# Detecting Selective Forwarding using Sentinels in Clustered IoT Networks

Rohini Poolat Parameswarath, Cheng Yujun Eugene, Nalam Venkata Abhishek, *Student Member, IEEE*, Teng Joon Lim, *Fellow, IEEE* and Biplab Sikdar, *Senior Member, IEEE*

***Abstract*—Compromised relays in clustered IoT networks can be used to launch attacks that cannot be easily detected using the traditional security algorithms. In this paper, we consider an attack where a compromised relay deliberately drops the packets received from the IoT devices it serves. Such an attack causes the IoT devices to retransmit more frequently, thereby increasing their processing load. As a result, their batteries will drain at a faster rate. The difficulty in differentiating a genuine packet drop event from a malicious packet drop event makes it necessary to develop a novel Intrusion Detection System (IDS) specially tailored for detecting such an attack. The IDS is installed in a special node called a sentinel, which monitors the network. The sentinel also estimates the packet retransmission rate of the IoT devices, a parameter required for the IDS. The effectiveness of the system is demonstrated experimentally on a clustered network.**

## I. Introduction

With the ability to automate everything around us and its business potential, the popularity of the Internet of Things (IoT) is increasing day by day. Many applications like smart grid, smart home, intelligent transport system, etc. can be realized using IoT [1]. With more than fifty billion devices (majority IoT devices) estimated to be connected to the Internet by 2050 [2], radio access networks can get heavily congested. However, the increased usage of IoT devices has introduced many challenges like scalability, security, etc. [1], [3], [4]. To address the scalability issue, one of the prominent solutions is the clustering approach [5]. In such an approach, a set of IoT devices are clustered and are assigned to a relay which would assist in forwarding the traffic to and from the main access point. The clustering strategy could be based on Quality of Service requirements, location, etc.

Due to hardware constraints, IoT devices lack security mechanisms and are vulnerable to various network attacks. Many of these attacks can be defended against through cryptography or through resilient software update patches [6]. However, clustered IoT networks are vulnerable to attacks that can be launched at the cluster head or gateway, like selective forwarding, channel degradation, black hole, etc. that cannot be addressed using cryptography or software patches. Such attacks affect the quality of service of the network and can have an adverse impact in networks within healthcare, transportation systems and other essential services.

These attacks can be realised if the attacker obtains root access to relays, which would often be user-installed equipment that are not professionally maintained and updated. Also, these devices present many security flaws and various software vulnerabilities. This can lead to an attacker gaining remote administration capabilities. These attacks are hard to distinguish from naturally occurring events, making them difficult to detect. It can also be seen that an adversary can adversely impact a set of IoT devices merely by compromising the relay they are associated with. Hence, developing Intrusion Detection Systems (IDSs) to detect such attacks is an important task, in addition to other defenses like secured pairing, integrity verification and secured architectures. IDS's can be categorized according to where most of the detection intelligence resides – centralized, distributed, or hybrid [14]. In this paper, we propose a centralized IDS system which relies on the rate of packet retransmissions of the IoT devices on the uplink estimated by a trusted server, which will then decide whether a relay has been compromised.

### A. Related Work

Researchers in the past have tried to investigate the behavior of adversaries, that compromise relays and disrupt the communication in clustered IoT networks. Machine learning algorithms (such as the ones in [7]), when designed using sufficient and appropriate training data samples, can provide the desired performance. However, in reality, it is problematic to inject malicious packets into the networks to build the training data. The authors in [8] proposed a detection technique called SVELTE to detect the presence of a selective forwarding attack. The proposed system detects the adversary when it filters all the packets or sends only the mapping request packets. In [9], the authors have presented an approach based on the channel conditions to detect selective forwarding attacks. A similar approach was proposed in [10] to detect forwarding misbehavior of nodes. However, a sensor monitoring the data packets of the forwarding nodes can be expensive in terms of the energy consumed. A sequential probability ratio based detection system was presented in [11] for detecting selective forwarding attacks. Their decision is based on the expected transmission count of the nodes. A light-weight heart-beat protocol is proposed in [12]. In this approach, an echo is sent to every node in the network and the attack is detected when there is no reply received from the affected nodes.

