Alibi framework for identifying reactive jamming nodes in wireless LAN*

Hoang Nguyen
University of Illinois at Urbana-Champaign
hnguyen5@illinois.edu

Thadpong Pongthawornkamol
University of Illinois at Urbana-Champaign
tpongth2@illinois.edu

Klara Nahrstedt
University of Illinois at Urbana-Champaign
klara@illinois.edu

ABSTRACT

Reactive jamming nodes are the nodes of the network that get compromised and become the source of jamming attacks. They assume to know any shared secrets and protocols used in the networks. Thus, they can jam very effectively and are very stealthy. We propose a novel approach to identifying the reactive jamming nodes in wireless LAN (WLAN). We rely on the half-duplex nature of nodes: they cannot transmit and receive at the same time. Thus, if a compromised node jams a packet, it cannot guess the content of the jammed packet. More importantly, if an honest node receives a jammed packet, it can prove that it cannot be the one jamming the packet by showing the content of the packet. Such proofs of jammed packets are called "alibis" - the key concept of our approach.

In this paper, we present an alibi framework to deal with reactive jamming nodes in WLAN. We propose a concept of *alibi-safe* topologies on which our proposed identification algorithms are proved to correctly identify the attackers. We further propose a realistic protocol to implement the identification algorithm. The protocol includes a BBC-based timing channel for information exchange under the jamming situation and a similarity hashing technique to reduce the storage and network overhead. The framework is evaluated in a realistic TOSSIM simulation where the simulation characteristics and parameters are based on real traces on our small-scale MICAz test-bed. The results show that in reasonable dense networks, the alibi framework can accurately identify both non-colluding and colluding reactive jamming nodes. Therefore, the alibi approach is a very promising approach to deal with reactive jamming nodes.

*This material is based upon work supported by the National Science Foundation under Award Number DE-OE0000097. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of those agencies.

Categories and Subject Descriptors

C.2.0 [**Computer-Communication Networks**]: General—*Security and protection*; C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Wireless communication*

General Terms

Experimentation, Measurements, Security

Keywords

Wireless security, reactive jamming attacks, intrusion detection

1. INTRODUCTION

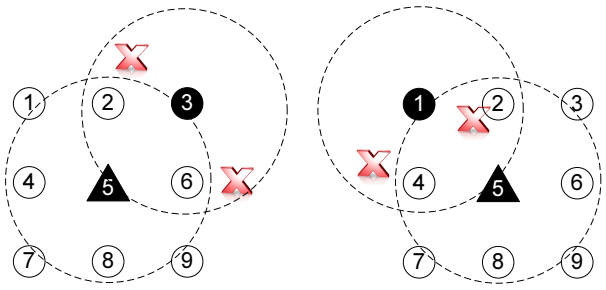
Wireless communications are inherently vulnerable to jamming attacks due to the open and shared nature of wireless medium. In the jamming attack, an attacker injects a high level of noise into the wireless system which significantly reduces the signal to noise and interference ratio (SINR) and reducing probability of successful message receptions.

There are two types of jamming strategies in wireless networks [31]: proactive and reactive jamming strategies. In proactive jamming strategies, attackers jam channels regardless of whether there are on-going communication activities on the channels. Typical examples of proactive jamming strategy are continuous jamming, periodic jamming and random jamming strategy. The active jamming strategies are easy to implement but they are not energy efficient. More importantly, they can be defended relatively easier than reactive jamming attacks (see Section 6).

Reactive jamming attacks, in contrast, only jam the channels when there are on-going communication activities. Typical examples are scan-and-jam (on multi-channel networks) and listen-and-jam strategy (on single-channel networks). The difficulty in defending against reactive jamming strategy lies on attackers' ability to become stealthy. For example, in a listen-and-jam strategy in a wireless LAN (WLAN), an attacker looks for an preamble signal and immediately sends a short packet to corrupt the body of the incoming packet. To become stealthy, it may want to jam some (possibly important) packets and behave like other nodes for the rest of the time. In this way, beside the fact that the attacker can do significant damage to the network, it also makes other honest nodes to back off more frequent and can enjoy extra throughput when behaving honestly. Thus, we refer to this type of attacks as *stealthy reactive jamming attacks.*

In this work, we consider the problem of identifying compromised nodes who launch stealthy reactive jamming attacks in half-duplex single-channel wireless LAN. This is a very challenging problem because the attackers are assumed to know any shared secret and protocols in the network and try to stay undetected as long as possible while maximizing the damage done to the network. To the best of our knowledge, none of the existing work can deal with this type of attackers in the context of WLAN (see Section 6) briefly due to two reasons. First, many approaches are only concerned about how to build jamming-resistant communications [27][26][22][9][11] without *identifying* the source of jamming. Jamming-resistant communications are necessary but not sufficient because as long as the jamming nodes are not identified, they always have effective jamming attacks on the network. Second, there are also several works on identifying mis-behaving nodes. However, because the attackers leave no identity information in the jammed packets (e.g. by corrupting the sender field), detection systems relying on identity clues to infer nodes causing the jammed packet do not work (e.g. [24][12])

We propose a novel framework to this challenging problem. The framework relies on "alibi" concept and thus is named as alibi framework. Alibi is "a form of defense whereby a defendant attempts to prove that he or she was elsewhere when the crime in question was committed". Because in a half-duplex wireless network where nodes cannot send and receive at the same time, attackers who jam will not be able to receive the content of jammed packets. Therefore, any nodes that can show proofs of corrupted packets are subject to getting an alibi in that time slot. In the long term, nodes who obtain least number of alibis are likely to get accused. Figure 1 illustrates how each honest node gets alibis on jamming events by node 5. In Figure 1(a), node 3's message is jammed. Node 2 and 6 receive a corrupted packet. If they both show the content corrupted packet, they can both claim an alibi. Similarly, in Figure 1(b), when node 1's message is jammed, node 2 and 4 will get an alibi by showing proofs of receiving an corrupted packet. In this manner, at some point each honest node in the network will get at least one alibi while the jammer (node 5) will have no alibis. Until then, the jammer can be identified.



(a) Node 3's message is jammed by node 5. Node 2, 6 get an alibi by showing the content of the corrupted packet.

(b) Node 1's message is jammed by node 5. Node 2, 4 gets an alibi by showing the content of the corrupted packet.

Figure 1: Example of Alibi Scheme.

Even though the concept of alibi framework may appear simple, there are numerous challenges to make it work. First, an attacker who jams in a time slot can always show a random corrupted packet content as a proof of reception. Therefore, the alibi system has to rely on the corrupted packet content from several receivers to justify the trustworthiness of the proofs of reception. Second, collided packets are not different from jammed packets. That means, nodes may get "false" alibis that are resulted from collisions. Thus, there will certainly be some noises in the detection process relying on number of alibis. Third, in an extreme situation where attackers decide to jam any single packet, the alibi system cannot operate on the main channel anymore. Therefore, there has to be a jamming-resistant communication to exchange alibi information in the network. The alibi framework uses a BBC-based timing channel with no pre-shared secret for nodes to communicate. Due to the low throughput of the BBC-based timing channel where raw packet content cannot be exchanged efficiently. The alibi framework uses a technique called "similarity hashing" to compress the packet content such that their similarity are still preserved. Fourth, there might be more than one attacker in the system. A jamming action of one attacker may help other attackers to obtain alibis. The situation is even worse when they can coordinate/collude with each other. Lastly, attackers are assumed to know any protocols used in the network. Thus, they can do certain types of attacks on the alibi system such as slander attacks.

We have designed and implemented the alibi framework using both analysis, simulation and MICAz test-bed experiments. Our major results and findings can be summarized as follows.

- We develop a novel alibi framework for the problem of identifying reactive jamming attacks.
- We carry out experiments on a small MICAz test-bed of 6 nodes to study the impact of reactive jamming attacks on the network performance and the hypothesis of reception similarity under reactive jamming attacks (Section 3).
- Based on the results from the experiments, we propose a concept of "alibi-safe" network topology where the alibi framework can identify attackers and corresponding alibi identification algorithms (Sections 2 and 4). We also prove the correctness of the alibi identification algorithms on alibi-safe topologies. To this end, we study the connection between physical network topologies and alibi-safe topologies (Section 5.1). This connection is very important in helping network designer to deploy the wireless networks to defend against the reactive jamming attacks.
- We design a practical alibi protocol including a BBC-based timing channel (Section 4.4) and a similarity hashing technique (Section 4.5) for more efficient proof exchange under the reactive jamming attacks.
- We implement and evaluate the alibi framework on a large-scale networks in TOSSIM simulator (Section 5). The implementation includes BBC-based timing channel, similarity hashing and the identification algorithms. We incorporate the small-scale MICAz experimental results including the packet error rate trace, the noise trace and the reception similarity trace into TOSSIM for accurate simulation results.

