Identifying insider-based jammers in multi-channel wireless networks

Hoang Nguyen, Thadpong Pongthawornkamol and Klara Nahrstedt
Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, IL 61801
E-mail:{hnguyen5,tpongth2,klara}@illinois.edu

Abstract

We consider the problem of identifying insider-based attacks in the form of jammers in multi-channel wireless networks, where jammers have the inside knowledge of frequency hopping patterns and any protocols used in the wireless network. We propose a novel technique, called "**alibi**", to identify the insider-based jammers in multi-channel wireless networks. 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. Starting from such simple concept, we develop an alibi framework to cope with insider-based jamming attackers in various situations including single/multiple jammer and lossy channels. We evaluate the framework according to several properties such as accuracy, detection time and network performance via TOSSIM simulation and analysis. The overall results of these protocols show a promising research direction to deal with insider-based jamming attacks.[1]

I. 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 probability of successful message receptions.

While there are various ways to carry out jamming attacks (cf. see Section VII), we consider a so-called insider-based jamming attack as follows. In an insider-based jamming attack, there are several nodes getting compromised either before the deployment or during the operation of the network. These compromised nodes are used to jam the network. The dangers of this type of attacks are two-fold. First, the attackers have shared knowledge that is supposed to remain secret within the network such as shared keys, shared hopping pattern and/or any protocols used by the network. Second, the attackers can be very stealthy if they want to stay undetected for long time and do further damage to the network. The stealthy nature of the attack also helps the attackers to conserve the energy if the devices are powered by batteries.

Most of the work in the jamming defense literature can only deal with outsider-based jamming attacks (cf. see Section VII). By "outsider", we mean the attackers with zero knowledge of any shared secrets among nodes in the network. One of the most effective ways to prevent such an outsider jammer is spread spectrum technique. By hopping the carrier frequency (frequency-hopping spread spectrum - FHSS) or spreading its signal in time (direct-sequence spread spectrum - DSSS), the network can force the jammer to spend several-fold more power than if spread spectrum were not used [1]. However, spread spectrum does not work if the attacker knows the hopping-pattern (HP) of the FHSS or the pseudo-noise chip (PN) sequence of DSSS. An insider-based jammer can easily obtain the shared hopping pattern of the network and jam very effectively. Thus, dealing with insider-based attackers is far more challenging than the outsider-based ones.

In this paper, we focus on the problem of *identifying* the insider-based jammers. Note that there is a difference between detection and identification. Detection is a weaker concept than identification. Detection only means that a jammer exists. Identification means that a specific node is the jammer. We propose a novel technique, called "**alibi**", to identify insider-based jammers in multi-channel wireless networks. By definition, *"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"*. In the context of jamming attacks, *honest nodes try to obtain alibis showing that they were doing legitimate actions observed by some witnesses while the jamming action took place*. From this core concept of alibi, we develop a framework, called **alibi framework**, to identify insider-based jammers. The key principle in building the alibi framework is that there has to be a significant difference in the way of obtaining alibis between honest nodes and attackers. For example, alibis can be defined in the way that only honest nodes can obtain alibis while attackers cannot obtain any alibis. In this way, when all honest nodes obtain at least one alibi, attackers are identified.

Even though alibi framework starts from a simple concept, there are numerous challenges to make it work in the context of identifying insider-based jammers in multi-channel wireless networks. First, because there is no clear distinction between a "normal-corrupted" packet (i.e. a packet corrupted by an unintentional collision) and a jammed packet (i.e. a packet corrupted by an intentional jamming action), we have to deal with "false" alibis. False alibis are alibis that can be falsely generated from

[1] This material is based upon work supported by the Department of Energy under Grant 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.

mis-identified packet corruption events. Thus, attackers can exploit this fact to get false alibis and stay undetected. Second, alibi is susceptible to slander attacks. In a slander attack, if the behaviors of honest nodes are *completely known* by the attackers, the attackers can *deterministically* avoid committing jamming actions whenever those honest nodes may potentially obtain alibis. By doing this strategy, the victim nodes will never be able to obtain any alibis and thus become as mis-identified as attackers. Third, there might be multiple attackers in the network. A jam event caused by one attacker can help another attacker to get an alibi. Lastly, alibi framework has to be able to cope with these challenges without much performance degradation of the network.