### B. Our Contribution

Our previous work in [13] describes the IDS in detail, however the results are limited to simulations. In this paper,

This research is supported by the National Research Foundation, Prime Minister's Office, Singapore under its Corporate Laboratory@University Scheme, National University of Singapore, and Singapore Telecommunications Ltd.

we have implemented the above IDS on an IoT network with one IoT device. We considered that the compromised relay is affecting the link between the relay and the IoT device i.e. through selective packet forwarding. We used a sentinel node to monitor the network and the IDS algorithm is based on Generalized Likelihood Ratio Test, as described in [13]. The algorithm requires the number of packets retransmitted by the IoT device. One key difference between the present paper and [13] is that we use the sentinel node to calculate the number of packets retransmitted by the IoT device and do not depend on the resource-constrained IoT device to calculate and transmit the same using the side channel. The sentinel sniffs all the packets being transmitted by the IoT device, and since a re-transmission can be differentiated from a first transmission, the number of retransmitted packets can be estimated.

## II. System Model

In this section we summarize the detection system in [13], obtain the sufficient statistics for the adversary parameter estimation and derive the bias of the estimator. We also compare the performance of the detection system with a scheme that does not require adversary parameter estimation.

### A. Network and Adversary Model

A clustered network with a set of $M$ IoT devices, $\mathcal{D} = \{D_j, j = 1, 2, \cdots, M\}$ was used for deriving the detection algorithm in [13]. The network is illustrated in Figure 1. The natural retransmission rate of the IoT device $D_j$ is assumed to be known and denoted by $\alpha_j$.

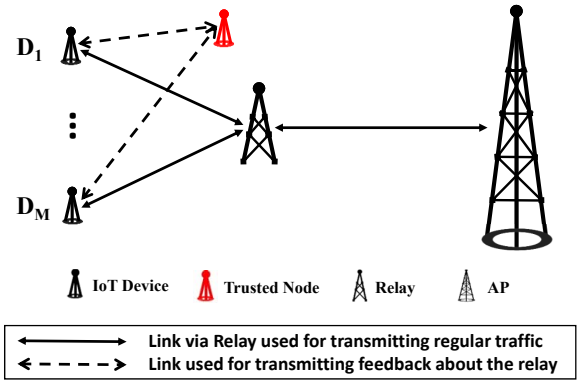


Fig. 1: Network Model

We now describe how an adversary, who has compromised the relay, disrupts the communication in the network. Once an adversary gains access to the relay, it is possible to implement Man-in-the-middle attacks like passive eavesdropping, selective forwarding (i.e. drop packets deliberately), data integrity attack (i.e. with some probability modify the payload), etc. Data Integrity attacks can be handled using efficient techniques like the one in [14]. In this paper, we assume that the adversary implements the selective forwarding attack. The probability of the relay maliciously dropping packets (i.e. attack probability) received from $D_j$ is unknown but assumed to be a constant, and is denoted by $\delta_j$. In the presence of such an adversary, the retransmission rate will increase to

$$\beta_j = \delta_j + (1 - \delta_j)\alpha_j \tag{1}$$

for $j \in \{1, 2, \cdots, M\}$. This affects the quality of communication in the network and resembles severe channel degradation, with the severity increasing with $\delta_j$.

### B. Intrusion Detection System

From (1), it can be established that $\beta_j > \alpha_j$ for $\delta_j > 0$. Therefore, in the presence of the adversary described in Section II-A, the retransmission rate increases and is a strong indicator that the attack is being implemented. Therefore, the detection system should be based on the retransmission rate of the IoT device as discussed in [13]. The IDS performs a binary hypothesis test:

- Hypothesis $H_1$: Relay is compromised and deliberately dropping packets
- Hypothesis $H_0$: Relay is not compromised and is in normal operation.

The number of packets retransmitted by $D_j$ out of the past $K$ packets is given by $n_j$. The estimated attack probability of $\delta_j$ is given as:

$$\hat{\delta}_j = \max\left(0, \frac{\frac{n_j}{K} - \alpha_j}{1 - \alpha_j}\right). \tag{2}$$

The detection algorithm in [13] decides in favor of hypothesis $H_1$ when

$$S = \sum_{j=1}^{M} S_j > \log(\gamma) = \Gamma \tag{3}$$

where $S_j = n_j \log\left(\frac{n_j(1-\alpha_j)}{\alpha_j(K-n_j)}\right) + K \log\left(\frac{K-n_j}{K(1-\alpha_j)}\right)$ if $\frac{n_j}{K}$ is greater than $\alpha_j$ else $S_j = 0$.

