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

D.SARITHA  
Department of Master of Computer Applications, Narayana Engineering College, Gudur.

K.V. KIRAN KUMAR  
Assistant Professor, Dept. of Master of Computer Applications, Narayana Engineering College, Gudur.

Abstract— Due to limited resources and scalability, security protocols for Internet of Things (IoT) need to be lightweight. Cryptographic solutions are not feasible 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 multi-hop IoT network. The Received Signal Strength Indicator (RSSI) of communicating IoT nodes are used to generate fingerprints at the server. The fingerprints are matched at the server to compute the correlation coefficient. Higher the value of correlation coefficient, higher the percentage of secured data transfer. 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 IoT nodes and overall network. The results show that the energy consumption of the system presented in this paper is 52 mJ to 53 mJ for each IoT node and 313.626 mJ for the entire network. 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.

INTRODUCTION

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 IoT. 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 non-linear fractional programing is also established.

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. H. Ritzdorf is .Security primitives included 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]. and each file access request must be granted by a pre-arranged threshold of t owners. We remark that existing cloud platforms, such as Amazon S3 or Dropbox, provide no support for shared ownership policies, and offer only basic 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.

System Model

- 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. The solution proposed in this paper incorporates lightweight 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.

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, 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 lightweight 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 intensive 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 Uncloneable 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 lightweight messaging protocol called MQTT by 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 help stop the threat. 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 name bedded label, where only the authorized fog node has access to the file. Hecaching 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 gametheoretic approach. Mutual preference based on nonlinear fractional programming 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: 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 an adversarial node 5) The server is not secured in a way that an 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 indBm ranging from 48 dBm to 20 dBm. The signal strength is calculated using Friis transmission equation which states that 

\[ P_r = P_t G_t G_r L_p \]

where, \( P_r \) is the received power, \( P_t \) represents the transmitted power, \( G_r \) and \( G_t \) are the receiving and transmitting antennas.

Path loss is expressed as:

\[ L_p = (4\pi d \lambda)^2, \]

(2) where \( d \) is the distance between two communicating IoT nodes, \( \lambda \) 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

![FIGURE 1: System Model](image)

![FIGURE 2: nodes used in the experiment](image)
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 \( \Delta \), which is given as, 
\[
\Delta = \frac{\text{Pr}(\text{max}) - \text{Pr}(\text{min})}{L}.
\]
(3) \( \text{Pr}(\text{max}) \) and \( \text{Pr}(\text{min}) \) 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. \( \text{LF}_{\text{encoded}}(1\rightarrow n) = \text{LF}_{1\rightarrow n} \oplus K_i \). (4) In 4, \( \oplus \) represents logical exclusive-OR operation, whereas \( \text{LF}_{\text{encoded}} \) is the encoded link fingerprint. Each IoT node sends \( \text{LF}_{\text{encoded}} \) 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 an 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, 
\[
\text{LF}_{\text{new}}[i] = \text{bin-dec conversion}(\text{LF}_{\text{encoded}}[i],\text{Keynode}(i)).
\]
(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 (\( \rho \)). If the value is between 0.8 and 1 then it is considered as highly correlated in a multi-hop network. Mathematically,
\[
\rho_X,Y = \frac{\text{cov}(X,Y)}{\sigma_X \sigma_Y}
\]
where, \( \text{cov} \) is the covariance and \( \sigma \) represents the standard deviation. A simplified equation can be written as;
\[
\rho = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \bar{X})^2 \sum_{i=1}^{n}(Y_i - \bar{Y})^2}}
\]
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 \( \rho \) returns a value in \([-1:1]\) where 1 indicates perfect correlation, 0 indicates no correlation, and -1 indicates anti-correlation. Algorithm 1 Link Fingerprint generation and encoding at IoT Node.

1. Initialize the IoT node
2. Read the RSSI values from adjacent IoT node
3. \( \text{LF}_{\text{new}}[i] = \text{LF}_{\text{new}}[i] + \text{gain} \)
4. Quantize \( \text{LF}_{\text{new}}[i] \)
5. Assign binary code-word to Quantized \( \text{LF}_{\text{new}}[i] \)
6. \( \text{LF}_{\text{encoded}}[i] = \text{XOR}(\text{LF}_{\text{new}}[i],\text{Keynode}(i)) \)
7. Send a copy to the server

Algorithm 2 Adversarial node’s detection at the server.

1. Initialize the server
2. Correlate the LFs of adjacent IoT nodes
3. If the correlation coefficient \( \rho \) has values between -1 and 1, then return No adversarial node is present else if \( 0.9 < \rho \leq 1 \) then return The RSSI values are not correctly measured
4. 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

This work is licensed under a Creative Commons Attribution 4.0 International License
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.

**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 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: 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.