The rest of the paper is organized as follows. First, in Section 2, we give the network model and attack model. In Section 3, we study the impact of reactive jamming attacks on the network performance and the reception similarity. Then, we show the details of the alibi framework in Section 4 including the algorithms, the framework and basic building blocks such as BBC-based timing channel and similarity hashing. We show the evaluation results of the alibi framework in Section 5. We discuss the related work in Section 6. Finally, we conclude the paper in Section 7.

2. SYSTEM MODEL

2.1 Network model

We consider a single-channel WLAN of n nodes. One node is the trusted base station. Denote $\mathcal{N}$ as the set of the nodes in the network (i.e, $|\mathcal{N}| = n$). Each node in the network is equipped with a half-duplex radio, i.e. it cannot transmit and receive at the same time. Thus, there will be non-negligible delay to switch from transmit mode to receive mode and vice versa. We assume CRC-failed packets are still delivered to the upper layer. We assume each node uses CSMA/CA MAC. We also assume a central detection model, i.e. nodes will send information to the base station.

2.2 Attack model

We assume some nodes in the network are the reactive jamming attackers. Thus, the attackers also have same physical capabilities as other nodes. The attackers, however, have the complete control of the MAC, physical parameters of the radio network interface. The attackers are insider attackers. That means, they are assumed to know any security-related information of the node such as security keys. They also know any protocols used in the system.

The goal of the attackers is to remain undetected while maximizing the number of jammed packets. The attackers use probabilistic reactive jamming strategy. That means whenever an attacker J senses an on-going packet by detecting the presence of a preamble, it will transmit a jamming packet with probability p_J. p_J is called the "reactive jamming probability" and is defined for *each sending packet*. This definition is different from traditional jamming rate in the literature which is defined over *each time slot* regardless of whether there is sending packet in that time slot. Henceforth, the term "jamming rate" refers to "reactive jamming rate" unless explicitly specified.

3. IMPACT OF REACTIVE JAMMING ATTACKS

It might be noticed that in the example above *we assume received corrupted packets, caused by the same jamming event, have the same content* (e.g., nodes 2 and 6 in Figure 1(a)). In practice, this might not be the case. Thus, in this section, we will carry out several experiments on a testbed of MICAz motes with CC2420 radio to answer following two questions: 1) *what are the capabilities of reactive jamming attacks?* and 2) *what is the similarity of corrupted packet contents under reactive jamming attacks?*.

3.1 Impact of reactive jamming attacks on network performance

In our experiments, a reactive jamming attack is performed on a set of 3 nodes as shown in Figure 2(a). Nodes are placed such that they can hear each other at the strongest power level (i.e. level 31, 0dbm). S and J are the sender and jammer, respectively. R is the receiver who receives packets from S, J. C acts as the experiment controller.

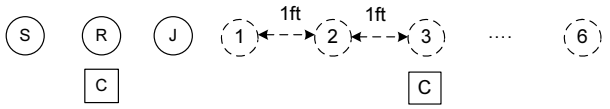
To produce a reactive jamming attack, C will broadcast a message. Upon receiving the broadcast message from C, S starts sending a message of 43 bytes including 32-byte random payload and 11-byte MICAz header. J also starts sending a message of 43 bytes, including 32-byte all-0 payload and 11-byte MICAz header, but with a delay $\delta > 0$. δ is chosen such that the jamming packet will arrive at the receiver after the preamble of the sender. This is just to make sure we have a correct implementation of a reactive jammer. In our experiments, δ is between $150\mu s$ and $200\mu s$. Note that we disable the clear channel assessment (CCA) and backoff mechanism of S and J to ensure concurrent transmissions and CRC check mechanism of R. R records any messages right after the broadcast message from C. The recorded messages are then time-stamped and sent back to C for trace collection. For confident statistics, each experiment is repeated 200 times.

It is known that signal-to-interference-noise-ratio (SINR) will decide the packet content. Theoretically, $SINR = \frac{P_S^R}{P_J^R + P_N}$ where P_S^R and P_J^R are the received powers of the signal sent from S and J at the receiver R; P_N is the noise power. P_S and P_J can be calculated as $P_S^R = P_S \times {d_{SR}}^{-\alpha}$ and $P_S^R = P_S \times {d_{JR}}^{-\alpha}$ where P_S, P_J are the sending powers of S, R; d_{SR}, d_{JR} are the distance from S, J to R; and α is the path loss factor. Therefore, the major factors affecting SINR are the distance between $S \to R, J \to R$ and the sending powers of S, J. Thus, in our experiments, we vary the distance between $S \to R, J \to R$ and the sending powers of S, J. We put 6 MICAz motes in the line as shown in Figure 2(b). In a reactive jamming attack scenario, we have one sender at mote i ($i = 1..6$) with the sending power level k ($k = 1..30$), one jammer at mote j ($j \neq i, j = 1..6$) with the sending power level l ($l = 1..30$) and 4 receivers at remaining nodes. We try all possible combinations of (i, j) in 6 positions where $j > i$. For each (i, j) pair, we will perform following experiments.

- We measure the received signal strength indication (RSSI) from the sender to each receiver R, denoted as $RSSI_{SR}(d_{SR}, P_S)$, without any sending of the jammer.
- Similarly, we also measure the RSSI from the jammer to each receiver R, denoted as $RSSI_{JR}(d_{JR}, P_J)$, without any sending of the sender.
- We collect corrupted packets at each receiver under reactive jamming attacks (i.e., both sender and jammer sending) for reception similarity calculation.

Figure 3(a) shows the RSSI at a receiver 1ft away from the sender and the receiver at different sending power levels, i.e. $RSSI_{SR}(1ft, P_S)$ and $RSSI_{JR}(1ft, P_J)$. The x-axis is sending power level of the sender and the jammer (P_S, P_J). The y-axis is the RSSI in dbm.

Figure 3(b) shows the packet error rate under reactive jamming attacks. The x-axis, denoted by "RSSI by sender" (i.e. $RSSI_{SR}$), is the RSSI of the signal from the sender to



(a) Reactive jamming attack configuration

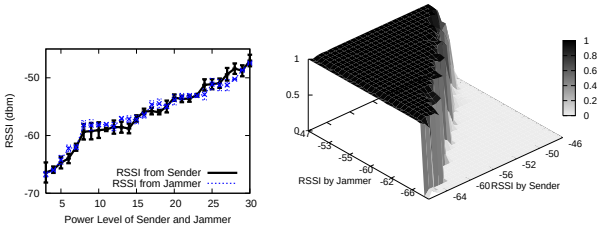
(b) Topology of 6 nodes

Figure 2: Reactive jamming experiment settings

the receiver. Note that the RSSI metric takes into account of both the sending power and the distance between the sender and the receiver. Similarly, the y-axis, denoted by "RSSI by jammer" (i.e. $RSSI_{JR}$), is the RSSI of the signal from the jammer to the receiver. A pair of RSSI from the sender and the jammer characterizes a receiver. The z-axis shows the packet error rate of each receiver.

From the graph, we have the following observations.

- If $RSSI_{SR}(d_{SR}, P_S) >> RSSI_{JR}(d_{JR}, P_J)$, the packet error rate decreases sharply to 0. This region is referred to as the "white" region. Receivers in this region are called white receivers.
- If $RSSI_{SR}(d_{SR}, P_S) << RSSI_{JR}(d_{JR}, P_J)$, the packet error rate increases sharply to 1. This region is referred to as the "black" region. Receivers in this region are called black receivers.
- When $RSSI_{SR}(d_{SR}, P_S)$ and $RSSI_{JR}(d_{JR}, P_J)$ are close ($< 5dbm$ difference), the packet error rate is between 0 and 1. This region is referred to as the "grey" region. Receivers in this region are called grey receivers.



