IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received May 7, 2018, accepted June 12, 2018, date of current version July 12, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2850821

Light-Weight Security and Data Provenance for Multi-Hop Internet of Things

MOHSIN KAMAL, (Member, IEEE), AND MUHAMMAD TARIQ, (Senior Member, IEEE)
Department of Electrical Engineering, National University of Computer and Emerging Sciences, Peshawar 25000, Pakistan
Corresponding author: Mohsin Kamal (mohsin.kamal@nu.edu.pk)

ABSTRACT Due to the limited resources and scalability, the security protocols for the Internet of Things (IoT) need to be light-weighted. The cryptographic solutions are not feasible to apply on small and low-energy devices of IoT because of their energy and space limitations. In this paper, 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 also 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, which is $O(1)$. The energy dissipation is calculated for the IoT nodes and overall network. The results show that the energy consumption of the system presented in this paper is 52–53 mJ for each IoT node and 313.626 mJ for the entire network. The RSSI values are taken in real time from MICAz motes and simulations are performed on MATLAB for adversarial node detection, data provenance, and time-complexity. Experimental results show that up to 97% correlation is achieved when no adversarial node is present in the IoT network.

INDEX TERMS IoT, link-fingerprints, light-weight, provenance, security, multi-hop.

I. 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 efficiently 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 efficient 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].

The rest of this paper is organized as follows. Section II provides an overview on the literature related to IoT security. Methodology of our work is discussed in section III. Experimental and simulation results are presented in section IV. The paper is concluded in section V.

II. 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 [8], Denial of Service (DoS) attacks [9], 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]. A cryptographic solution to secure the IoT network is provided using Advanced Encryption Standard (AES)-128 Algorithm and Inverse AES-128 Algorithm [11]. 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 [12]. 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 [13]. 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 [14]. 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 [15]. 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 [16]. A trusted Internet of Vehicles (IoV) network is proposed in [17]. Both the physical and social layer information are combined for realizing rapid content dissemination in device-to-device vehicle-to-vehicle (D2D-V2V)-based IoV 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. Mutual preferences based on nonlinear fractional programing is also established [18], [19].

III. METHODOLOGY

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 as shown in Fig 1. 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. The following scenarios are taken in account when performing the experiments and simulations:

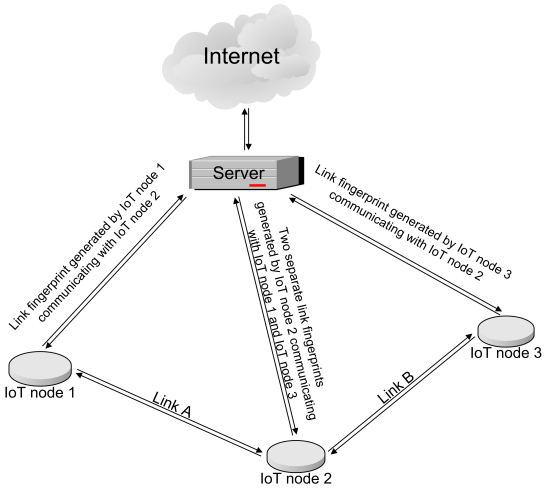

FIGURE 1. System model.

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

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.

A. ADVERSARIAL NODE DETECTION

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

$$P_r = \frac{P_t G_t G_r}{L_p}, \tag{1}$$

where, P_r is the received power, P_t represents the transmitted power, G_r and G_t are the receiving and transmitting antennas gains, respectively and L_p 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:

$$L_p = (\frac{4\pi d}{\lambda})^2, \tag{2}$$

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

$$\Delta = \frac{P_{r(max)} - P_{r(min)}}{L}. \tag{3}$$

$P_{r(max)}$ and $P_{r(min)}$ 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). The link fingerprint (each 8-bit binary stream representing RSSI value) is then encoded with an 8-bit secret key i.e., K_1 for IoT node 1, K_2 for IoT node 2 and K_3 for IoT node 3.

$$LF_{encoded(1\rightarrow n)} = LF_{1\rightarrow n} \oplus K_i. \tag{4}$$

In 4, $\oplus$ represents logical exclusive-OR operation, whereas $LF_{encoded}$ is the encoded link fingerprint. Each IoT node sends $LF_{encoded}$ 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.