In our previous work [2], we only deal with the case of a single insider-based jammer in a single-hop wireless network - the simplest case of a challenging problem. In this paper, we will deal with the case of single/multiple attackers in multi-channel wireless networks.

The rest of the paper is organized as follows. We start with the system model including network model, jammer model and problem formulation in Section II. We present the general alibi framework including the basic ideas and desired properties for any alibi-based protocols in Section III. We then give the analysis of alibi in Section IV. We evaluate the framework in Section VI. Finally, we conclude our paper in Section VIII.

II. System Model

A. *Network Model*

We consider a multi-hop multi-channel ad hoc networks as shown in Figure 1. Several nodes are compromised and become insider-based jammers. Each jammer can affect at least one victim node in the network, i.e. the attacker can disrupt any *packet reception* at victim nodes.

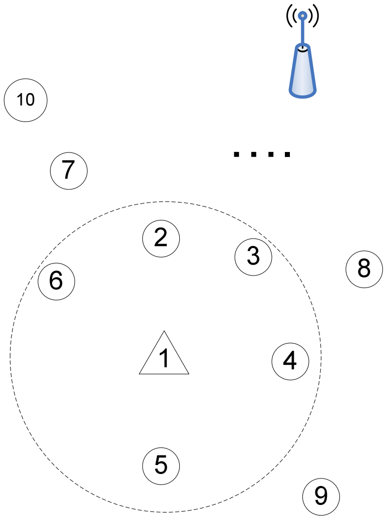


Fig. 1. 10 nodes and a trusted BS. Nodes 2 to 6 are in the jamming range of node 1. Nodes 7, 8 and 9 are the direct neighbors of the jammed nodes.

Each node is equipped with a **single transceiver**. That means, a node cannot send and receive simultaneously (i.e. half-duplex). There will be also non-negligible transmit-to-receive and receive-to-transmit turn-around time. The channel switching delay of a node is also assumed to be non-negligible.

Nodes in the network have a set orthogonal channels $\mathcal{C} = \{c_1, ...c_{|\mathcal{C}|}\}$ that they can switch to. They use a multi-channel MAC protocol such as Slotted Seeded Channel Hopping (SSCH) [3] or McMAC [4]. The main reasons for the suggestion of these multi-channel MAC protocols are:

1) They improve the capacity of the wireless networks. They generally do not require any special hardware other than a commodity wireless cards (e.g. 802.11).
2) They are multiple-rendezvous protocols in which multiple device pairs can make agreements simultaneously on distinct channels [4]. This eliminates the problem of single control channel bottleneck - a sweet spot for the jammer to target on. This is very important because a jammed control channel may drastically reduce the effective throughput close to zero [5][6].

In the design of this framework, we use SSCH [3]. In SSCH, each node i has a set of n_i^{ssch} randomly generated channel seeds. Each channel seed is a pair of *(x, a)*, where x $(x \in \mathcal{C})$ is the initial channel and a $(a \in \mathcal{C})$ is the seed of the schedule.

Each channel seed is used to calculate the new channel from the old channel in each time slot. Specifically, the new channel x_{new} is calculated from the old channel x_{old} as

$$x_{new} = (x_{old} + a) \mod |\mathcal{C}|.$$

, where $|\mathcal{C}|$ is the number of channels.

Channel seeds are used in round-robin manner. Specifically, in seed slot s $(s = 0, \ldots, n_i^{ssch} - 1)$, node i will hop to the channel calculated from the channel seed s-th. Figure 2 illustrates SSCH schedules for two nodes A and B. Node A has two channel seeds $(1, 2)$ and $(2, 1)$. Node B also has two channel seeds $(1, 2)$ and $(0, 1)$. A and B use the two channel seeds alternatively in each time slot. Initially, in the first time slot, A uses the first channel seed $(1, 2)$ and thus goes to channel 1. Similarly, B goes to channel 1 in the first time slot. In the second time slot, A and B will use the second channel seed. That means, A, B will go to channel $2, 0$, respectively. In the third time slot, A will use the first channel seed $(1, 2)$. Because the old channel corresponding to this channel seed that A used is 1, the new channel that A uses is $(1+2) \mod 3 = 0$. Similarly, in the third time slot, B's new channel is $(1 + 2) \mod 3 = 0$. For the time slot $4, 5, 6$, the new channels for A and B are calculated in a similar manner. The time slot 7 (i.e., the slot with shaded background) is the parity slot in which the channels that A and B use are set to the seed being used, instead of the calculated channel like previous time slots. The reason for the parity time slots is to ensure that any pair of nodes will meet occasionally, even when their channel seeds have different initial channel (i.e., x_i) and the same seed (i.e., a_i).