(a) Power level vs. RSSI

(b) Packet Error Rates (z-axis)

Figure 3: Results on reactive jamming experiments

3.2 Reception similarity under reactive jamming attacks

We want to see the similarity of received packet content of any pair of receivers under reactive jamming attacks. We treat a packet content as a binary string. The similarity of two packet contents is defined as the similarity of the two corresponding binary strings. The similarity of two binary strings B_1, B_2 of length l is defined as

$$sim(B_1, B_2) = 1 - \frac{\mathcal{H}(B_1, B_2)}{l} \quad (1)$$

where $\mathcal{H}(B_1, B_2)$ is the Hamming distance of B_1 and B_2 defined as the number of positions at which the corresponding bits of B_1 and B_2 are different. The similarity has a range of $[0, 1]$. Similarity of 0 means two packet contents are completely different. Similarity of 1 means two packet contents are identical.

We correlate all received packet content of each pair of receivers. Thus, we have a reception similarity table whose lines are tuples of $(RSSI_{SR_1}, RSSI_{JR_1}, RSSI_{SR_2}, RSSI_{JR_2}, C)$. The pairs $(RSSI_{SR_1}, RSSI_{JR_1})$ and $(RSSI_{SR_2}, RSSI_{JR_2})$ characterize the first and the second receiver, respectively. C is the average of similarity of received packet content of R_1 and R_2. Note that in our experiments, the lowest similarity is 0.5, statistically. This is because the content of the sending packet is uniformly generated, i.e. each bit is uniformly generated between 0 and 1. After constructing the reception similarity table, we have the following observations.

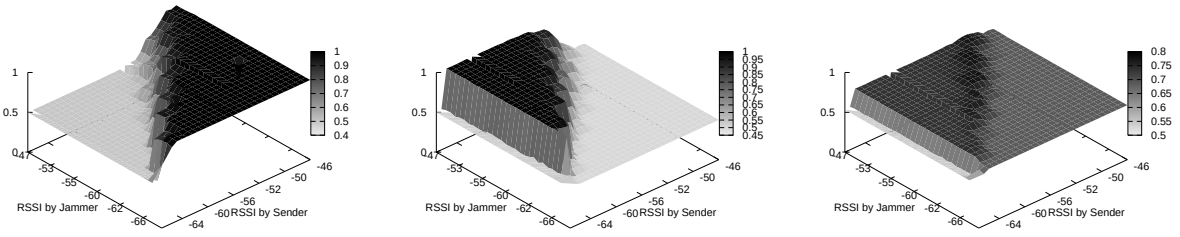
- For a white receiver (i.e. successful reception), it will have very strong reception similarity (close to 1) with other white receivers. This is obvious because nodes in the white region have successful packet reception. It has a wide range of weak reception similarity with grey receivers (range of $[0.55, 0.95]$). This can be explained as the grey receivers still have some correct bits from the sender's content. However, a white receiver has very weak reception similarity with black receivers (range of less than 0.55). Figure 4(a) shows the reception similarity of a typical white receiver R ($RSSI_{SR} = -50dbm$, $RSSI_{JR} = -64dbm$) with all other receivers.
- Similarly, for a black receiver (i.e. unsuccessful reception), it has a very strong reception similarity with other black receivers, a wide range of weak reception similarity with grey receivers and very weak reception similarity with white receivers. Figure 4(b) shows the reception similarity of a typical black receiver R ($RSSI_{SR} = -59dbm$, $RSSI_{JR} = -54dbm$) with all other receivers.
- For a grey receiver, it has a strong reception similarity (range of $[0.6, 0.8]$) with other grey receivers in the grey region. It has wide range of a week reception similarity with black and white receivers. Figure 4(c) shows the reception similarity of a grey receiver R ($RSSI_{SR} = -59dbm$, $RSSI_{JR} = -61dbm$) with all other receivers.

From this experimental study, we can conclude that **the reception similarity of received packet content under reactive jamming attacks has the locality property**. That means, receivers closer in the RSSI plane have stronger reception similaritys of packet contents. This is very important for the design of alibi framework as shown in Section 4.

4. ALIBI FRAMEWORK

4.1 Definitions and notations

Denote $\mathcal{S}(t)$ as the set of transmitters at time slot t. Denote $P_{\mathcal{S}\to r}(t)$ as the packet content received by the receiver r under the concurrent sending of senders in $\mathcal{S}(t)$. Denote $PR_{\mathcal{S}\to r}(t)$ ($\mathcal{S} \subset \mathcal{N}\backslash r$) the *proof of reception* for a receiver r at time slot t.



(a) Reception similarity of a white receiver R ($RSSI_{SR} = -52dbm$, $RSSI_{JR} = -64dbm$)

(b) Reception similarity of a black receiver R ($RSSI_{SR} = -59dbm$, $RSSI_{JR} = -54dbm$)

(c) Reception similarity of a grey receiver R_1 ($RSSI_{SR_1} = -59dbm$,$RSSI_{JR_1} = -61dbm$)

Figure 4: Experiment results on the reception similarity (z-axis: darker colors represents higher values)

DEFINITION 1 (ALIBI OF RECEPTION (AR)). *An alibi of reception for two receivers r and q at time slot t under the set of sender $\mathcal{S}$ is defined as*

$$AR_{\mathcal{S}\to r,q}(t) = sim(PR_{\mathcal{S}\to r}(t), PR_{\mathcal{S}\to q}(t))$$

where sim is the reception similarity function defined in Equation 1.

DEFINITION 2 (α-ALIBI NEIGHBORS). *Two nodes r and q are called α-alibi neighbors ($0 \le \alpha \le 1$) under a set of senders $\mathcal{S}$ in the time slot set $\mathcal{T} = t_1, t_2, .., t_{|\mathcal{T}|}$ if*

$$\mathbb{E}[AR_{\mathcal{S}\to r,q}(\mathcal{T})] = \frac{1}{|\mathcal{T}|}\sum_{\forall t\in\mathcal{T}} AR_{\mathcal{S}\to r,q}(t) \ge \alpha$$

The definition above involves the *average* of reception similarity of two receivers over a set of time slots. This is because we do not know the exact distribution of the reception similarity of any two receivers[1]. Thus, our analysis will rely only on the average and deviation of the reception similarity derived from the experiments described in Section 3.

DEFINITION 3 (β-JAMMER). *A jammer is called β-alibi-jammer ($\beta \ge 0.5$) if it can guess correctly the outcome of the packet from a sender $s \in \mathcal{N}$ caused by its jamming actions with probability β.*

Based on our experiments, which is also confirmed in [25], it is very hard to guess the content of the jammed packet. Thus, the best the jammer can do is a random guess which results in $\beta = 0.5$.

DEFINITION 4 (COMPLETE ALIBI-SAFE TOPOLOGY). *A wireless network topology is called a (α, κ)-alibi-safe topology if for all pairs of sender $s \in \mathcal{N}$ and jammer $j \in \mathcal{N}$, every node $r \in \mathcal{N}\backslash\{s, j\}$ has at least κ α-alibi neighbors.*

The definition above is strict because it requires that for any possible locations of the jammer and the sender, any receiver always has at least κ α-neighbors. This guarantees that any honest node is always alibi-safe regardless of where the attackers may be. However, if we know some nodes in the network that can be trusted, we can have a looser definition. For example, in our WLAN setting, the base station can be trusted. Thus, we only need to make sure that every node has at least κ α-neighbors under the sending messages from the base station jammed by any possible jammer.

DEFINITION 5 (ALIBI-SAFE TOPOLOGY WITH TRUSTED SENDERS). *A wireless network topology is called a $(\alpha, \kappa, \Lambda)$-alibi-safe topology under a set of trusted senders $\Lambda \subset \mathcal{N}$ if for all pairs of sender $s \in \Lambda$ and jammer $j \in \mathcal{N}\backslash\Lambda$, every receiver $r \in \mathcal{N}\backslash\{j, \Lambda\}$ has at least κ α-alibi neighbors.*

4.2 Alibi identification algorithms

4.2.1 Identifying non-colluding attackers

Algorithm: The basic identification algorithm relies on the fact that a jammer can only do a correct guess with probability β on the content of the packet it jams. That means, whenever it jams a packet, it only has β reception similarity while the other nodes get at least α reception similarity (statistically). Let us define an alibi score function *ascore* for a node r in a time slot set $\mathcal{T}$ as follows

$$ascore_r^\alpha(\mathcal{T}) = \sum_{t\in\mathcal{T}} \delta_r^\alpha(t)$$

where $\delta_r^\alpha(t) = \begin{cases} 1 & \text{if } \max_{q\in\mathcal{N}\backslash r} AR_{S\to r,q}(t) \ge \alpha \\ 0 & \text{otherwise} \end{cases}$

$\delta_r^\alpha(t)$ is the alibi indicator function for r at time t. It indicates whether at time slot t, a node r has an reception similarity greater than α with some node. Thus, the necessary condition to identify a jammer j is

$$ascore_r^\alpha(\mathcal{T}) > ascore_j^\alpha(\mathcal{T}), \forall r \in \mathcal{N}\backslash j \qquad (2)$$

. In other words, a node is accused if it has lowest alibi score.