K_1, K_2 and K_3 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,

$$LF_{1\rightarrow n} = K_i \oplus LF_{encoded(1\rightarrow n)}. \tag{5}$$

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 (ρ). If the value is between 0.8 and 1 then it is considered as highly correlated in a multi-hop network. Mathematically,

$$\rho_{X,Y} = \frac{cov(X, Y)}{\sigma_X \sigma_Y}, \tag{6}$$

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

$$r = \frac{\sum_{i=1}^{n}(X_i - \bar{X})\sum_{i=1}^{n}(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \bar{X})^2}\sqrt{\sum_{i=1}^{n}(Y_i - \bar{Y})^2}}, \tag{7}$$

where X_i and Y_i are the RSSI values of the ith packet received at communicating IoT nodes and $\bar{X}$ and $\bar{Y}$ 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.

Algorithm 1 Link Fingerprint Generation and Encoding at IoT Node. $i = 1 \rightarrow n$ and $j = 1 \rightarrow$ Number of IoT Nodes

Initialize the IoT node
Read the *RSSI* values from adjacent IoT node
$RSSI_{new}[i] \leftarrow RSSI[i]+$ gain
Quantize $RSSI_{new}[i]$
$LinkFingerprint[i] \leftarrow$ Assign binary code-word to Quantized $RSSI_{new}[i]$
$RSSI_{en}[i] \leftarrow XOR(LinkFingerprint[i], Key_{node(j)})$
$RSSI_{en}[i]$ 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

$LinkFingerprint[i] \leftarrow XOR(RSSI_{en[i]}, Key_{node(a)})$
$RSSI_{new[i]} \leftarrow$bin-dec conversion$(LinkFingerprint[i])$
$LinkFingerprint[j] \leftarrow XOR(RSSI_{en[j]}, Key_{node(b)})$
$RSSI_{new[j]} \leftarrow$bin-dec conversion$(LinkFingerprint[j])$
$\rho\ (RSSI_{new[i]}, RSSI_{new[j]})$
if $0.9 < \rho \leq 1$ **then**
  return No adversarial node is present
else if $\rho = -1$ *to* 0.9 **then**
  return Adversarial node is present
else
  return The *RSSI* values are not correctly measured
end if

B. 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 in a way described in Fig 1. 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;

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>95% **then**
    return $i \leftarrow i - 1$
  else
    Data forged between $IoTnode[i]$ and $IoTnode[i-1]$
  end if
end for
The origin of the packet is $IoTnode[i]$

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 mentioned situation in Fig 1, 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.

As IoT nodes will be large in number, the physical protection will not be possible for most of the nodes. The data can be easily forged or tempered. If the data is tempered at IoT node 2 and sent to IoT node 3 afterwards, the data provenance cannot be achieved rather the adversarial node's involvement can be detected. The process can tell exactly between which link the data has been forged. This is a very useful information in data forensics. The highly uncorrelated result is achieved when comparing the link fingerprints in the header and the ones present at the server. Algorithm 3 represents the achievement of data provenance.

IV. RESULTS

A. EXPERIMENTAL RESULTS

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 flash memory and 512 KB measurement (serial) flash memory. It is very energy efficient with current draw of 8 mA in active mode and less than 15 μA in sleep mode. The user interface consists of 3 LEDs - red, green and yellow [20]. The MICAz is capable of running TinyOS 2.1.2, 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 and MICAz motes are shown in Fig 2a and 2b, respectively, while the layout of experimental premises is shown in Fig 3. 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 in a way as shown in Fig 1. The RSSI values are plotted in Fig 4 and 6 with a gain provided to all RSSI values received in order to make them positive.

(a) (b)

FIGURE 2. nodes used in the experiment. (a) Base station. (b) MICAz motes.

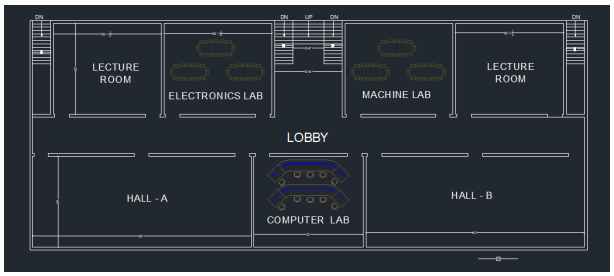

FIGURE 3. Layout of experimental premises.