### C. Sufficient Statistic

In this section, we prove that the number of packets retransmitted is a sufficient statistic for estimating the attack probability. The probability mass function (PMF) of the number of packets retransmitted out of $K$ packets transmitted, in the presence of attack, by device $D_j$ is

$$P(N_j = n|H_1) = \binom{K}{n}(\beta_j)^n(1 - \beta_j)^{(K-n)}. \tag{4}$$

The same can be re-written as

$$\begin{aligned} P(N_j = n|H_1) =& \binom{K}{n}(\alpha_j + \delta_j - \alpha_j\delta_j)^n \\ & \times (1 - \alpha_j)^{(K-n)}(1 - \delta_j)^{(K-n)} \\ =& g(T(n), \delta_j) \times h(n) \end{aligned}$$

$$P(N_j = n|H_1) = g(T(n), \delta_j) \times h(n) \tag{5}$$

where the functions $g(T(n), \delta_j)$, $h(n)$ and $T(n)$ are defined as

$$g(T(n), \delta_j) = \left( (1-\delta_j)^K \left( \frac{\alpha_j}{1-\delta_j} + \alpha_j \right)^n \right) \quad (6)$$

$$h(n) = \left( \binom{K}{n} (1-\alpha_j)^{(K-n)} \right) \quad (7)$$

$$T(n) = n \quad (8)$$

Clearly, using Neyman-Fisher Factorization theorem [15], $T(n) = n$, i.e. the number of packets retransmitted, is a sufficient statistic for estimating $\delta_j$, the attack probability.

### D. Bias of the Estimator

The mean ($\hat{\mu}_j$) of $\hat{\delta}_j$ can be written as

$$\hat{\mu}_j = \sum_{n=\lfloor K\alpha_j \rfloor + 1}^{K} \frac{\frac{n_j}{K} - \alpha_j}{1-\alpha_j} \times P(N_j = n | H_1) \quad (9)$$

where $\lfloor y \rfloor$ is the largest integer not larger than $y$. Since obtaining a closed form expression for $\hat{\mu}_j$ is difficult, we calculate a bound on the mean. Say, $\hat{\delta'}_j = \frac{\frac{n_j}{K} - \alpha_j}{1-\alpha_j}$ which implies that $\hat{\delta}_j = \max(0, \hat{\delta'}_j)$. It can be seen that $\hat{\delta'}_j \leq \hat{\delta}_j$ which implies that $E[\hat{\delta'}_j] \leq E[\hat{\delta}_j]$. Hence, the bound on the mean is as follows:

$$\hat{\mu}_j \geq \frac{\frac{E[n_j]}{K} - \alpha_j}{1-\alpha_j} \quad (10)$$

$$\Rightarrow \hat{\mu}_j \geq \frac{\beta_j - \alpha_j}{1-\alpha_j} \quad (11)$$

$$\Rightarrow \hat{\mu}_j \geq \delta_j \quad (12)$$

The estimator is biased since the average of the estimator is greater than the true value of the parameter. The bias, $b(\delta_j)$, of the estimator is given as follows:

$$b(\delta_j) = \hat{\mu}_j - \delta_j \quad (13)$$

For sufficiently large $K$, the binomially distributed variable $N_j, j \in \{1, \cdots, M\}$ become approximately Gaussian due to the Central Limit theorem. In other words, $N_j \sim \mathcal{N}(K\beta_j, K\beta_j(1-\beta_j))$, as $K \to \infty$. Using this, $\hat{\mu}_j$ can be approximated as

$$\hat{\mu}_j = \int_{K\alpha_j}^{\infty} \left( \frac{\frac{n_j}{K} - \alpha_j}{1-\alpha_j} \right) \mathcal{N}(K\beta_j, K\beta_j(1-\beta_j)). \quad (14)$$

Solving the integral in (14) and substituting it in (13) we obtain the below.

$$b(\delta_j) = \sqrt{\frac{\beta_j(1-\beta_j)}{2\pi K(1-\alpha_j)^2}} \exp\left( \frac{-\delta_j}{\beta_j(1-\delta_j)} \right) - \delta_j Q\left( \frac{\sqrt{K}(\beta_j - \alpha_j)}{\sqrt{\beta_j(1-\beta_j)}} \right) \quad (15)$$

where,

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^{\infty} \exp\left( -\frac{u^2}{2} \right) du.$$

It can be observed that for a higher $K$, the bias is approximately zero. Hence, we can establish that for higher values of $K$ the estimator in (2) is unbiased.