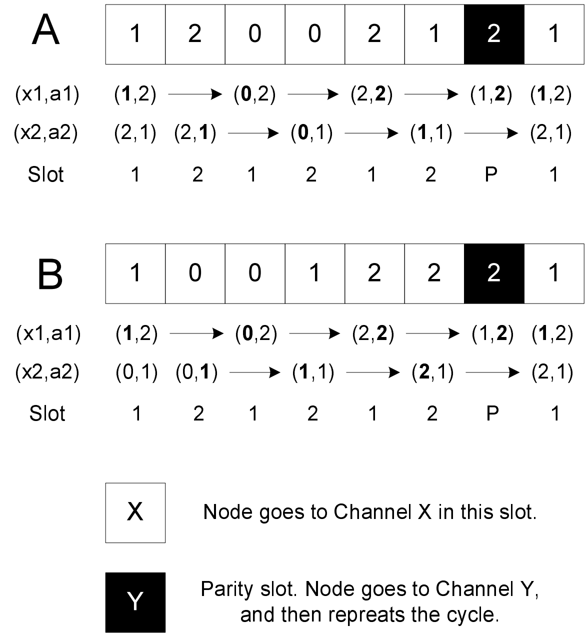

A
1 2 0 0 2 1 2 1
(x1,a1) (1,2) (0,2) (2,2) (1,2) (1,2)
(x2,a2) (2,1) (2,1) (0,1) (1,1) (2,1)
Slot 1 2 1 2 1 2 P 1
B
1 0 0 1 2 2 2 1
(x1,a1) (1,2) (0,2) (2,2) (1,2) (1,2)
(x2,a2) (0,1) (0,1) (1,1) (2,1) (2,1)
Slot 1 2 1 2 1 2 P 1
X
Node goes to Channel X in this slot.
Y
Parity slot. Node goes to Channel Y, and then repreats the cycle.

Fig. 2. An illustration of channel hopping schedules for two nodes A and B with 3 channels and two channel seeds. P denotes the parity slot.

If the number of channels is a prime number, SSCH schedules have a mathematical property that two nodes will guarantee to meet at least once per cycle. Specifically, consider any pair of two channel (x, a) and (x', a'), the following properties hold as shown in [3].

- If $x = x'$ and $a = a'$, they are synchronized with each other.
- If $x = x'$ and $a \neq a'$, they will meet exactly once per cycle.
- If $x \neq x'$ and $a = a'$, they will meet exactly once per cycle at the parity slot.

In the third case above, SSCH has to use parity slots where nodes hop to the channel equal to the seed to prevent the off-schedule of nodes sharing the same seed.

When joining the network, each node will randomly generate a set of channel seeds. Nodes periodically broadcast their schedule, represented by the set of generated channel seeds. For node A wanting to send to node B, A first learns the schedule of node B when they are occasionally in the same channel. Then, A will change one of its channel seeds to one of the channel seeds broadcasted by B. In this way, A will partially synchronize with B. When A is in the same channel with B, it uses CSMA/CA to transmit the packets to B to minimize the collisions with other senders in the same channel. Also, when A changes the channel seed to synchronize with B for its transmission, A will mark that channel seed as "sending seed". Node B, when seeing a channel seed that always has packets destined to it, will also mark that channel seed as "receiving seed" and *lock* that channel seed. A locked channel seed cannot be changed for any sending purpose. However, if a node wants to send packets and all channel seeds are locked, it will pick a channel seed and change it to the intended receiver's channel seed. In the example above, assuming A wants to send to B, A is partially synchronized with B because their first channel seeds are identical. A will mark its first channel seed (i.e., $(1, 2)$) as "sending seed". B will mark its first channel seed (i.e., $(1, 2)$) as "receiving seed".