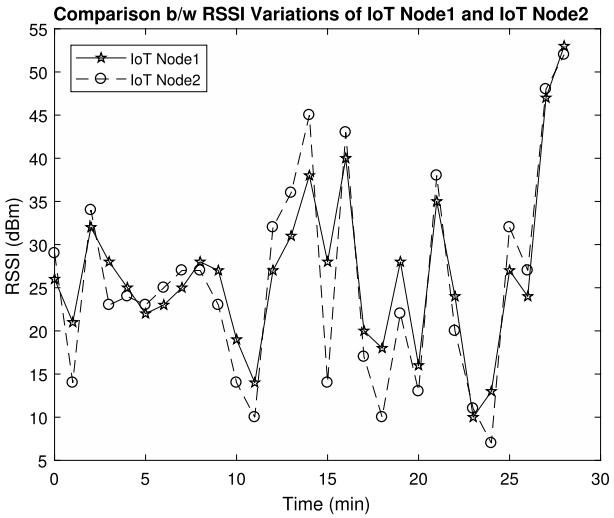

FIGURE 4. The comparison of RSSI variations of IoT node 1 and IoT node 2 when no adversarial node is present in IoT network.

B. SIMULATION RESULTS

The RSSI values acquired from MICAz motes are simulated on MATLAB R2017a. The results have been achieved for various scenarios described in Section III. Each scenario is presented below;

1) ADVERSARIAL NODE DETECTION

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

A significant improvement in results are seen by filtering out the RSSI variations. The comparative results are shown in Table 1.

Case 1 (No Adversarial Node in the Network): If there is no adversarial node present in the network then the link fingerprint will correlate at the server and we get the correlation coefficient greater than 0.95.

Fig 4 and 6 represent the RSSI variation comparison of link A and link B respectively as shown in Fig 1. IoT node 1 communicating with IoT node 2 and IoT node 2 communicating

TABLE 1. Pearson correlation coefficient (r) calculated for various cases.

Scenario	IoT node 1 and IoT node 2		IoT node 2 and IoT node 3		Confidence Interval (CI)
	r	filtered r	r	filtered r	
Case 1	0.9270	0.9614	0.8420	0.9713	95%
Case 2	-0.0038	0.0287	0.9280	0.9515	95%
Case 3	0.8913	0.9628	0.0628	0.2056	95%
Case 4	-0.0063	-0.3693	-0.1740	-0.5125	95%
Case 5	-0.2753	-0.3384	0.8369	0.9520	95%
Case 6	0.8382	0.8590	0.5269	0.7643	75%

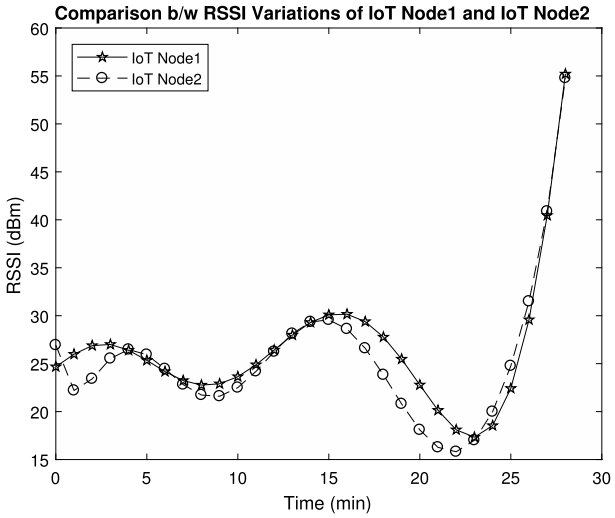

FIGURE 5. Filtered RSSI variations of IoT node 1 and IoT node 2 when no adversarial node is present in IoT network.

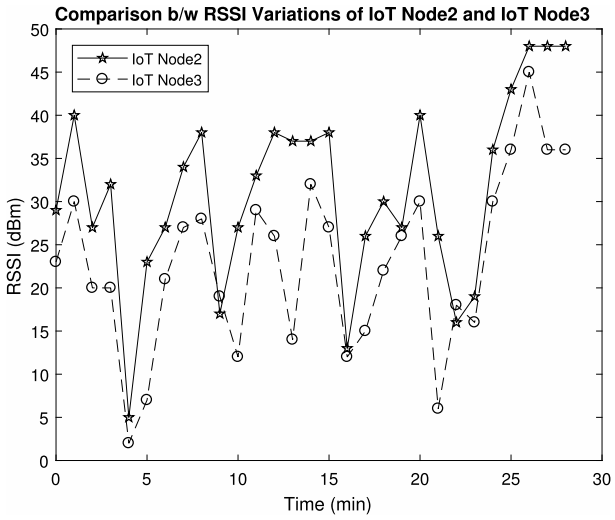

FIGURE 6. The comparison of RSSI variations of IoT node 2 and IoT node 3 when no adversarial node is present in IoT network.

