacent IoT node. Time complexity of the system remains the same no matter how lengthy the code becomes.

References

[1] Y. Yang, L. Wu, G. Yin, L. Li, and H. Zhao, "A survey on security and privacy issues in Internet-of-Things, " *IEEE Internet Things J.,*vol. 4, no. 5, pp. 12501258, Oct. 2017. [Online]. Available: http://ieeexplore. ieee.org/document/7902207/

[2] S. Satpathy, S. Mathew, V. Suresh, and R. Krishnamurthy, "Ultra-low energy security circuits for iot applications, "*in Proc. IEEE 34th Int. Conf. Comput. Design (ICCD),* Oct. 2016, pp. 682685.

[3] M. N. Aman, K. C. Chua, and B. Sikdar "Mutual authentication in IoT systems using physical unclonable functions "*IEEE Internet Things J.* vol. 4, no. 5, pp. 13271340, Oct. 2017. [Online]. Available: http://ieeexplore. ieee.org/document/7924368/

[4] T. Idriss, H. Idriss, and M. Bayoumi "A puf-based paradigm for IoT security, " *in Proc. IEEE 3rd World Forum Internet Things (WF-IoT)* Dec. 2016, pp. 700705.

[5] S.-R. Oh and Y.-G. Kim "Security requirements analysis for the IoT " *in Proc. Int. Conf. Platform Technol. Service (PlatCon)* Feb. 2017, pp. 16.

[6] M. I. M. Saad, K. A. Jalil, and M. Manaf "Achieving trust in cloud computing using secure data provenance, " *in Proc. IEEE Conf. Open Syst. (ICOS)*Oct. 2014, pp. 8488.

[7] S. T. Ali, V. Sivaraman, D. Ostry, G. Tsudik, and S. Jha, "Securing rst-hop data provenance for bodyworn devices using wireless link ngerprints " *IEEE Trans. Inf. Forensics Security,* vol. 9, no. 12, pp. 21932204, Dec. 2014.