B. Jamming Model

A packet consists of a set of symbols. A jammed packet has several corrupted symbols whose locations are unknown. However, the corruption of a packet can be detected by checking its cyclic redundancy check (CRC). Corrupted packets, due to either jamming or unintentional interference, are still delivered to the upper layer to provide information on reasons of corruption [7]. We assume that any two receivers listening on the same channel at the same time will receive the same packet content with probability p_r, regardless whether the packet is corrupted or not. However, for the sake of simplicity of our analysis, we assume $p_r = 1$.

C. Jammer Model

The jammer uses reactive jamming strategy [8]. It will start sending a packet whenever it senses a preamble signal from any senders. The jammer is assumed not to leave any trivial information to trace it back such as its MAC address. The jammer is assumed to be able to jam multiple channels by doing channel switching. That means, a jammer may jam one channel, which takes him sometime to corrupt few symbols, and switch to another channel to do jamming. However, we assume within a time slot s, jammers cannot jam all channels. This is a necessary condition for any in-band anti-jamming defenses. Otherwise, no in-band communication is possible and thus no in-band jamming defense is possible.

D. Detection Model

We assume there is a trusted entity G (e.g. a base station) where nodes can report to. We also assume that when necessary, nodes can have a secure communication to G by using either public/private key or pre-shared key.

E. Problem Formulation

Consider the network of nodes that are directly affected by the jammers. The affected nodes are those that have high packet error rates (PER). This is similar to the jammed-area mapping service proposed in [9]. These jammed nodes and their direct neighbors are put into a suspect list. Let us denote $\mathcal{N}$ the set of suspect jamming nodes and $n = |\mathcal{N}|$. For example, for the network topology shown in Figure 1, node 1 to node 9 are put into suspect list, i.e. $n = 9$. The problem is to find out all jammers in the suspect list.

III. Alibi's Framework

A. Alibi

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. The alibi framework is built up from this core concept. Our alibi definition is as follows.

Definition 1 (Alibi & Defendant)**.** *An alibi for a defendant is a proof including time and channel information which shows that the defendant was doing legitimate actions at the time the jamming action was committed. A legitimate action is either sending or receiving a packet.*

Definition 2 (Proof & Witness)**.** *A witness is a node showing proofs of a defendant doing an action at a specific time.*

In our alibi framework, a defendant cannot claim an alibi by itself. Rather, alibis for defendants are generated from proofs collected by witnesses. Thus, more number of witnesses for an alibi also increases the trustworthiness of that alibi.

From Def. 1, there are two types of alibis: sending-based alibis and receiving-based alibis.

Definition 3 (Sending-based alibi (S-alibi))**.** *A sending-based alibi for a node shows that the defendant was observed, by several witnesses, sending an uncorrupted packet over one* whole *time slot in one channel at the time the jamming action took place in another channel.*

This definition exploits the fact that a jammer cannot jam one channel and send an uncorrupted packet of one time slot in another channel simultaneously.

Definition 4 (Receiving-based alibi (R-alibi))**.** *A receiving-based alibi for a node shows that the defendant 1) was receiving a jammed packet, by showing a (hashed) packet content that matches with the (hashed) packet content received by other witnesses or 2) was receiving an uncorrupted packet by showing a correct CRC check.*

This definition exploits half-duplex nature of the jammer: it cannot both send and receive a packet simultaneously. In the receiving-based alibis, an R-defendant of a jamming event is also an R-witness of other R-defendants of the same event. Simply said, a node that can show that it was receiving a corrupted packet or uncorrupted packet while there was a jamming event will get an R-alibi.

B. The principle in using alibis to identify attackers

The key principle in using alibis to identify attackers is that there has to be significant difference of alibis obtained by good nodes and attackers. The difference can be deterministic such as "only good nodes can obtain alibi while attackers cannot" or statistical such as "a good node statistically obtains higher number of alibis than an attacker". With these differences, as time goes on, the attackers will be eventually identified. If attackers can manage to remove the differences, the alibi framework will fail to differentiate the good nodes and the attackers. Thus, it is very important to have the right definition and implementation of alibis.