### E. Comparison with Rao's Test

In this section, we compare the detection algorithm obtained using GLRT against a scheme based on Rao's Test [16]. Rao's test is computationally simple as compared to GLRT since we need not estimate the MLE under hypothesis $H_1$. The detection algorithm is obtained as follows:

1) The joint probability distribution of the variables $n_j, j \in \{1, 2, \cdots, M\}$, given $\delta_j, j \in \{1, 2, \cdots, M\}$, is:
$$f(\mathbf{D}) = \prod_{j=1}^{M} P(N_j = n_j; \delta_j) \quad (16)$$
where $\mathbf{N} = [N_1, \cdots, N_M]$, $\mathbf{n} = [n_1, \cdots, n_M]$, $\mathbf{D} = [\delta_1, \cdots, \delta_M]$ and $P(N_j = n; \delta_j) = \binom{K}{n}(\beta_j)^n (1-\beta_j)^{(K-n)}$.
2) The Rao's test decides $H_1$ if
$$\frac{\partial f(\mathbf{D})}{\partial \mathbf{D}}\Big|_{\mathbf{D}=\mathbf{0}}^{T} \mathbf{I}^{-1}(\mathbf{0}) \frac{\partial f(\mathbf{D})}{\partial \mathbf{D}}\Big|_{\mathbf{D}=\mathbf{0}} > \Gamma_r \quad (17)$$
where $\mathbf{I}(\mathbf{0})$ is the Fisher information under $H_0$ and $\Gamma_r$ is a pre-defined threshold.
3) After simplifying (17), we establish that the network is under attack iff
$$\sum_{j=1}^{M} \frac{(n_j - K\alpha_j)^2}{K(1-\alpha_j)} > \Gamma_r. \quad (18)$$

Considering the expressions in (3) and (18), it can be established that the average number of operations required for GLRT based test is greater than the number required by Rao's test. We now compare the performances of both these tests. The performance characteristics (i.e. false alarm and missed detection probabilities) of the detection scheme in (3) and the detection scheme in (18) are plotted in Figure 2. The result was generated using MATLAB. We used a network setup with one access point, one relay and eight IoT devices associated with the relay. The value of $K$ is 100. The simulated natural PDPs and attack probabilities on every device are given in Table I. The probability that $H_1$ is decided in the absence of attack is called the false alarm probability and the probability that $H_0$ is decided in the presence of attack is called the missed detection probability. To calculate the simulated false alarm probability, $P_{FA}$, the following steps were followed:

- We setup the network using $H_0$, i.e., all the values of $\delta_j, j \in \{1, \cdots, M\}$ are equal to zero. In every iteration, using simulations, we determine the number of packets dropped for every IoT Device and then calculate $\hat{\delta}_j, j \in \{1, \cdots, M\}$ using (2).
- We then plug in the values in (3) and (18) and compare with $\Gamma$ and $\Gamma_r$ to decide $H_0$ or $H_1$.

A similar approach was carried out for obtaining the simulated missed detection probability values with the only difference being that the network is setup using $H_1$. It can be seen from Figure 2 that the GLRT scheme outperforms the Rao's test one and therefore GLRT based test is preferred over Rao's test even though the latter is computationally simple.

| Device | $D_1$ | $D_2$ | $D_3$ | $D_4$ | $D_5$ | $D_6$ | $D_7$ | $D_8$ |
|---|---|---|---|---|---|---|---|---|
| $\alpha_j$ | 0.06 | 0.24 | 0.13 | 0.97 | 0.09 | 0.07 | 0.02 | 0.12 |
| $\delta_j$ | 0.2 | 0 | 0 | 0 | 0.1 | 0 | 0 | 0.2 |