CLAIM 1 (IDENTIFYING NON-COLLUDING ATTACKERS). *In (α, κ)-alibi-safe topology ($\kappa \ge 1$) with k_a β-jammers $\mathcal{J} = j_1, ..j_{k_a}$ with jamming rate $p_{j_1}, ...p_{j_{k_a}}$, the alibi scheme can identify any attacker $j\mathcal{J}$ if*

$$p_s^{max} \le 2 - \frac{1 - p_j}{p^{agg}(\mathcal{J})}$$

where $p^{agg}(\mathcal{J}) = 1 - \prod_{i=1}^{k_a}(1 - p_{j_i})$ is the aggregated jamming rate of the jammer set $\mathcal{J}$ and p_s^{max} is the maximum of sending probabilities of the honest nodes.

[1]In fact, to the best of our knowledge, there is no theoretical model capturing the distribution of the reception similarity under concurrent transmissions.

Proof: See Appendix.

It is easy to see that the above identification algorithm also works for alibi-safe topologies with trusted senders. Instead of considering packet sending events from any senders, we only consider the packet sending events from the trusted senders.

4.2.2 Identifying colluding attackers

DEFINITION 6 (COLLUDING ATTACKERS). *Colluding attackers are those who have pre-shared knowledge among themselves.*

Note that the definition above allows the attackers to collude through a pre-shared knowledge only. It does not consider the case where attackers can share new knowledge during the network operations (e.g. their proofs of reception). However, collusion via pre-shared knowledge is a very strong attacker model as follows. First, with pre-shared knowledge, colluding attackers can agree on a content of "fake" proofs sent to the detector. In this way, an attacker who jams is still able to get reception similarity with the other attackers. This collusion will help the jammers to escape from the above alibi identification algorithm. In a bigger picture, with pre-shared knowledge, the attackers can arbitrarily manipulate the reception similarity among themselves. However, they cannot manipulate the reception similarity with other honest nodes. Second, with pre-shared knowledge, colluding attackers can coordinate who jams which slot. That means, any two attackers will never jam at the same time slot. In this way, the jammers never waste their jamming effort. Third and last, it is possible to perform a coordinated jamming attack to create "real" proofs as follows. At a time slot, first jammer sends a packet. Second jammer jams the packet of the first sender. Thus, the rest of jammers and honest nodes can all get real proofs and real reception similarity among each other. Even though the first two jammers might not get any reception similarity[2], they can get compensated later when other jammers take turn. Apparently, this type of coordinated jamming attack can break any alibi identification algorithms in complete alibi-safe topologies where no senders can be trusted. Thus, we are now going to present an identification algorithm to *identify colluding attackers in alibi-safe topologies with trusted senders*. Note that in WLAN, the set of trusted nodes could be the base station alone.

Algorithm: The basic strategy to deal with colluding attackers is to exploit their only weakness: they cannot manipulate the reception similarity with other honest nodes under the jammed packets from the trusted senders $S \in \Lambda$. Thus, we define the alibi indicator function for a node r as follows.

$$\tilde{\delta}_r^{\alpha}(t) = \begin{cases} 1 & \text{if } q_\kappa \in \Upsilon_r(\mathcal{N}), AR_{S\to r,q_\kappa}(t) \geq \alpha \\ 0 & \text{otherwise} \end{cases}$$

where $\Upsilon_r(\mathcal{N})$ is the list of nodes in $\mathcal{N}$ decreasingly sorted by the similarity with r at time t and Q_k is the k-th element in $\Upsilon_r(\mathcal{N})$. Once again, the alibi score is defined as

$$\tilde{asc\hat{o}re}_r^{\alpha}(\mathcal{T}) = \sum_{t\in\mathcal{T}} \tilde{\delta}_r^{\alpha}(t)$$

[2]In fact, they may be able to get reception similarity because both the content of sending packet and jamming packet are known.

. Similar to the case of non-colluding attackers, an attacker j is identified if

$$\tilde{asc\hat{o}re}_r^{\alpha}(\mathcal{T}) > \tilde{asc\hat{o}re}_j^{\alpha}(\mathcal{T}), \forall r \in \mathcal{N}\backslash j$$

CLAIM 2 (IDENTIFYING COLLUDING ATTACKERS). *In a (α,κ,Λ)-alibi network topology ($\kappa > 1$) with k_a β-alibi colluding jammers $\mathcal{J} = j_1,..,j_{k_a}$ with jamming rate $p_{j_1},..,p_{j_{k_a}}$ and the total jamming target $p_T = \sum_{i\in\mathcal{J}} p_i$, the alibi scheme can identify at least one attacker if $p_s^{max} < \frac{p_T}{k_a}$.*

Proof: See Appendix.

4.3 Alibi protocol

The identification algorithm in Section 4.2 requires the proofs to be present at the detector. That means, every node in the network has to participate in the detection algorithm. Nodes that do not report proofs are at risk of being accused as attackers and received appropriate reaction from the system such as their removals.

Jamming detection: Because sending proofs incurs overhead to nodes, the base station only collects proofs when there is a jamming attack. To detect the presence of jamming attacks, we use a similar detection techniques proposed in [31]. For the uplink traffic (i.e. from nodes to the base station), a jamming attack is declared if the base station receives a significant number of corrupted packets with strong received signal strength. For the unicast downlink traffic, the base station declares the presence of jamming attacks after getting a significant number of sending packets without receiving acknowledgements.

- *Step 1:* Once the base station detects a jamming attack, it starts a *proof-collection* period of $T_{collect}$ in which it starts to broadcast "test" packets at random intervals. The content of the test packets are uniformly drawn from the message space, i.e. each bit in the content is uniformly drawn from $\{0,1\}$. The uniform randomness ensures that the attacker can only have a blind guess on the content. Note that each honest node always maintains proofs in the latest window interval of $T_{collect}$. At the end of the proof-collection period, the base station broadcasts a *proof-exchange* message.

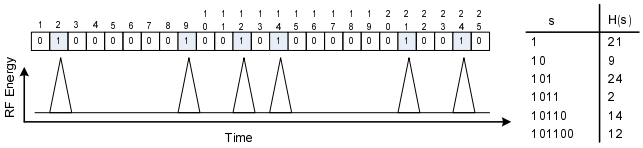
- *Step 2:* When nodes receive the *proof-exchange* message from the base station, they immediately start sending the proofs that they have just collected in the last $T_{collect}$ time interval to the base station.

- *Step 3:* After receiving proofs from nodes collected in the last $T_{collect}$ time interval, the base station starts the identification algorithm based on the set of time slots in which it sent "test" packets. It removes any identified attackers. After that if there is still jamming attack going on, it repeats Step 1.

All messages between nodes and the base station are transmitted using BBC-based timing channel. The BBC-based timing communication relies on the timing pattern of sending packets to convey the actual message. It has a strong resistance to reactive jamming attacks and allows concurrent transmissions of multiple senders. One major disadvantage of this timing channel is its low throughput. More details are discussed in Section 4.4.

To cope with the low throughput nature of the BBC-based jamming-resistant timing-channel on which proofs are exchanged, the alibi framework compresses the proofs using a hashing technique called "similarity preserving hashing" (or locality sensitive hashing). Unlike other hashing techniques such as MD5 or SHA-1, similarity hash functions have a special property that the Hamming distance of hash values of two objects is proportional to their original Hamming distance. Therefore, instead of storing and exchanging the raw packet content, each node only needs to keep the hash version of proofs to reduce the storage and communication overhead. Section 4.5 will give more details.

4.4 BBC-based timing channel



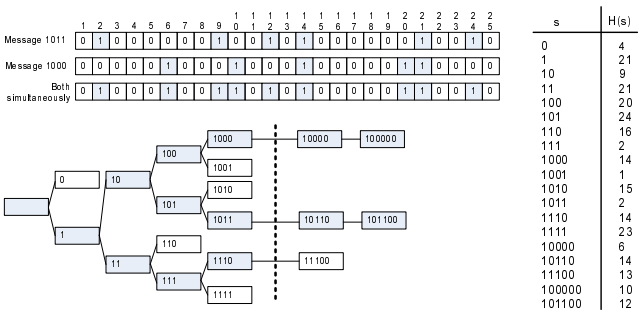
s	H(s)
1	21
10	9
101	24
1011	2
10110	14
101100	12

Figure 5: An example of BBC broadcast using pulses (excerpted from [9]).