C. Slander attacks

If the behaviors of defendants are deterministic, attackers can do slander attacks on any victim nodes as follows. Whenever victim nodes become defendants, the attackers will not commit any jamming actions. By doing this, the victim nodes cannot obtain any alibis and thus have no difference with the attackers. This violates the principle of the alibi framework. Thus, to avoid slander attackers, we have to introduce randomness into defendants' behaviors. However, because the defendants in S-alibis have to actually send packets to obtain alibis, introducing randomness in their behaviors also introduce additional collisions in the network. The collisions not only degrade the network performance but also cause additional "false" alibis because collisions can be considered as "unintentional" jamming actions. This is not a problem for defendants in R-alibis as they only have to listen to channels. Thus, even though there might be ways to mitigate the problem of S-alibis to cope with slander attacks, **we will only discuss R-alibis in this work due to its advantage in dealing with slander attacks**. Thus, alibis refer to R-alibis from now on, unless specified.

D. Alibi protocol

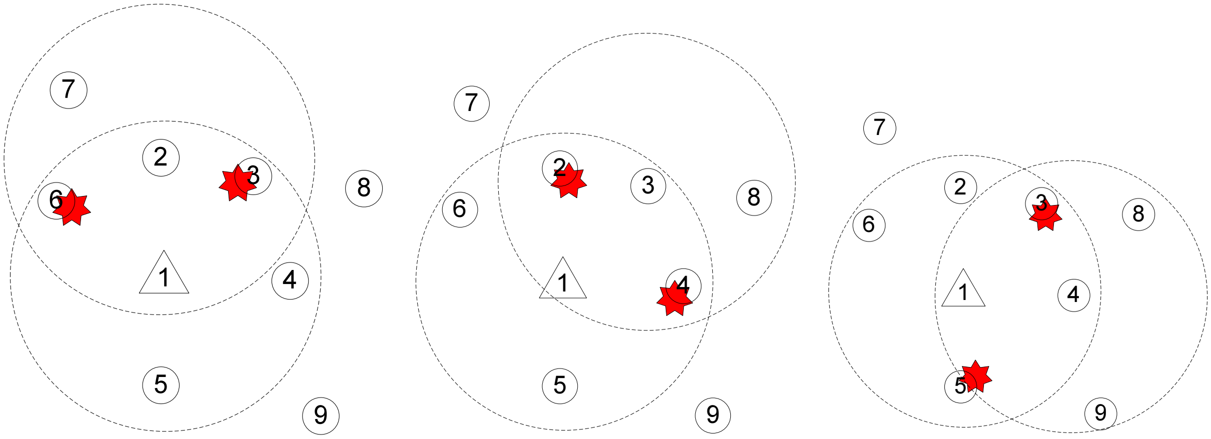
When an honest node is idle in any time slots (i.e. no sending or receiving), it switches to a uniformly random channel in Γ with probability p_w to become an R-witness (also R-defendant). For a node, increasing p_w will increase the probability of being R-witness and potentially increase the probability of getting alibis but also decrease its network performance. For example, if a node always has a packet to send, $p_w = 0.2$ means it will lose 20% of its either sending or receiving capabilities. Thus, p_w can be used as a parameter to control the trade-off between the probability of getting alibis and the degradation of the network performance.

When a node N_i becomes a R-witness in a time slot t on channel c, it will receive the whole packet content p regardless of whether the packet is decodable or not. It will get the hashed content of the received packet by using any good hash function H (e.g. CRC, SHA1 or MD5) and create a proof m sent to G in the form of $(t, c, H(p))$, which is then encrypted and sent to G. The central trusted detector G collects proofs m and generates alibis for each node in the network according to R-alibi definitions. It maintains an "alibi score" for each node. Specifically, whenever a group of nodes show the matching proofs, G will increase the alibi score of every node in that group by 1. When the alibi score of a node is too low compared to other nodes (see Section III-E), it will be accused as attackers. Figure 3 gives an example of how the alibi works for nodes in the jammed region.

E. Distance-based Outlier Detection Algorithms

Given a set of alibi scores $ascore_r(\mathcal{T}), \forall r \in \mathcal{N}$ calculated over 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. In the alibi framework, we use a distance-based outlier detection technique as follows.

Denote μ, σ the mean and standard deviation of $ascore_r(\mathcal{T})$, 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 $d(r) = (ascore_r - \mu)^2\sigma^{-1}$. Mahalanobis squared distance $d(i)$ is used rather than Euclidian distance because Mahalanobis distance normalizes the original distances into the scale-invariant distances that can be compared to the χ^2 distribution. 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 $\chi^2(0.1) = 2.706$.