TABLE I: Parameters of the Devices

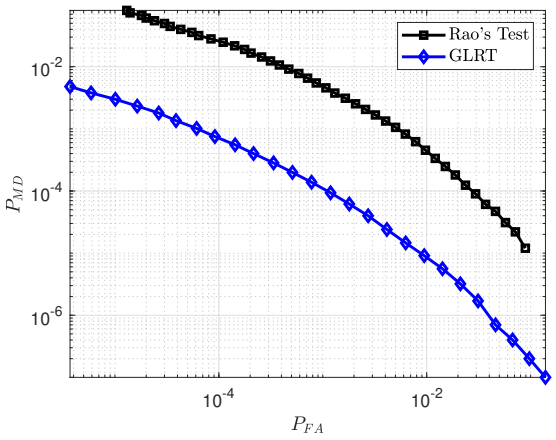


Fig. 2: Comparison with the scheme based on Rao's Test

## III. Experimental Setup

This section mainly describes the experimental setup used for demonstarting the performance of the IDS in (3). It includes the network setup (i.e. the clustered IoT network), the adversary implementation (i.e. how an adversary intrudes and disrupts the communication) and the function of the sentinel.

### A. Network Setup

We consider an IoT network with one IoT Device ($D$), one Relay ($R$) and one Access Point ($AP$). The IoT device exchanges information with a server, via $R$ and $AP$. The server in our testbed is connected directly to the Access Point, but will normally be situated in a remote location. The wireless links operate in IEEE 802.11bg mixed mode. The following is a brief description of the setup:

1) **IoT Device**: A DHT 11 Temperature and Humidity Sensor is used. This is connected to a RaspberryPi 3B+ which forwards the readings to the $AP$. Paho-MQTT Library is installed on the RaspberryPi to enable the above transmission. The device operates as a 'publisher' (an MQTT client) publishing topics 'Temperature' and 'Humidity'.
2) **Relay**: We used a TP-Link AC 1750 dual band router as the relay.
3) **Access Point**: A RaspberryPi 3B+ device is used. Paho-MQTT Library is installed here as well and the device operates as a 'subscriber' to subscribe to the topics published by the IoT device.
4) **Sentinel Node**: This is an inexpensive Wi-Fi adapter capable of monitor mode used to sniff packets in a network. For our experiment, we use a D-Link DWA-137 Wireless N High-Gain USB Wi-Fi adapter which is readily available as an off-the-shelf purchase. This experiment is performed on a host (Intel Core i7-7770 @ 3.60GHz with 16GB RAM of memory) running Kali Linux. The Wi-Fi adapter is connected to the host machine and set to monitor mode on the desired channel.

The network model is illustrated in Figure 3. The dotted lines indicate that the sentinel is sniffing all the packets on the link $D \rightarrow R$, and not only those addressed to it (if any). The IoT device uses MQTT (Message Queuing Telemetry Transport) protocol to communicate with the server. MQTT is a lightweight, publish-subscribe network protocol and runs over TCP/IP. When a packet is dropped due to network non-idealities, it does not reach the destination and hence there will be no acknowledgment sent from the destination. The IoT device will retransmit all unacknowledged packets. We can assume that the retransmission rate of the IoT device $D$ under normal conditions can be estimated and is constant at $\alpha$.

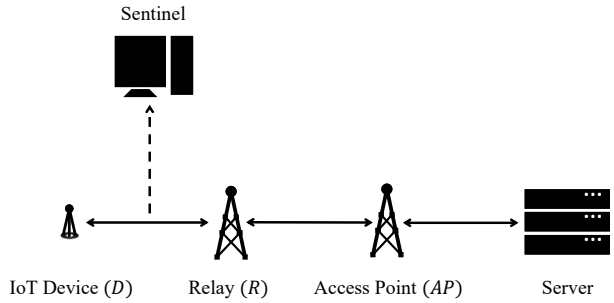


Fig. 3: Experimental Network Setup.

### B. Adversary Implementation

Man-in-the-middle attacks (MITMs) can be implemented using ARP (Address Resolution Protocol) Poisoning. ARP is used for mapping a MAC address to an IP address. In an ARP poisoning attack, an attacker sends false ARP reply messages over a local network (LAN) to associate the attacker's MAC address with the IP address of a legitimate machine on the network. Once the attacker's MAC address is mapped to the IP address of a legitimate machine, any message directed to the legitimate machine goes to the attacker's machine. **In our testbed, the adversary performs an ARP poisoning attack to direct all traffic from the IoT device to a machine within the attacker's control, as illustrated in Figure 4.** The adversary is now able to sniff, modify or drop the data packets. Kali Linux running on a VirtualBox was setup as the attacker's machine. Ettercap, an open source network security tool on Kali Linux, is used to carry out the Man-in-the-middle attack.