BBC is a keyless jamming-resistant broadcast communication proposed in [9]. The basic idea is to have the sender create *"indelible" marks* in an additive-OR channel to convey a sending message. The receiver receives a "packet" containing all the marks and decodes the original sending message. Indelible marks in the additive-OR channel have an important property: the jammers cannot erase the existence of the marks in the channel; they can only create extra marks. Therefore, a received BBC packet may contain more marks than those created by the senders. In other words, the set of marks in the received BBC packet is the super-set of the marks created by senders. Thus, the coding scheme is called *concurrent code* and BBC code is the only known implementation of the concurrent code.

A message M of length n bits is encoded into a message M' of length $m = ne$, where e is the expansion factor ($e > 1$). Denote $\Pi(m, i)$ to be the first i bits of message m and a hash function H (e.g. MD5 or SHA1). For each $i \in (1, n)$, we calculate the location $L(i)$ of the i-th mark corresponding to bit i as $L(i) = H(\Pi(m, i)) \mod m$. Therefore, the message M is encoded into n marks whose locations are from $0..m$. The encoded message M' will then be transmitted in m slots. The marked slots correspond to transmissions of a pulse (i.e. a preamble-only packet). The unmarked slot is equivalent to no transmission. Figure 5 shows an example of a BBC encode and broadcast using pulses. Message $M = 1011$ is padded with two 0-bits and becomes $M = 101100$. An *indelible mark* is a radio pulse, and its location is the time when pulse occurs relative to the start of the message. The table on the right shows part of the definition of the hash function $H(x)$. Each prefix of the padded message is hashed using the hash function $H(x)$. The results of the hash are used as the locations of the pulse.

There are several ways to create an additive-OR channel [9]. In the alibi framework, we build a pulse-based channel from the data channel. A pulse is 4-byte preamble-only packet. Multiple pulses sent at the same time results in only one single pulse. In this way, this pulse-based channel is an additive-OR channel. For BBC code, a pulse is also



s	H(s)
0	4
1	21
10	9
11	21
100	20
101	24
110	16
111	2
1000	14
1001	1
1010	15
1011	2
1110	14
1111	23
10000	6
10110	14
11100	13
100000	10
101100	12

Figure 6: Decode tree for the BBC broadcast using pulses (excerpted from [9]).

an indelible mark. Reactive jammers cannot erase marks because they jam only after sensing the preambles[3]. In addition, concurrent sending of multiple marks will result in only one mark. Therefore, the BBC-based timing channel built in this way is not only robust to the reactive jammers but also allows multiple concurrent message transmissions.

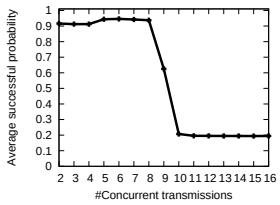
It is important to emphasize that there is a critical threshold on the number of concurrent transmissions below which all concurrent transmission are highly successful and above which no concurrent transmission are successful. This critical threshold implies a trade-off between the maximum number of successful concurrent transmissions and the transmission length of the messages. Specifically, to have n nodes sending a message of b bits concurrently with high successful rate, the transmission length has to be $O(n \times b)$. Figure 7(a) shows the performance of the BBC-based timing channel in which the maximum number of concurrent transmissions is targeted at 10. As shown in the Figure 7(a), there is a sharp decrease of the successful transmission probability (y-axis) from 0.95 to 0.2 when the number of concurrent transmission (x-axis) approaches 10 and goes beyond. Figure 7(b) further shows the transmission length (y-axis) versus the targeted maximum number of concurrent transmissions (x-axis).

It is also important to emphasize that the proof transmission using BBC in Step 2 of the alibi protocol is not encrypted. If in Step 2, a jammer decides to listen for some proofs, which takes a significant time, and start the proof transmission to the BS, the start of its proof transmission period will be significantly late compared to any other nodes and can easily be detected by the BS.

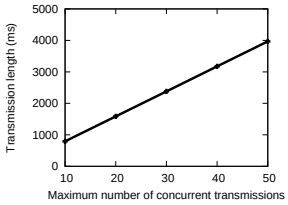
4.5 Similarity Hashing (SimHash) & Alibi

Locality Sensitive Hashing (LSH) is a popular technique used in information retrieval to detect near-duplicate documents [6]. Essentially, LSH is a method of performing probabilistic dimension reduction of high-dimensional data. The basic idea is to hash high-dimension input objects so that similar objects are mapped into the same buckets with high probability. In other words, input objects are hashed such that their similarities are preserved in the hash space. This similarity-preserving property is completely opposite to normal hashing techniques (e.g., SHA1, MD5), where a small difference of inputs might lead to a completely different hash

[3]In fact, even if the attackers jam the preambles, it is still possible to detect the jammed preambles. However, the preamble detection has to rely on the energy detection rather than the preamble signature.



(a) The critical threshold of BBC for 10 nodes



(b) Maximum no. concurrent transmissions vs. Transmission length

Figure 7: Performance of the BBC-based timing channel

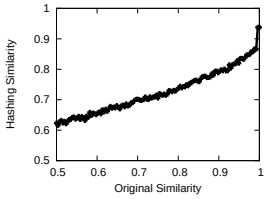


Figure 8: Accuracy of Simhash

outputs. Formally, a locality sensitive hashing scheme is a distribution on a family $\mathcal{F}$ operating on a collection of objects, such that for two objects x, y, $\mathbf{Pr}_{h\in\mathcal{F}}[h(x) = h(y)] = similarity(x, y)$, where $similarity(x, y) \in [0, 1]$ is the similarity function defined on the collection of objects. In the alibi framework, a similarity hashing function $h \in \mathcal{F}$ is used to hash the packet content. That means, the reception similarity of two packet content B_1, B_2 will be calculated as $sim(B_1, B_2) = 1 - \frac{h(B_1),h(B_2)}{l}$.

While there are several techniques to implement $\mathcal{F}$ [6][10][3][8], we choose the random projection technique, also referred to as *simhash*, proposed by Charikar [6] due to its simple and efficient implementation [18]. The random projection method of the similarity preserving hashing is designed to approximate the cosine distance between vectors. The basic idea of this technique is to choose a random hyperplane (defined by a normal unit vector $\vec{r}$) at the outset and use the hyperplane to hash input vectors. Formally, we let $h(\vec{v}) = sgn(\vec{v} \cdot \vec{r})$, i.e., $h(\vec{v}) = \pm 1$ depending on which side of the hyperplane $\vec{v}$ lies. Therefore, each possible choice of r results in a binary output. If we choose the vector $\vec{r}$ t times uniformly, we will have a t-bit output of vector $\vec{v}$. For any two vectors $\vec{u}, \vec{v}$, the bits of two t-bit outputs match with probability proportional to the cosine of the angle between them. In [6], the author proves that $\mathbf{Pr}[h_r(\vec{u}) = h_r(\vec{v})] = 1 - \frac{\theta(\vec{u},\vec{v})}{\pi}$, where θ is the angle of vectors $\vec{u}$ and $\vec{v}$.

The point of using *simhash* is to reduce the amount of data needed for storing and transmitting proofs. For example, a 32-byte packet content can be sim-hashed into a 2-byte fingerprint while still preserving reasonable accuracy of the similarity with other packet content. Figure 8 shows the accuracy of 2-byte sim-hash for a 32-byte vector. Two 32-byte vectors sampled uniformly are sim-hashed. The plot shows the similarity of original 32-byte vectors (i.e., X-axis) and the similarity of the 2-byte sim-hash values (i.e., Y-axis).

4.6 Outlier detection algorithms

Due to the probabilistic nature of the alibi score defined in Section 4.2, we need a statistical detection algorithm to detect outliers in the set of alibi scores calculated for nodes. Specifically, given a set of $ascore_R^\alpha(\mathcal{T}), \forall R \in \mathcal{N}$ in the time slot set $\mathcal{T}$, we need to identify the set of nodes with the alibi scores that are too low compared to other nodes.