(a) Node 2's message is jammed by node 1. Node 3 and 6 get an alibi.

(b) Node 3's message is jammed by node 1. Node 2 and 4 get an alibi.

(c) Node 4's message is jammed by node 1. Node 3 and 5 get alibis.

Fig. 3. An example of the alibi scheme: Node 1 is the jammer and cannot obtain any alibis while the rest of nodes eventually can get at least one alibi.

F. Jamming-resistant communication for proof-exchange

To make the proof-exchange protocol jamming-resistant, we use a similar technique in the UFH system [10]. During the proof-exchange period, honest nodes randomly pick a channel from $\mathcal{C}$ in each time slot. Note that nodes still use CSMA/CA to send proof packets to reduce collisions. When getting acknowledged from G for a proof packet, a node moves on to the next proof packet. The purpose of the random selection of the channel during the proof-exchange period is to make it harder for the attackers to jam. More importantly, it will prevent the jammers to perform slander attacks on any node because the sending pattern is random and unknown to the attackers. However, because the schedule of G is known to the attackers, it is still possible that attackers specifically target the jamming attacks on node G to block any possible communication to G. To avoid this situation, node G has to randomize its schedule as well. Specifically, node G starts its *proof-exchange* period in which it will randomize its schedule under two conditions: 1) it receives a significant number of corrupted packets and 2) it receives a significant number of distinct nodes sending proof packets to it. The first condition covers the situation where node G is under jamming attack. The second condition covers the situation where the jamming attackers target all other nodes but node G so that G does not start the proof-exchange mode. Both conditions involve thresholds which are system parameters. When switching to proof-exchange period, node G will randomly pick a channel in each time slot. Furthermore, it will only stay in the receiving mode until it can identify the jammers.

G. Desired Properties

Desired properties for any alibi-based protocols are as follows.

Detection time: This property is concerned about the time to detect the attackers. Specifically, any alibi-based protocols must show that the detection is bounded.

Accuracy: This property is concerned about the false alarm and miss detection. Specifically, any alibi-based protocols must show that $P[false\_alarm]$ and $P[miss\_detection]$ are bounded. In our scheme, we will show later that if all channels are lossless, there will be no false alarm and miss detection. However, in case one of the channels is lossy, false alarm and miss detection will happen and be bounded with the loss rate of the channel.

Availability/Network performance: This property defines fraction of time the channels is available for communication. Intuitively, the more the attackers jam the channels, the less availability of the channels is and the faster the attackers get detected. Thus, this becomes the trade-off for the attackers.

Scalability: This property specifies how much overhead is incurred in an alibi-based protocol and thus how well it scales with the network size. Specifically, it measures how many extra messages have to be sent for alibi schemes for a given network size.

IV. Dealing with non-colluding attackers

In this section, we give the analysis of the case of the non-colluding attackers under the basic alibi protocol proposed in Section III-D. The analysis for the case of single attacker can be found in [2]. Note that in this analysis, we assume a lossless channel condition.

Lemma 1 (Identifying non-colluding attackers)**.** *In the network $\mathcal{N}$ with the set of k non-colluding jammers $\mathcal{J} = j_1, \ldots, j_k$ with the jamming rate $p_{j_1}^{jam}, \ldots, p_{j_k}^{jam}$, the R-alibi framework can identify any jammer $j \in \mathcal{J}$ if*

$$p_{rmin}^{witness} \times (1 - (1 - \frac{1 - p_j^{jam}}{|\mathcal{C}|})(\frac{1-b}{1 - \frac{p_{rmin}^{witness}}{|\mathcal{C}|}}))$$

$$> \frac{(1 - p_j^{jam})ab}{(1 - (1 - \frac{p_j^{jam}}{|\mathcal{C}|})(1-a))}$$

, where $p_{rmin}^{witness}$ is the minimum probability of being witness of all honest nodes in the network,

$$a = 1 - \prod_{i \in \mathcal{J} \backslash j} (1 - \frac{p_i^{jam}}{|\mathcal{C}|}) \text{ and } b = 1 - \prod_{i \in \mathcal{N} \backslash j} (1 - \frac{p_i^{witness}}{|\mathcal{C}|})$$

Proof: Consider a honest node $r \in \mathcal{N} \backslash \mathcal{J}$ and a jammer $j \in \mathcal{J}$. In any time slot, node r will get an alibi if all following three events happen:

- r becomes a witness
- $A(\mathcal{J})$: at least an attacker in $\mathcal{J}$ jams on the channel r is witnessing
- $B(\mathcal{N} \backslash r)$: at least another node witnesses on the channel r is witnessing.