![Comparison b/w RSSI Variations of IoT Node1 and IoT Node2](image)

While finding the origin of data, if an 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

http://indusedu.org

This work is licensed under a Creative Commons Attribution 4.0 International License
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 fingerprint 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 the packet adds its link fingerprint. The server checks the header for the origin and gets a 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 in IoT networks

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 an 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 base station.
values to the static base station. The MICAz motes do not cross each other.

The FIGURE 3: Layout of experimental premises 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.

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

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.
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.

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.

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 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.
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.

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

This work is licensed under a Creative Commons Attribution 4.0 International License.
observed that the correlation coefficient can be high at times but not high enough to remain unnoticed. The results in Fig. 1 and 12 are taken when 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 header matches 100% with 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 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 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% 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.

Table 1: Pearson correlation coefficient (r) calculated for various cases

<table>
<thead>
<tr>
<th>Scenario</th>
<th>IoT node 1 and IoT node 2</th>
<th>IoT node 2 and IoT node 3</th>
<th>Confidence Interval (CI)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case 1</td>
<td>0.9270</td>
<td>0.9614</td>
<td>0.8420</td>
</tr>
<tr>
<td>Case 2</td>
<td>-0.0038</td>
<td>0.0287</td>
<td>0.9280</td>
</tr>
<tr>
<td>Case 3</td>
<td>0.8913</td>
<td>0.9628</td>
<td>0.0628</td>
</tr>
<tr>
<td>Case 4</td>
<td>-0.0063</td>
<td>-0.3693</td>
<td>-0.1740</td>
</tr>
<tr>
<td>Case 5</td>
<td>-0.2753</td>
<td>-0.3384</td>
<td>0.8369</td>
</tr>
<tr>
<td>Case 6</td>
<td>0.8832</td>
<td>0.8590</td>
<td>0.5269</td>
</tr>
</tbody>
</table>

The last header is the first inserted header from IoT node which is received at IoT node 3. 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-B2 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

Table 2: Pearson correlation coefficient (r) calculated for various cases

<table>
<thead>
<tr>
<th>Scenario</th>
<th>IoT node 1 and IoT node 2</th>
<th>IoT node 2 and IoT node 3</th>
<th>Confidence Interval (CI)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case 1</td>
<td>0.9270</td>
<td>0.9614</td>
<td>0.8420</td>
</tr>
<tr>
<td>Case 2</td>
<td>-0.0038</td>
<td>0.0287</td>
<td>0.9280</td>
</tr>
<tr>
<td>Case 3</td>
<td>0.8913</td>
<td>0.9628</td>
<td>0.0628</td>
</tr>
<tr>
<td>Case 4</td>
<td>-0.0063</td>
<td>-0.3693</td>
<td>-0.1740</td>
</tr>
<tr>
<td>Case 5</td>
<td>-0.2753</td>
<td>-0.3384</td>
<td>0.8369</td>
</tr>
<tr>
<td>Case 6</td>
<td>0.8832</td>
<td>0.8590</td>
<td>0.5269</td>
</tr>
</tbody>
</table>

Table 2: Data Provenance

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Correlation of IoT node header with all available LPs at the server</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case 1</td>
<td>100% 100% 100% 100% 100%</td>
<td>The origin is IoT node 1</td>
</tr>
<tr>
<td>Case 2</td>
<td>100% 100% 100% 100% 100%</td>
<td>The data is tampered at IoT node 1</td>
</tr>
</tbody>
</table>

A user has read access to the file if and only if he has read access to the file’s tokens by at least t owners. We refer to such a user as an “authorized reader”.

To securely enforce shared ownership policies, Commune is designed to fulfill the following properties.

- **P1**: A malicious writer (i.e., a user who has been granted write access by fewer than t owners), must not be able to publish a file F as if F were authored by an authorized writer.

- **P2**: A malicious reader (i.e., a user who has been granted read access to a file F by fewer than t owners), must not be able to recover the file content. This property must also hold in case of revocation. Assume that, at the time $\tau_1$, $U$ has read access to F granted by at least t owners. Also assume that, at the time $\tau_2 > \tau_1$, $U$ has his access rights revoked. This happens if, at the time $\tau_2$, some of the owners decide to revoke read access to $U$ so that $U$ is left with fewer than t read grants. We must ensure that, starting from time $\tau_2$, $U$ cannot recover meaningful bits of F. We remark that, as is common for access
control systems, we cannot prevent $U$ from storing a local copy of $F$ at the time $t_1$ and reading it even after his read right has been revoked. Commune must also provide collusion resistance. That is, coalitions of users—where no single user is an authorized reader—must not be able to pool their credentials to escalate their read access rights.

Property P1 ensures protection against malicious writers who try to disseminate content despite lacking the required credentials. Property P2 guarantees that malicious readers cannot read content written to the shared repository.