with IoT node 3 are showing the highly correlated pattern. The correlation coefficients achieved are 0.9270 and 0.8420, respectively. A higher values of 0.9614 and 0.9713 are achieved by applying the filter, which further smooths down the RSSI variations. A linear relationship is observed among the RSSI variations of connected IoT nodes as shown in Fig 5 and 7. These results are achieved at the server when

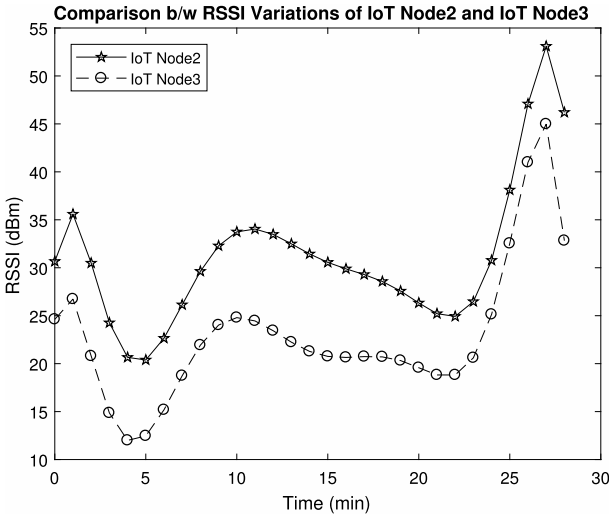

FIGURE 7. Filtered RSSI variations of IoT node 2 and IoT node 3 when no adversarial node is present in IoT network.

it decodes the encoded link fingerprints and then compares the concerned dBm values.

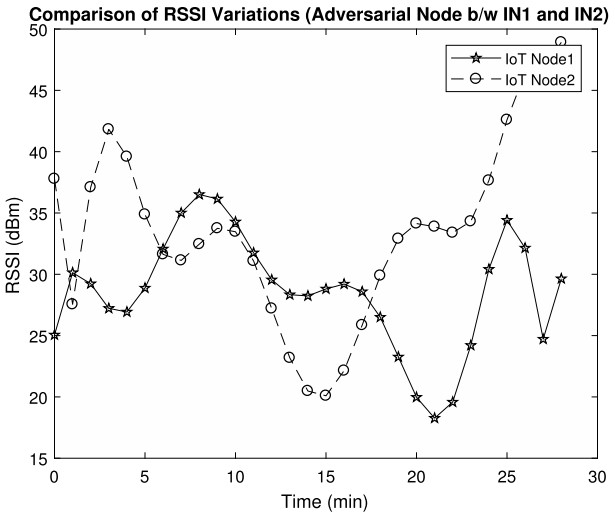

FIGURE 8. When adversarial node is present between IoT node 1 and IoT node 2. The link is 1 →AdvNode →2→3.

Case 2 (Adversarial node is present between IoT node 1 and IoT node 2): As the adversarial node is present between IoT node 1 and IoT node 2, the link fingerprints generated at IoT node 1 and IoT node 2 will be different. The uncorrelation is quite obvious in Fig 8 by observing the

relationship in RSSI variations of in-line IoT nodes. The variations relationship is more monotonic than linear. Though, high correlation is observed between the link fingerprints of IoT node 2 and IoT node 3. The correlation details are given in Table 1.

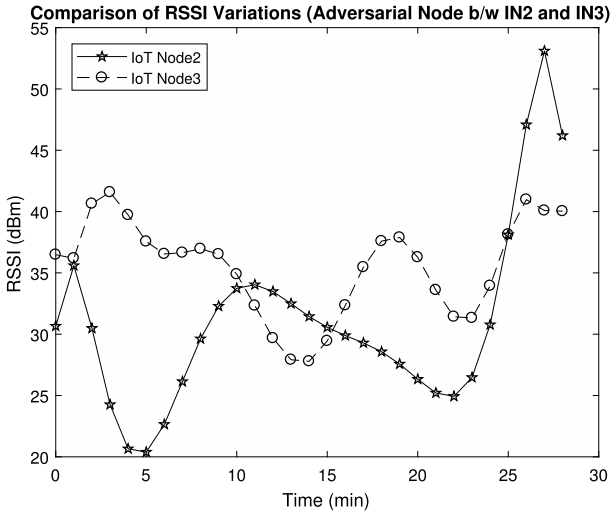

FIGURE 9. When adversarial node is present between IoT node 2 and IoT node 3. The link is 1 →2 →AdvNode →3.