Because we do not know the distribution of the alibi scores, the outlier detection algorithm has to be non-parametric. Therefore, there will certainly be the tradeoff between the false alarm probability and correct detection probability. In the alibi framework, we use a distance-based outlier detection technique as follows.

Denote μ, σ the mean and standard deviation of $ascore_R^\alpha(\mathcal{T})$, respectively. A node R is determined as outlier if its distance to the "center" (i.e. μ) is larger than a pre-determined threshold ξ. We use the Mahalanobis squared distance calculated as $D(R) = (ascore_R^\alpha - \mu)^2\sigma^{-1}$. Mahalanobis squared distance $D(i)$ is used rather than Euclidian distance because Mahalanobis distance takes into account the deviation of the alibi scores and does not depend on the absolute value of the alibi score. Specifically, Mahalanobis distance has a property that the probability of $D(i) > \chi^2(\gamma)$ is γ where $\chi^2(\gamma)$ is the upper (100γ)-th percentile of a chi-square distribution.

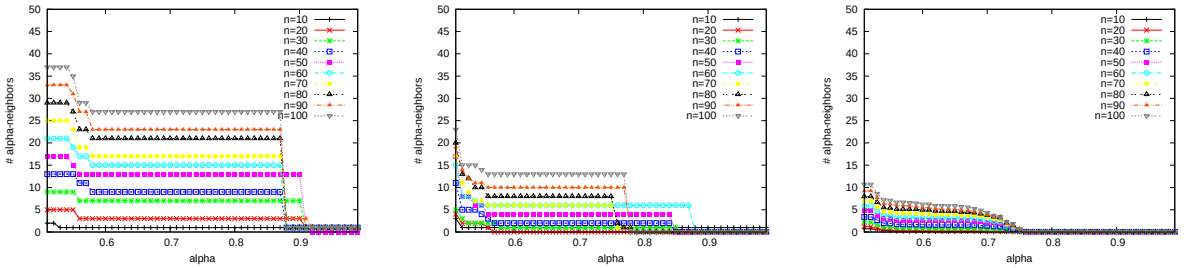
A node R is accused as an attacker if $D(R) < \mu$ and $D(R) \geq \xi$. The first condition ensures that we only accuse nodes that have alibi scores lower than the mean μ. The second condition specifies the threshold ξ in which R is accused based on its distance $D(R)$. Intuitively, lower value of ξ increases the detection probability (i.e. accusing R when R is an attacker) but also increases the false alarm probability (i.e. accusing R when R is an honest node). In the alibi framework, ξ is chosen based on the target false alarm probability γ. Specifically, $\xi = \chi^2(\gamma)$. For example, if the target false alarm probability γ is 0.1, ξ is set to $\chi^2(0.1) = 2.706$.

5. EVALUATION

5.1 Evaluation of alibi-safe topologies

There is a strong connection between alibi-safe topologies and physical topologies. Thus, it is important to determine whether a given physical topology is an alibi-safe topology. Because alibi-safe topology only relies on the received signal strength, we need to assume a propagation model in order to calculate RSSI from the sending power. In the following, we assume a log-normal shadowing path-loss model (see Appendix). To test whether a given physical topology is an (α, κ)-alibi-safe topology under a set of senders Λ, we perform following steps. First, for each sender S in Λ, we calculate the RSSI at each receiver for the message sent by S. We use the reception similarity table constructed in the experiments described in Section 3 to determine whether two receivers are α-neighbor. Then, for each receiver, we count the number of α-neighbors. If any receiver has less than κ α-neighbors under the sender S, the topology is not (α, κ)-alibi-safe topology.

We perform the alibi-safe test for three types of physical topology in a $20m \times 20m$ square: star topology, grid topology and random topology where the set Λ only has the trusted base station located at the center of the square. Figure 9 shows the results of the alibi-safe tests for the network size from 10 to 100. The x-axis is the α value and the y-axis is the *minimum* number of α-neighbors for every node (i.e κ). That means, each point represents a possible (α, κ)-alibi-safe topology. For star topology where nodes surround



(a) Star topology (b) Grid topology (c) Random topology

Figure 9: Physical topologies vs. Alibi-safe topologies

the base station, there is high chance for a node to have alibi neighbors as shown in Figure 9(a). For grid topology, it is less alibi-safe as shown in Figure 9(b) because nodes have different distance to the base station. The alibi-safe of random topologies really depends on network density as shown in Figure 9(c).

In terms of robustness to colluding attackers, star topologies are the strongest. In a star topology, each node has roughly 30% of other nodes as its 0.55-neighbors. For grid topologies and random topologies, these numbers are around 15% and 7%, respectively.

5.2 Evaluation of alibi on TOSSIM simulator

We use TOSSIM as our network simulator for a large-scale evaluation of the alibi framework. TOSSIM does not assume any radio propagation model; instead, it provides a radio abstraction between two-node communication. Specifically, it derives the packet error rate based on the empirical RSSI. Furthermore, it also has a more accurate noise generator using Close Pattern Matching (CPM) algorithm on a given noise traces. In our simulation, the noise trace is the one obtained from the experiments described in Section 3. Because TOSSIM does not provide the reception similarity of packet contents under reactive jamming attacks, we add that feature at the physical layer using the reception similarity table obtained from the experiments in Section 3. BBC-based timing channel is implemented at the MAC layer in TOSSIM and is transparent to the application layer.

There are two alibi protocols: *the Omniscient protocol (OMS)* and *the Alibi-BBC-Simhash protocol (ABS)*. The Omniscient protocol is the optimal protocol because it assumes every proof is available immediately at the central detector right after its creation. The ABS protocol uses the BBC-based timing channel and simhash.

We evaluate the two alibi protocols under three types of topologies: star topology, grid topology and random topology in a square of $20m \times 20m$ and two types of attackers: non-colluding and colluding attackers. We vary the simulation parameters as shown in Table 1. We obtain the detection accuracy (i.e., the detection probability and the false alarm rate), the detection time, the network performance and the network overhead. However, due to the space limit, we only show the results for the star topology.

Detection accuracy: Figure 10 shows the detection accuracy of the ABS protocol under non-colluding jamming attacks and colluding jamming attacks. Specifically, Figures 10(a) and 10(b) show the detection accuracy for non-colluding attackers using the same jamming rate of 0.2. It is shown that as the number of attackers increases, the detection probability decreases. This is because when the network has more attackers, the chance for an attacker to get alibis by the jamming actions of the other attackers also increase. It is shown that as the network size increases from 10 to 40, the detection probability increases. This is because the larger network size increases the number of α-neighbors and makes the attackers busier to jam. The figures also show that the false alarm rate remains similar under different network sizes and number of attackers. This shows the expected behavior of the proposed outlier detection technique. We also obtain the results for different γ but do not plot them here. Essentially, larger value of γ will lead to lower false alarms but also lower detection probability. Figures 10(c) and 10(d) show the detection accuracy for colluding attackers who collude and coordinate to target at 100% jamming rate. The trends are similar to the case of the non-colluding case. However, the detection probability is slightly worse compared to the non-colluding cases. This is due to the fact that 1) colluding jammers need to jam less than non-colluding jammers for the same aggregated jamming target and 2) each node needs more α-neighbors than the non-colluding case.

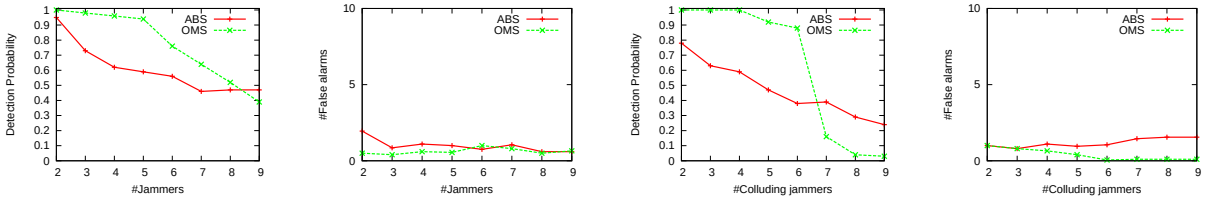
Figure 11 shows the detection performance of ABS protocol against the OMS protocol for $n = 40$. ABS protocol has a gap of around $0.2 - 0.3$ to the OMS protocol. This is because the nodes cannot send all the proofs to the central detector due to the low throughput of the timing channel.

Packet error rate: Figure 12 shows the packet error rate of the non-colluding and colluding attackers. As shown in Figure 12, with the same jamming rate of 0.2, having more non-colluding attackers only increases the packet error rate sub-linearly. In contrast, colluding attackers can coordinate to achieve 100% jamming. This shows the danger of colluding attackers over the non-colluding attackers.