The first condition happens with probability $p_r^{witness}$. The second condition happens with probability $P[A(\mathcal{J})] = 1 - \prod_{i \in \mathcal{J}}(1 - \frac{p_i^{jam}}{|\mathcal{C}|})$, where $(1 - \frac{p_i^{jam}}{|\mathcal{C}|})$ is the probability that attacker $i \in \mathcal{J}$ does not jam on a particular channel. The third condition happens with probability $P[B(\mathcal{N} \backslash r)] = 1 - \prod_{i \in \mathcal{N} \backslash r}(1 - \frac{p_i^{witness}}{|\mathcal{C}|})$, where $(1 - \frac{p_i^{witness}}{|\mathcal{C}|})$ is the probability that node i does not witness on a particular channel. Thus, in an time slot, node r gets an alibi with the probability

$$p_r^{witness} \times P[A(\mathcal{J})] \times P[B(\mathcal{N} \backslash r)]. \tag{1}$$

Similarly, for the attacker j, the probability that it can obtain an R-alibi is

$$p_j^{witness} \times P[A(\mathcal{J} \backslash j)] \times P[B(\mathcal{N} \backslash j)]. \tag{2}$$

To have r obtains alibis faster than j, we need the term in Eq. 1 to be greater than the term in Eq. 2.

$$p_r^{witness} \times P[A(\mathcal{J})] \times P[B(\mathcal{N} \backslash r)] > p_j^{witness} \times P[A(\mathcal{J} \backslash j)] \times P[B(\mathcal{N} \backslash j)].$$

It is obviously best for the attacker j to set $p_j^{witness} = 1 - p_j^{jam}$. Thus, the condition becomes

$$p_r^{witness} \times P[A(\mathcal{J})] \times P[B(\mathcal{N} \backslash r)] > (1 - p_j^{jam}) \times P[A(\mathcal{J} \backslash j)] \times P[B(\mathcal{N} \backslash j)] \tag{3}$$

.

We have $P[A(\mathcal{J} \backslash j)] = 1 - \prod_{i \in \mathcal{J} \backslash j}(1 - \frac{p_i^{jam}}{|\mathcal{C}|})$. Thus,

$$\prod_{i \in \mathcal{J} \backslash j} (1 - \frac{p_i^{jam}}{|\mathcal{C}|}) = 1 - P[A(\mathcal{J} \backslash j)]$$

. By multiplying both sides by $(1 - \frac{p_j^{jam}}{|\mathcal{C}|})$, we obtain

$$\prod_{i \in \mathcal{J}} (1 - \frac{p_i^{jam}}{|\mathcal{C}|}) = (1 - \frac{p_j^{jam}}{|\mathcal{C}|}) \times (1 - P[A(\mathcal{J} \backslash j)])$$

Therefore,

$$P[A(\mathcal{J})] = 1 - \prod_{i \in \mathcal{J}} (1 - \frac{p_i^{jam}}{|\mathcal{C}|}) = 1 - (1 - \frac{p_j^{jam}}{|\mathcal{C}|})(1 - P[A(\mathcal{J} \backslash j)]).$$

Denote $a = P[A(\mathcal{J} \backslash j))]$. The condition in Eq. 3 becomes

$$p_r^{witness} \times (1 - \frac{p_j^{jam}}{|\mathcal{C}|})(1-a)) \times P[B(\mathcal{N} \backslash r)] > (1 - p_j^{jam}) \times a \times P[B(\mathcal{N} \backslash j)] \tag{4}$$

.