Case 3 (Adversarial Node Is Present Between IoT Node 2 and IoT Node 3): When adversarial node is present between IoT node 2 and IoT node 3, all the packets reach IoT node 3 from IoT node 2 via adversarial node. The comparison of RSSI variations is presented in Fig 9. Both IoT node 2 and IoT node 3 send their respective encoded link fingerprints to the server. The server upon correlating the link fingerprints of both the IoT nodes computes correlation coefficient approximately equals to 0 as shown in Table 1. This reflects the adversarial node presence in between IoT node 2 and IoT node 3. The correlation coefficient is quite high for IoT node 1 and IoT node 2 where no adversarial node is present in between.

Case 4 (Adversarial Node Is Present Between IoT Node 2 and IoT Node 3, and IoT Node 2 and IoT Node 3): When two adversarial nodes are present in the IoT network, i.e. one between the link of IoT node 1 and IoT node 2 and other between the link of IoT node 2 and IoT node 3, then all the link fingerprints mismatch at the server because the RSSI variations comparison is uncorrelated. The reason is that they are connected to the adversarial node. The links are established through the adversarial nodes. We are getting low correlation coefficient for both links as shown in the Table 1.

Case 5 (Data Tempering): This scenario is implemented at IoT node 1 by considering that the data has been forged at IoT node 1 and the same can be applied for any other IoT node as well. If the data is forged or tempered at any of the IoT node then the link fingerprints at the server do not correlate. The server receives different RSSI link fingerprints because the original binary stream of link fingerprints are forged by the intruder. In this case, IoT node 1 sends a different link

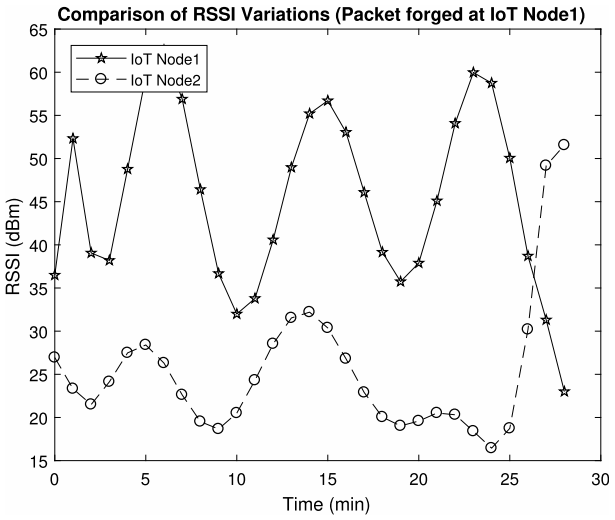

FIGURE 10. RSSI comparison of IoT node 1 and IoT node 2 when the packet at IoT node 1 is forged.

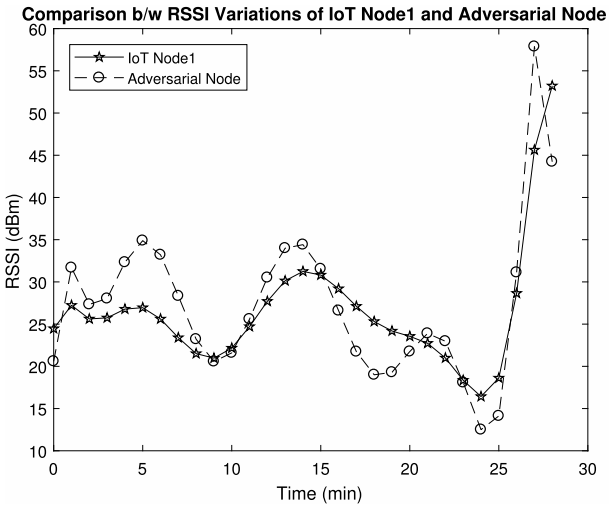

FIGURE 11. RSSI comparison of IoT node 1 and Adversarial node (IoT node 2 is replaced by Adversarial Node).

fingerprint compared to the link fingerprints of IoT node 2. Fig 10 represents the uncorrelated plot for both filtered RSSI variations. The correlation is high between the RSSI variation patterns of IoT node 2 and IoT node 3.