![RSSI comparison of Adversarial node and IoT node 3 (IoT node 2 is replaced by Adversarial Node).](image)

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

![Time complexity of system at node and server leve](image)

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

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 packets 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 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 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 received from IoT node2linkedwithIoTnode1.Nowthesameprocessisdone for the next in sequence of header. A 100% match in link fingerprints from the header with part of IoT node 1’s link fingerprintsisachieved.Bynow,alltheheadersequencesare 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 node3intheend.Table2showstheresults 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-B2 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.

<table>
<thead>
<tr>
<th>Case</th>
<th>Packet Situation</th>
<th>Process Description</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Packet is not forged</td>
<td>Decoding header in sequence</td>
<td>100% match</td>
</tr>
<tr>
<td>2</td>
<td>Packet is forged</td>
<td>Decoding header in sequence</td>
<td>Low percentage of match</td>
</tr>
</tbody>
</table>

3) Time Complexity The time complexity is calculated by comparing computational time at node and server level. The time remains constant when the number of bytes to be transmitted increases. This shows that the packet data is not forged at IoT node 1.

4) Energy Consumption In this section, energy consumption is calculated for the system model presented. The time complexity is calculated by comparing computational time at node and server level. The time remains constant when the number of bytes to be transmitted increases. This shows that the packet data is not forged at IoT node 1.

Table 4 shows the energy consumption at IoT node level for link fingerprints transmission to the server.

<table>
<thead>
<tr>
<th>IoT node</th>
<th>Fingerprint (bytes)</th>
<th>Transmission Cost (µJ)</th>
<th>AES-128 (µJ)</th>
<th>SHA-1 (µJ)</th>
<th>ECDSA-160 (µJ)</th>
<th>Total (µJ)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>16</td>
<td>76.8</td>
<td>1.83</td>
<td>300.0</td>
<td>52</td>
<td>32,306</td>
</tr>
<tr>
<td>2</td>
<td>32</td>
<td>151.6</td>
<td>3.66</td>
<td>616.0</td>
<td>52</td>
<td>32,773</td>
</tr>
<tr>
<td>3</td>
<td>16</td>
<td>76.8</td>
<td>1.83</td>
<td>300.0</td>
<td>52</td>
<td>32,306</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>97,385</td>
</tr>
</tbody>
</table>

Table 5 shows the energy consumption at IoT node level for data provenance protocol.

<table>
<thead>
<tr>
<th>IoT node</th>
<th>Fingerprint (bytes)</th>
<th>Transmission Cost (µJ)</th>
<th>AES-128 (µJ)</th>
<th>SHA-1 (µJ)</th>
<th>ECDSA-160 (µJ)</th>
<th>Total (µJ)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>19.204</td>
<td>0.98</td>
<td>0.98</td>
<td>52</td>
<td>52,020</td>
</tr>
<tr>
<td>2</td>
<td>8</td>
<td>31.408</td>
<td>1.14</td>
<td>1.56</td>
<td>52</td>
<td>52,041</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>19.204</td>
<td>0.98</td>
<td>0.98</td>
<td>52</td>
<td>52,020</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>156,082</td>
</tr>
</tbody>
</table>

when the packet is checked for IoT node 1, the match is 100% otherwise a lower percentage of match is observed. This shows that the packet data is not forged at IoT node 1.
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.

**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 lightweight 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**


MOHSIN KAMAL
(M’16) received the B.S. degree in Telecommunication Engineering from National University of Computer & Emerging Sciences, Peshawar, Pakistan, in 2008 and the M.S. degree in Electrical Engineering from Blekinge Tekniska Högskola, Karlskrona, Sweden, in 2012. He is currently pursuing the Ph.D. degree in Electrical Engineering at National University of Computer & Emerging Sciences, Peshawar, Pakistan. Since 2013, he is working as Assistant Professor at National University of Computer & Emerging Sciences, Peshawar, Pakistan. He is also the IEEE student branch counselor at the same institute 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) is the Director of FAST National University of Computer & Emerging Sciences (NUCES) Peshawar Campus. Before taking charge as a director, he was head of Department of Electrical Engineering in the same university. He is the author/co-author of 50+ research articles, and received many awards for his work. He has presented his research work in various IEEE flagship conferences held around the world. He rendered his Technical Committees services in various IEEE flagship conferences and transactions. He has authored a book on smart grid with leading researchers from Europe, China, Japan, and the U.S., which was published by John Wiley and Sons in 2015. In 2017, Chinese government selected him as High End Foreign Expert through International Cooperation Project funded by State Administration of Foreign Experts Affairs China. Dr. Tariq completed his Postdoc from Princeton University as a Fulbright scholar under the supervision of Prof. H. Vincent Poor in 2016. Dr. Tariq completed his PhD as a Japanese govt (MEXT) scholar, from Waseda University, Japan in 2012. He completed his MS from Hanyang University, South Korea as an HEC scholar. He has delivered research talks as guest/invited/keynotespeaker at various forums and universities in Pakistan, China, Saudi Arabia, and the U.S. He is the programs’ evaluator (PEV) of Pakistan Engineering Council.