### C. Functions of the Sentinel

In [13], two methods are proposed to implement the IDS. In both the proposals, the IoT device calculates the retransmission rate and sends it periodically to a trusted machine (i.e. either the sentinel or the Access Point) which raises an alert when the attack is detected. We implemented this proposal making use of the 'Scapy' library of Python.

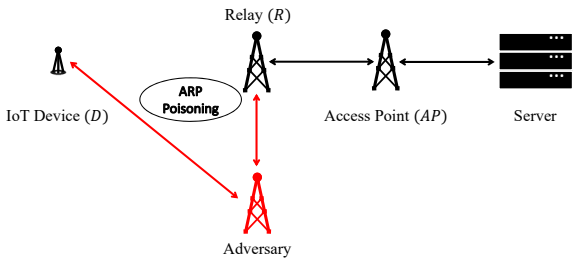


Fig. 4: Attack Scenario.

But this method requires the resource-constrained IoT device to calculate its own retransmission rate. In this paper, we propose instead to use a relatively inexpensive sentinel node for monitoring and calculating the number of packets retransmitted by the IoT device. The tasks executed by the sentinel are described briefly below:

1) **Estimating the number of restransmitted packets by the IoT Devices**: The IoT device is assumed to transmit in an 802.11 mode compatible with the sniffing device. The sentinel operating in monitor mode will now be able to capture the traffic of the IoT device using Wireshark. Using this method, we can count the number of retransmitted packets out of the past $K$ packets transmitted by the IoT device.
2) **Implementing the Intrusion Detection System**: Our network model has only one IoT device. Therefore, the Intrusion Detection System running on the sentinel decides that the relay is malicious iff

$$S = n\log(a) + K\log(1-\hat{\delta}) > \Gamma \tag{19}$$

where $a = \frac{\hat{\beta}}{\alpha(1-\hat{\delta})}$, $\hat{\beta} = \hat{\delta} + (1-\hat{\delta})\alpha$ and $K = 100$. The value of the estimated attack probability, $\hat{\delta}$, is

$$\hat{\delta} = \max\left(0, \frac{(n/K) - \alpha}{1-\alpha}\right) \tag{20}$$

In the absence of attack, according to [13], $S$ follows a Gamma distribution with its inverse scale parameter equal to 1.25 and shape parameter $g(\alpha) \triangleq 0.1777e^{0.4565\alpha} - 0.189e^{-786.1\alpha}$. Since the expressions for missed detection probability are not available, the value of the threshold is set using the false alarm probability expression. For a desired false alarm probability $\rho$, $\Gamma$ can be obtained using the cumultive distribution function of the Gamma distribution such that

$$P(S > \Gamma | H_0) = \rho \tag{21}$$

In general, there would be more than one IoT device in the network. Since this is an IEEE 802.11 based network and follows Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) protocol, at a given instant there can be at most one transmission in the network. Therefore, even if the number of IoT devices is greater than one, the sentinel will be able to capture every transmission and will also be able to estimate the number of retransmitted packets of all the IoT Devices present in the network. The transmissions of different IoT devices can be differentiated using their IP addresses. Note that, during the attack, the IP address is not affected. Also, there could be more than one relay in the network. Since the multiple access scheme is CSMA/CA, the neighboring relays transmit in orthogonal channels and therefore a sentinel node can monitor traffic from multiple relay clusters [17].

## IV. Experimental Results

In this section we present our experimental results. To calculate the value of $\alpha$, using the sentinel, we estimated the number of retransmissions out of 25000 packets transmitted by the IoT Device $D$ in four trials. For our setup, the value of $\alpha$ was estimated as 0.035. The following results are demonstrated in this section:

1) The estimated attack probabilities are compared against their actual values.
2) The performance characteristics of the IDS.
3) The expressions obtained for the false alarm probability is validated.