Case 6 (IoT Node Replaced by the Intruder): It is assumed for this case only that the adversarial node is able to send data to the server. When IoT node 1 is replaced by adversarial node, the adversarial node sends the link fingerprints to the server. The adversarial node has no information of the key to encode the data, rather it sends the unencoded data to the server. The server assumes that the link fingerprint is encoded and decodes it with the key of that node which is replaced by adversarial node. Here after performing multiple experiments, it is observed that the correlation coefficient can be high at times but not high enough to remain unnoticed. The results in Fig 11 and 12 are taken when IoT node 2 is replaced by adversarial node.

TABLE 2. Data provenance.

Scenario	Correlation of IoT node header with all available LFs at the server				Remarks
Case 1	100%	100%	100%	100%	The origin is IoT node 1
Case 2	100%	100%	100%	40.7407%	The data is tempered at IoT node 1

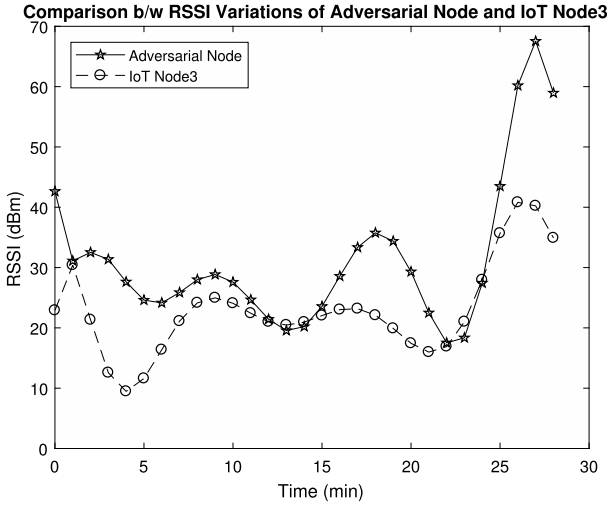

FIGURE 12. RSSI comparison of Adversarial node and IoT node 3 (IoT node 2 is replaced by Adversarial Node).

2) DATA PROVENANCE

Data provenance has been achieved using the same data received at the base station as described in subsection IV-A. Simulation is performed for two cases. They are as under,

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 as shown in Fig 1. 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 100% 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 K_2 and 100% 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% match is found with a part of all link fingerprints present at the server recived from IoT node 2 linked with IoT node 1. Now the same process is done for the next in sequence of header. A 100% 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. Table 2 shows the results obtained.

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. The process described in case 1 of subsection IV-B.2 is applied by decoding 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% rather a very low percentage of match is observed. This shows that the packet data is forged at IoT node 1.

3) TIME COMPLEXITY

The time complexity comparison is performed by calculating the computational time at node and server level. As shown in Fig 13, the time remains constant if we increase the number of RSSI samples to be quantized at node level and for correlation at the server. This shows that as the number of bits are increased the computational time is not effected. The time complexity of out system is $O(1)$ which is as better compared to other state of the art cryptographic solutions referred at [21]. Table 3 shows the comparison of time complexity of our model with other available data security algorithms.

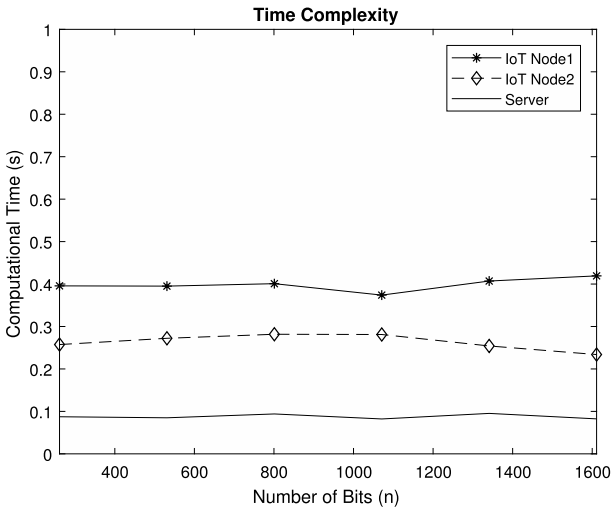

FIGURE 13. Time complexity of system at node and server level.

TABLE 3. Time complexity comparison of our system with various state of the art algorithms for data security. N is a constant that depends upon the underlaying hardware used for encryption.

AES	DES	3DES	RC4	BlowFish	Our System
O(N)	O(N)	O(N)	O(N)	O(N)	O(1)

TABLE 4. Energy consumption at IoT node level for link fingerprints transmission to the server.