Detection Time: Figure 13 shows the detection time for the case of non-colluding and colluding attackers. As shown in the figures, as the number of jammer increases, it takes longer time to identify them. The explanation is similar to the detection probability.

6. RELATED WORK

There has been plethora body of research work on jamming attacks and defenses. Jamming attacks can be classified as proactive or reactive. In the proactive jamming strategy, the attacker jams the channel without caring about the on-going communication. A typical example of this type is the continuous jamming [30][28]. This strategy is the



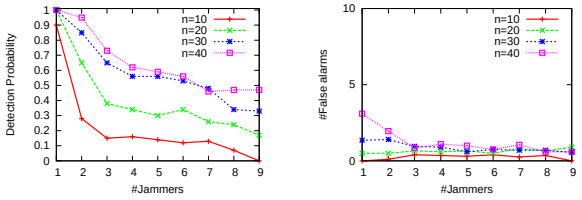
(a) Detection probability (non-colluding)

(b) False alarms (non-colluding)

(c) Detection probability (colluding)

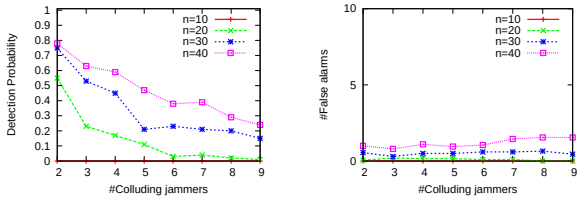
(d) False alarms (colluding)

Figure 11: Detection Accuracy: ABS vs. OMS (n=40)



(a) Detection probability (non-colluding)

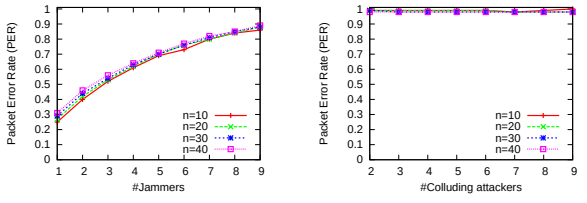
(b) False alarms (non-colluding)



(c) Detection probability (colluding)

(d) False alarms (colluding)

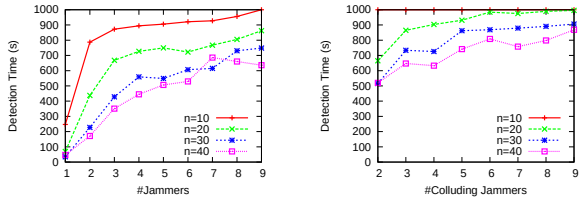
Figure 10: Detection accuracy of ABS protocol



(a) Non-colluding attackers

(b) Colluding attackers

Figure 12: Packer error rate of the ABS protocol



(a) Non-colluding attackers

(b) Colluding attackers

Figure 13: Detection time of ABS protocol

simplest way to perform a jamming attack. However, it is not energy-efficient due to the continuous jamming activity which also makes the attacker easy to detect. Reactive jamming strategy [2][19][13][17][15][20][4][12][28][23][21], in contrast, avoids these drawbacks by intelligently listening and jamming the channel. Thus, reactive jamming attacks are more difficult to detect and are more energy-efficient.

Due to the dangers of various jamming attacks, jamming defenses have gained much attention from researchers. Spread spectrum techniques have been the most effective jamming defense mechanisms. Direct Sequence Spread Spectrum (DSSS), Frequency Hopping Spread Spectrum (FHSS), and Chirp Spread Spectrum (CSS) are three common forms of spread spectrum techniques. By using the spread spectrum technique, the network can force the jammer to spend several-fold more power than if spread spectrum were not used [29][21]. However, spread spectrum does not work if the jammer knows the hopping-pattern (HP) of the FHSS, the pseudo-noise chip (PN) sequence of DSSS or the pulse-pattern of CSS (Chirp Spread Spectrum) and UWB (Ultra-wide band). Once the attacker knows such knowledge, he can jam the channel very effectively. For example, the PN of 802.11 DSSS is a common knowledge [1]. By just using the COTS 802.11 cards, the attacker can easily modify the firmware to have an effective 802.11 jammer [28]. Therefore, there is a strong motivation to avoid using one single shared secret.

One approach is to use multiple shared secrets rather just one single shared secret. In [5][7][14], the common idea is to divide nodes into multiple groups and assign one unique shared secret to each group. The assigned shared secret of a group is used to derive the hopping-pattern for that group to the base station. If the communication of a group is jammed, one of a group member must be a jammer because no nodes outside the group knows the shared secret. By further splitting the jammed group into sub-groups with different shared secrets, the network can avoid mitigate the

jamming effect caused by the compromised shared secrets.

Another approach is to remove the dependencies on the pre-shared secret of traditional spread spectrum technologies. Recently, researchers have been proposing spread spectrum communication without any pre-shared secrets. Uncoordinated Frequency Hopping (UFH) [26][27][11] allows two nodes that do not any common secret to establish a secret key for future FHSS communication. Uncoordinated Direct Sequence Spread Spectrum (UDSSS) [22] avoids jamming by randomly selecting a spread code sequence from a pool of code sequences. However, UDSSS is vulnerable to reactive jamming attacks. RD-DSSS [16] also proposes a similar technique that can be resistant to reactive jamming attacks. BBC [9] proposes a coding approach to encode data to be transmitted into "indelible" marks that can be decoded without any prior knowledge of keys. The main drawback of this approach is the low communication throughput (compared to other classic spread spectrum techniques). Therefore, such zero-shared-secret spread spectrum should only be used to facilitate the process of delivering new shared secret for the network.

Even though spread spectrum techniques have raised the bar for jamming defense, they are not sufficient to deal with the situation where attackers are compromised nodes in the network. In such a case, any attempts to deliver new shared secrets are useless because the attackers are still inside the network. Thus, it is necessary to first identify compromised nodes that launch jamming attacks to the network and then deliver the new shared secrets to un-compromised nodes only. Researchers have been looking into the problem of identifying mis-behaving/compromised nodes. In [24][12], the authors propose the detection schemes to identify mis-behaving nodes that greedily consume the bandwidth by modifying its MAC parameters. However, these detection schemes will fail to detect stealthy reactive jamming attacks considered in this paper because they rely on the identity-related clues to infer the mis-behaving nodes.

Alibi framework is a complement to the above approaches. It needs a jamming-resistant communication like BBC[9]. In terms of detection, it collects the proofs showing good behaviors of nodes instead of collecting proofs of bad behaviors of nodes[24][12].

7. CONCLUSIONS

We have presented a design and implementation of the alibi framework to deal with reactive jamming nodes. The framework relies on the novel concept of "alibi" in which the detector collects good proofs of nodes to infer the compromised nodes. We have shown a necessary condition, the alibi-safe topology condition, to check whether a given network topology is safe under reactive jamming nodes if the alibi framework is used. This test can help to deploy the network in a safe manner from reactive jamming nodes. It also helps the network designer to decide whether to add more trusted nodes into the network to make the network more alibi-safe. We have also evaluated the framework in analysis, simulation and test-bed experiments. The results show that the alibi framework can deal with both non-colluding and colluding attackers in reasonably dense networks.

Parameter	Values
Number of nodes	$n = [10 - 50]$
Number of attackers	$[1 - 9]$
Jamming rate	$[0.1 - 1.0]$
Value for α-neighbors	$\alpha = 0.55$
γ	0.05
Simulation time	1000 seconds
BBC maximum number of concurrent transmissions	50
BBC message length in time	4s
Number of BBC messages per proof-exchange period	10
Number of bits for simhash	16 bits

Table 1: Simulation parameters

8. APPENDIX

8.1 Proofs

Proof of Claim 1: Consider a set of time slots $\mathcal{T} = t_1, ..., t_{|\mathcal{T}|}$ that have sending packets. For each time slot $t \in \mathcal{T}$, a jammer j can increase its alibi point by 1 if it does not jam and at least another jammer jams. This probability is $(1-p_j) \times p_j^{agg}$. If j jams at time t, it is obvious that $\delta_j^\alpha = 0$. Thus, the overall alibi score of j is

$$ascore_j^\alpha(\mathcal{T}) = |\mathcal{T}| \times (1 - p_j) \times (1 - \prod_{i=1, i\neq j}^{k_a} (1 - p_{j_i})).$$

Similarly, for the honest node r that has the highest sending probability p_s^{max}, its accumulated alibi score is

$$ascore_r^\alpha(\mathcal{T}) = |\mathcal{T}| \times (1 - p_s^{max}) \times (1 - \prod_{i=1}^{k_a} (1 - p_{j_i})).$$

Substitute to the Inequality 2 and do some manipulations, we have

$$p_s^{max} < 1 - (1-p_j) \times \frac{(1 - \prod_{i=1, i\neq j}^{k_a}(1 - p_{j_i}))}{(1 - \prod_{i=1}^{k_a}(1 - p_{j_i}))} = 2 - \frac{1 - p_j}{p^{agg}(\mathcal{J})}.$$

$\diamond$.