### A. Attack Probability Estimation

We considered four scenarios. In the first scenario, the value of $\delta$ is 0 i.e. there is no attack. In the second scenario, the overall drop probability i.e. $\beta$ is 0.1, in the third case, it is equal to 0.15 and in the fourth, it is equal to 0.2. The average values of the estimated attack probabilities, averaged over 250 trials, are tabulated in Table II. To calculate the estimated attack probability, the following steps were followed:

1) We setup the network either in $H_0$ or $H_1$ according to the scenario being implemented.
2) At the sentinel, for every 100 packets transmitted by $D$, the number of retransmissions is estimated.
3) We then plug in the above value in (20) and obtain the estimated attack probability.

It can be seen that the average of the estimated values are close to their actual values.

| | $\delta$ | $E[\hat{\delta}]$ |
|---|---|---|
| Scenario 1 | 0 | 0.0057 |
| Scenario 2 | 0.0674 | 0.0677 |
| Scenario 3 | 0.1192 | 0.1211 |
| Scenario 4 | 0.1710 | 0.1756 |

TABLE II: Attack Probabilities

### B. Performance Characteristics

To demonstrate the performance, we empirically obtained the false alarm ($P_{FA}$) and missed detection ($P_{MD}$) probabilities. To calculate the false alarm probability, the following steps were followed:

1) We setup the network using $H_0$ i.e. $\delta$ is equal to zero.
2) At the sentinel, for every 100 packets transmitted by $D$, the number of retransmissions is estimated.
3) The value of $\hat{\delta}$ is calculated using (20).
4) We then plug in this value in (21) and compare it with a pre-defined threshold to decide $H_0$ or $H_1$.

The missed detection probability is also estimated similarly except that the network is setup using $H_1$. We considered

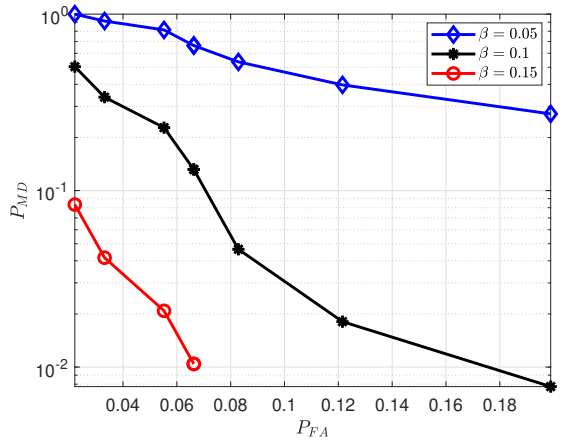


Fig. 5: Performance Characteristics.

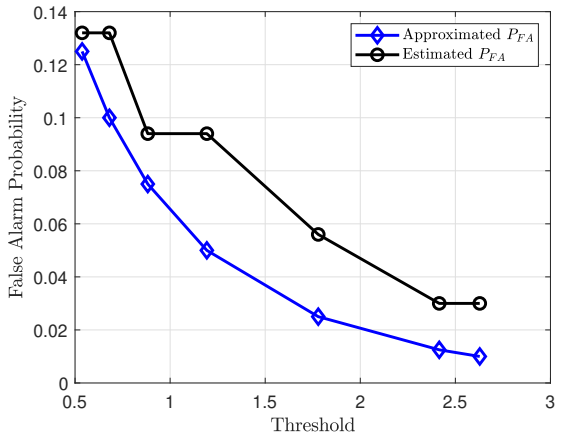


Fig. 6: Approximated and Estimated $P^u_{FA}$.

three scenarios where $\beta$ was equal to 0.05, 0.1 and 0.15, respectively. The performance obtained is illustrated in Figure 5. It can be seen that the IDS is able to detect the attack effectively even when only one device is present. The result also demonstrates that the missed detection probability decreases as the attack probability increases.

We now validate the expression obtained for $P_{FA}$ in (21). For the same, we calculate $P_{FA}$ using the Gamma approximation (termed as Approximated $P_{FA}$) and compare it with the empirically obtained $P_{FA}$ (termed as Estimated $P_{FA}$). The discrepancy observed between the approximated $P_{FA}$ and the estimated $P_{FA}$, as can be observed in Figure 6, is because we used only one IoT device in our setup.