IoT node (1,2,3)	Fingerprint (bytes)	Transmission Cost (μJ)	AES-128 (μJ)	SHA-1 (μJ)	ECDSA-160 (mJ)	Total (mJ)
1	16	76.8	1.83	308.0	52	52.386
2	32	153.6	3.66	616.0	52	52.773
3	16	76.8	1.83	308.0	52	52.386
Total Energy dissipated at node level						157.545

TABLE 5. Energy consumption at IoT node level for data provenance protocol.

IoT node (1,2,3)	Fingerprint (bytes)	Transmission Cost (μJ)	AES-128 (μJ)	SHA-1 (μJ)	ECDSA-160 (mJ)	Total (mJ)
1	4	19.204	0.568	0.768	52	52.020
2	8	38.408	1.144	1.536	52	52.041
3	4	19.204	0.568	0.768	52	52.020
Total Energy dissipated at node level						156.082

TABLE 6. Energy dissipated by each IoT node and by the whole network.

IoT node (1,2,3)	Energy Dissipated (mJ)
1	104.406
2	104.814
3	104.406
Network	313.626

TABLE 7. Energy dissipation comparison.

System	Packet (Bytes)	Energy Dissipated (mJ)
Our	32 (max.)	52.773
Level Crossing [7]	52	53.305
Ranking [7]	598	66.450
Raw RSSI [7]	2292	109.801

4) ENERGY CONSUMPTION

In this section, energy consumption is calculated for the system model presented. The specifications of MICAz motes are already presented in section IV-A. The standard values specified for MICAz motes are used for energy calculations. Furthermore, the energy benchmarks of MICAz motes used in the litrature are applied to the presented protocols. The energy consumption for AES-128 encryption (128 bits), SHA-1 Hash (64 bits), ECDSA-160 Sign and Transmit 1 bit are 1.83 μJ, 154 μJ, 52 μJ and 0.6 μJ respectively [7]. As the decoding is carried out at the server, the energy calculations are not done for the server. The server is not energy limited. Two scenarios are presented:

1) After every 5 minutes and 20 seconds, each IoT node sends its respective quantized and encoded RSSI values of 16 bytes to the server.
2) IoT nodes add certain bytes as headers to the payload which contain encoded link fingerprints.

Table 4 and 5 show the energy consumption at each node level when the packet is transmitted from IoT node 1 to IoT node 3 via IoT node 1 considering the hash data and session identifiers as part of protocols used previously by [7]. Table 6 shows the total energy dissipated at the node level and of overall IoT network when the nodes are performing in full capacity. It can be seen from Table 7 that previously used techniques generate link fingerprints of a larger length due to which the energy consumption is more compared to the mechanism provided in this paper. By applying various optimization techniques, the link fingerprint can be further reduced.

V. 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. 1250–1258, 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. 682–685.

[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. 1327–1340, 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. 700–705.

[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. 1–6.