Proof of Claim 2: With the definition of the alibi indicator function, if a jammer j decides to make its reception similarity with other $(\kappa - 1)$ jammers no less than α, then the κ-th reception similarity of $\Upsilon_j(\mathcal{N})$ is a reception similarity with an honest node. If j decides to make its reception similarity with other κ' jammers ($\kappa' < \kappa - 1$) no less than α, the κ-th reception similarity of $\Upsilon_j(\mathcal{N})$ is a reception similarity with another jammer. However, the value of the reception similarity is less than α and thus $\tilde{\delta}_j^\alpha = 0$. Thus, regardless of how a jammer manipulates its reception similarity with other $(\kappa - 1)$ jammers, its alibi score only increases by 1 if it really has an reception similarity with an honest node that is no less than α. This will ensure that the collusion will not bring any advantages for the attackers compared to other honest nodes in terms of obtaining alibi scores.

Using a similar analysis in the Claim 1, it can be proved that to satisfy Inequality 2, we need $p_s^{max} < p_j^{max}$ where $p_j^{max} = \max_{i \in \mathcal{J}} p_{j_i}$. Thus, if $p_s^{max} < \frac{p_\mathcal{T}}{k_a} \leq p_j^{max}$, the alibi scheme can identify at least one attacker. $\diamond$

8.2 Log-normal shadowing path-loss model

In the log-normal shadowing path-loss model, $P_R = P_S - PL(d)$ where $PL(d) = PL(d_0) + 10\alpha_{pl} \log_{10} \frac{d}{d_0} + N_\sigma$ is the path-loss of the radio over a distance d; P_R is the power of the signal at the receiver; P_S is the power of signal at the sender; d_0 is the reference distance and $PL(d_0)$ is the power

decay for this distance; α_{pl} is the signal decay factor. N_σ is the zero-mean Gaussian (in *db*) with standard deviation σ representing the multi-path effects.

9. REFERENCES

[1] IEEE standard 802.11, September 2004.

[2] G. Alnifie and R. Simon. A multi-channel defense against jamming attacks in wireless sensor networks. In *ACM workshop on QoS and security for wireless and mobile networks (Q2SWinet)*, 2007.

[3] A. Z. Broder, M. Charikar, A. M. Frieze, and M. Mitzenmacher. Min-wise independent permutations (extended abstract). In *ACM symposium on Theory of computing (STOC)*, 1998.

[4] T. X. Brown, J. E. James, and A. Sethi. Jamming and sensing of encrypted wireless ad hoc networks. In *ACM International symposium on Mobile ad hoc networking and computing (MobiHoc)*, 2006.

[5] A. Chan, X. Liu, G. Noubir, and B. Thapa. Control channel jamming: Resilience and identification of traitors. In *IEEE International Symposium on Information Theory (ISIT), 2007*, 2007.

[6] M. S. Charikar. Similarity estimation techniques from rounding algorithms. In *ACM symposium on Theory of computing (STOC)*, 2002.

[7] J. Chiang and Y.-C. Hu. Dynamic jamming mitigation for wireless broadcast networks. In *IEEE Conference on Computer Communications (INFOCOM)*, 2008.

[8] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In *Annual symposium on Computational geometry (SCG)*, 2004.

[9] L. C. B. III, W. L. Bahn, , and M. D. Collins. Jam-resistant communication without shared secrets through the use of concurrent codes. Technical report, U.S. Air Force Academy, 2007.

[10] P. Indyk and R. Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *ACM symposium on Theory of computing (STOC)*, 1998.

[11] T. Jin, G. Noubir, and B. Thapa. Zero pre-shared secret key establishment in the presence of jammers. In *ACM International Symposium on Mobile ad hoc networking and computing (Mobihoc)*, 2009.

[12] P. Kyasanur and N. H. Vaidya. Detection and handling of MAC layer misbehavior in wireless networks. In *International Conference on Dependable Systems and Networks (DSN)*, 2003.

[13] Y. W. Law, L. van Hoesel, J. Doumen, P. Hartel, and P. Havinga. Energy-efficient link-layer jamming attacks against wireless sensor network mac protocols. In *ACM workshop on Security of ad hoc and sensor networks (SASN)*, 2005.

[14] L. Lazos, S. Liu, , and M. Krunz. Mitigating control-channel jamming attacks in multi-channel ad hoc networks. In *ACM Conference on Wireless network security (WiSec)*, 2009.

[15] M. Li, I. Koutsopoulos, and R. Poovendran. Optimal jamming attacks and network defense policies in wireless sensor networks. In *IEEE Confrence on Computer Communications (INFOCOM)*, 2007.

[16] Y. Liu, P. Ning, H. Dai, and A. Liu. Randomized differential dsss: Jamming-resistant wireless broadcast communication. In *IEEE Conference on Computer Communications (INFOCOM)*, 2010.

[17] R. K. Mallik, R. A. Scholtz, and G. P. Papavassilopoulos. Analysis of an on-off jamming situation as a dynamic game. *IEEE Transaction on Communications*, 48:1360–1373, 2000.

[18] G. S. Manku, A. Jain, and A. D. Sarma. Detecting near-duplicates for web crawling. In *International conference on World Wide Web (WWW)*, 2007.

[19] J. M. McCune, E. Shi, A. Perrig, and M. K. Reiter. Detection of denial-of-message attacks on sensor network broadcasts. In *IEEE Symposium on Security and Privacy*, 2005.

[20] V. Navda, A. Bohra, S. Ganguly, and D. Rubenstein. Using channel hopping to increase 802.11 resilience to jamming attacks. In *IEEE Conference on Computer Communications (INFOCOM Mini-conference)*, Anchorage, AK, May 2007.

[21] R. Negi and A. Perrig. Jamming analysis of mac protocols. Technical report, Carnegie Mellon Technical Memo, 2003.

[22] C. Pöpper, M. Strasser, and S. Čapkun. Jamming-resistant broadcast communication without shared key. In *USENIX security Symposium*, 2009.

[23] S. Radosavac, J. S. Baras, and I. Koutsopoulos. A framework for MAC protocol misbehavior detection in wireless networks. In *ACM workshop on Wireless security (WiSe)*, 2005.

[24] M. Raya, J.-P. Hubaux, and I. Aad. Domino: A system to detect greedy behavior in IEEE 802.11 hotspots. In *International conference on Mobile systems, applications, and services (MobiSys)*.

[25] L. Sang and A. Arora. Capabilities of low-power wireless jammers. In *IEEE Conference on Computer Communications (INFOCOM mini-conference)*, 2009.

[26] M. Strasser, C. Pöpper, and S. Čapkun. Efficient uncoordinated fhss anti-jamming communication. In *ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, 2009.

[27] M. Strasser, C. Pöpper, S. Čapkun, and M. Čagalj. Jamming-resistant key establishment using uncoordinated frequency hopping. In *IEEE Symposium on Security and Privacy (Oakland)*, 2008.

[28] D. J. Thuente and M. Acharya. Intelligent jamming in wireless networks with applications to 802.11b and other networks. In *Military Communications Conference (MILCOM)*, 2006.

[29] M. Z. Win and R. A. Scholtz. Ultra-wide bandwidth time-hopping spread-spectrum impulse radio for wireless multiple-access communications. *IEEE Transactions on Communications*, 48, 2000.

[30] A. D. Wood, J. A. Stankovic, and G. Zhou. Deejam: Defeating energy-efficient jamming in ieee 802.15.4-based wireless networks. In *IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, 2007.

[31] W. Xu, W. Trappe, Y. Zhang, and T. Wood. The feasibility of launching and detecting jamming attacks in wireless networks. In *ACM international symposium*

on Mobile ad hoc networking and computing (MobiHoc), 2005.