## V. Conclusion and Future Work

In this paper, we experimentally demonstrated the effectiveness of the Intrusion Detection System proposed in [13]. The IDS requires the number of retransmitted packets by the IoT device to decide whether the relay is malicious or not. We were able to show that, using a sentinel, we can calculate the same. We proved, using Neyman-Fischer factorization theorem, that the statistics used to monitor the relays are sufficient. We also showed that bias of the estimator tends towards zero as the interval size increases. We also demonstarted that the detection algorithm based on GLRT outperforms the detection algorithm based on Rao's test. The results presented demonstrate that the selective forwarding attack, which results in draining the battery of the device and increasing the latency of the network, can be effectively detected.

As a part of out future work, we will demonstrate the performance of the IDS using more than one IoT device. We will also be looking at the possibilities where an IoT device tries to deceive the IDS by deliberately retransmitting successfully sent packets. By doing so, the sentinel can be tricked into deciding that the relay is malicious when it is not. **We also plan to explore attacks beyond ARP poisoning that can cause similar damage to the networks.**

## References


[1] S. Chen, H. Xu, D. Liu, B. Hu, and H. Wang, "A vision of IoT: Applications, challenges, and opportunities with china perspective," *IEEE Internet of Things journal*, vol. 1, no. 4, pp. 349–359, 2014.

[2] D. Evans, "The Internet of Things: How the Next Evolution of the Internet Is Changing Everything," 2011.

[3] M. M. Hossain, M. Fotouhi, and R. Hasan, "Towards an analysis of security issues, challenges, and open problems in the Internet of Things," in *Services, 2015 IEEE World Congress on*. IEEE, 2015.

[4] A.-R. Sadeghi, C. Wachsmann, and M. Waidner, "Security and privacy challenges in industrial Internet of Things," in *Design Automation Conference (DAC), 2015 52nd ACM/EDAC/IEEE*. IEEE, 2015.

[5] A. A. Abbasi and M. Younis, "A survey on clustering algorithms for wireless sensor networks," *Computer communications*, vol. 30, no. 14, pp. 2826–2841, 2007.

[6] K. T. Nguyen, M. Laurent, and N. Oualha, "Survey on secure communication protocols for the Internet of Things," *Ad Hoc Networks*, vol. 32, pp. 17–31, 2015.

[7] M. Zamani and M. Movahedi, "Machine learning techniques for Intrusion Detection," *arXiv preprint arXiv:1312.2177*, 2013.

[8] S. Raza, L. Wallgren, and T. Voigt, "SVELTE: Real-time intrusion detection in the Internet of Things," *Ad hoc networks*, vol. 11, no. 8, pp. 2661–2674, 2013.

[9] J. Ren, Y. Zhang, K. Zhang, and X. Shen, "Adaptive and channel-aware detection of selective forwarding attacks in wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 15, no. 5, pp. 3718–3731, 2016.

[10] S. Lim and L. Huie, "Hop-by-Hop cooperative detection of selective forwarding attacks in energy harvesting wireless sensor networks," in *Computing, Networking and Communications (ICNC), 2015 International Conference on*. IEEE, 2015, pp. 315–319.

[11] F. Gara, L. B. Saad, and R. B. Ayed, "An intrusion detection system for selective forwarding attack in IPv6-based mobile WSNs," in *Wireless Communications and Mobile Computing Conference (IWCMC), 2017 13th International*. IEEE, 2017, pp. 276–281.

[12] L. Wallgren, S. Raza, and T. Voigt, "Routing Attacks and Countermeasures in the RPL-based Internet of Things," *International Journal of Distributed Sensor Networks*, vol. 9, no. 8, p. 794326, 2013.

[13] N. V. Abhishek, A. Tandon, T. J. Lim, and B. Sikdar, "A GLRT Based Mechanism for Detecting Relay Misbehavior in Clustered IoT Networks," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 435–446, 2020.

[14] M. N. Aman, B. Sikdar, K. C. Chua, and A. Ali, "Low Power Data Integrity in IoT Systems," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 3102–3113, Aug 2018.

[15] S. M. Kay, "Fundamentals of statistical signal processing. Vol 1, Estimation theory," 1993.

[16] S. M. Kay, "Fundamentals of statistical signal processing: Detection theory, vol. 2," 1998.

[17] A. Tandon, T. J. Lim, and U. Tefek, "Sentinel based malicious relay detection in wireless IoT networks," *Journal of Communications and Networks*, vol. 21, no. 5, pp. 458–468, Oct 2019.