[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. 84–88.
[7] S. T. Ali, V. Sivaraman, D. Ostry, G. Tsudik, and S. Jha, "Securing first-hop data provenance for bodyworn devices using wireless link fingerprints," *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 12, pp. 2193–2204, Dec. 2014.
[8] K. Zhang, X. Liang, R. Lu, and X. Shen, "Sybil attacks and their defenses in the Internet of Things," *IEEE Internet Things J.*, vol. 1, no. 5, pp. 372–383, Oct. 2014. [Online]. Available: http://ieeexplore.ieee.org/document/6868197/
[9] G. Kecskemeti, G. Casale, D. N. Jha, J. Lyon, and R. Ranjan, "Modelling and simulation challenges in Internet of Things," *IEEE Cloud Comput.*, vol. 4, no. 1, pp. 62–69, Jan./Feb. 2017. [Online]. Available: http://ieeexplore.ieee.org/document/7879128/
[10] J. Pacheco and S. Hariri, "IoT security framework for smart Cyber infrastructures," in *Proc. IEEE Int. Workshops Found. Appl. Self Syst.*, Sep. 2016, pp. 242–247.
[11] Z. Bohan, W. Xu, Z. Kaili, and Z. Xueyuan, "Encryption node design in Internet of Things based on fingerprint features and cc2530," in *Proc. IEEE Int. Conf. Green Comput. Commun., Internet Things, IEEE Cyber, Phys. Soc. Comput.*, Aug. 2013, pp. 1454–1457.
[12] J.-Y. Lee, W.-C. Lin, and Y.-H. Huang, "A lightweight authentication protocol for Internet of Things," in *Proc. Int. Symp. Next-Gener. Electron. (ISNE)*, May 2014, pp. 1–2.
[13] M. N. Aman, K. C. Chua, and B. Sikdar, "Secure data provenance for the Internet of Things," in *Proc. 3rd ACM Int. Workshop IoT Privacy, Trust, Secur.*, 2017, pp. 11–14. [Online]. Available: http://dl.acm.org/citation.cfm?doid=3055245.3055255
[14] J. Qian, H. Xu, and P. Li, "A novel secure architecture for the Internet of Things," in *Proc. Int. Conf. Intell. Netw. Collaborative Syst. (INCoS)*, Sep. 2016, pp. 398–401. [Online]. Available: http://ieeexplore.ieee.org/document/7695208/
[15] Y. Xie and D. Wang, "An item-level access control framework for inter-system security in the Internet of Things," *Appl. Mech. Mater.*, vols. 548–549, pp. 1430–1432, Apr. 2014. [Online]. Available: https://www.scientific.net/AMM.548-549.1430
[16] Q. Wang, D. Chen, N. Zhang, Z. Qin, and Z. Qin, "LACS: A lightweight label-based access control scheme in IoT-based 5G caching context," *IEEE Access*, vol. 5, pp. 4018–4027, 2017. [Online]. Available: http://ieeexplore.ieee.org/document/7872420/
[17] Z. Zhou, C. Gao, C. Xu, Y. Zhang, S. Mumtaz, and J. Rodriguez, "Social big-data-based content dissemination in Internet of vehicles," *IEEE Trans. Ind. Inform.*, vol. 14, no. 2, pp. 768–777, Feb. 2018. [Online]. Available: http://ieeexplore.ieee.org/document/7995077/
[18] Z. Zhou, K. Ota, M. Dong, and C. Xu, "Energy-efficient matching for resource allocation in D2D enabled cellular networks," *IEEE Trans. Veh. Technol.*, vol. 66, no. 6, pp. 5256–5268, Jun. 2017. [Online]. Available: http://ieeexplore.ieee.org/document/7585029/
[19] Z. Zhou, H. Yu, C. Xu, Y. Zhang, S. Mumtaz, and J. Rodriguez, "Dependable content distribution in D2D-based cooperative vehicular networks: A big data-integrated coalition game approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 3, pp. 953–964, Mar. 2018. [Online]. Available: http://ieeexplore.ieee.org/document/8264740/
[20] *Micaz-Wireless Measurement System*, Crossbow Technol., Milpitas, CA, USA, Apr. 2007.
[21] C.-L. Wu and C.-H. Hu, "Computational complexity theoretical analyses on cryptographic algorithms for computer security application," in *Proc. IEEE 3rd Int. Conf. Innov. Bio-Inspired Comput. Appl. (IBICA)*, Sep. 2012, pp. 307–311.

MOHSIN KAMAL (M'16) received the B.S. degree in telecommunication engineering from the National University of Computer and Emerging Sciences, Peshawar, Pakistan, in 2008, and the M.S. degree in electrical engineering from the Blekinge Tekniska Högskola, Karlskrona, Sweden, in 2012. He is currently pursuing the Ph.D. degree in electrical engineering with the National University of Computer and Emerging Sciences.

Since 2013, he has been an Assistant Professor with the National University of Computer and Emerging Sciences, where he is the IEEE Student Branch Counselor since 2016. His research interests include the development of light-weight solutions for various IoT applications, cooperative communication, and cognitive radio networks.

MUHAMMAD TARIQ (S'08–M'12–SM'17) received the M.S. degree from Hanyang University, South Korea, as an HEC Scholar, the Ph.D. degree from Waseda University, Japan, in 2012, as a Japanese Government (MEXT) Scholar, and postdoc from Princeton University in 2016 as a Fulbright Scholar, under the supervision of Prof. H. V. Poor. He was the Head of the Department of Electrical Engineering, National University of Computer and Emerging Sciences, Peshawar Campus, where he is currently the Director. He has authored or co-authored over 50 research articles and co-authored a book on smart grids with leading researchers from Europe, China, Japan, and USA, which was published by John Wiley and Sons in 2015. He has received many awards for his work. He has presented his research work in various IEEE flagship conferences held around the world. In 2017, the Chinese Government selected him as the High End Foreign Expert through the International Cooperation Project funded by the State Administration of Foreign Experts Affairs, China. He has delivered research talks as a guest/invited/keynote speaker at various forums and universities in Pakistan, China, Saudi Arabia, and USA. He is the Programs' Evaluator of the Pakistan Engineering Council. He rendered his technical committee services in various IEEE flagship conferences and transactions.