Denote $b = P[B(\mathcal{N}\backslash j)] = 1 - \prod_{i\in\mathcal{N}\backslash j}(1-\frac{p_i^{witness}}{|\mathcal{C}|}) = 1-(1-\frac{p_r^{witness}}{|\mathcal{C}|})\prod_{i\in\mathcal{N}\backslash\{j,r\}}(1-\frac{p_i^{witness}}{|\mathcal{C}|})$. Thus,

$$\prod_{i\in\mathcal{N}\backslash\{j,r\}}(1-\frac{p_i^{witness}}{|\mathcal{C}|}) = \frac{1-b}{1-\frac{p_r^{witness}}{|\mathcal{C}|}}$$

. By substituting the above term and $1-p_j^{witness} = p_j^{jam}$ into $P[B(\mathcal{N}\backslash r)]$, we have

$$P[B(\mathcal{N}\backslash r)] = 1-\prod_{i\in\mathcal{N}\backslash r}(1-\frac{p_i^{witness}}{|\mathcal{C}|}) = 1-(1-\frac{p_j^{witness}}{|\mathcal{C}|})\prod_{i\in\mathcal{N}\backslash\{j,r\}}(1-\frac{p_i^{witness}}{|\mathcal{C}|})$$

$$= 1-(1-\frac{1-p_j^{jam}}{|\mathcal{C}|})(\frac{1-b}{1-\frac{p_r^{witness}}{|\mathcal{C}|}})$$

Thus, the condition in Eq. 4 becomes

$$p_r^{witness}\times(1-(1-\frac{p_j^{jam}}{|\mathcal{C}|})(1-a))\times(1-(1-\frac{1-p_j^{jam}}{|\mathcal{C}|})(\frac{1-b}{1-\frac{p_r^{witness}}{|\mathcal{C}|}})) > (1-p_j^{jam})ab$$

◇

V. Dealing with colluding attackers

In this case, we consider a set of k colluding jammers $\mathcal{J} = j_1, \ldots, j_k$ *that can share any information among themselves immediately by any means of communication*. There are several problems when collusion is possible. The first problem is that attackers can coherently lie about their proofs (i.e. hashed content of jammed packets) to create "fake" R-alibis. To cope with this, we require at least $k_{alibi} = k+1$ witnesses presenting same hashed content of a jamming packet to create a R-alibi for all witnesses. k_{alibi} is referred to as alibi threshold. The second problem is that *attackers can share alibis*. For example, let us consider the case of 2 colluding attackers. One attacker jams the network and the other attacker collects alibis. If there is no alibi-sharing, the jamming one can be detected by our previous proposed detection schemes. However, if alibis are shared to the jamming one, both attackers can get alibis at the rate of other normal nodes and thus cannot be detected. In what follows, we will discuss how to cope with colluding attackers using the concept of R-chains.

A. R-chains

Consider an attacker j_1 who jams on channel $c_1 \in \mathcal{C}$ at time slot t. To limit the possibility that j_1 gets a shared alibi from another attacker $j_2 (j_2 \in \mathcal{J}\backslash j_1)$ which *correctly* obtains an alibi on channel $c_2 (c_2 \in \mathcal{C}\backslash c_1)$ at time slot t, we require j_1 has to be able to *explain its presence* on channel c_2 at time t. If j_1's explanation can be verified, j_1's alibi is valid. Thus, for a node to be able to explain its presence at time slot t on channel $c \in \mathcal{C}$, it has to declare its *sequence of being R-defendant* before time slot t. Let us denote *R-chain(i,s,l)* the sequence of l pairs (c_1, s)...$(c_l, s+l-1)$ in which node i becomes an R-defendant on channel c_x at time slot $s+x-1$ $(x = 1..l)$. Thus, *R-chain(i,s,l)* can be used to verify the validity of any R-alibi for node i at any time in between $[s, s+l-1]$. In other words, node G will only generate an R-alibi for a node i at time slot t on channel c if and only if 1) it receives *R-chain(i,s,l)* before time slot t and 2) the pair (c,t) exists in the chain *R-chain(i,s,l)*.

R-chain can drastically reduce the possibility of alibi-sharing behaviors of the attackers. Essentially, any two attackers j_1 and j_2 can share an R-alibi at time slot t on channel c only when (c,t) exists in both *R-chain(j_1,t_1,l)* and *R-chain(j_2,t_2,l)*. Thus, if all nodes (including attackers) are required to declare their *R-chains* before trying to obtain any R-alibis, the attackers cannot share alibis arbitrarily anymore. Unfortunately, if R-chain of an honest node is known by the attackers, the node is vulnerable to slander attacks. Basically, attackers can deterministically avoid jamming on channel c at time t if (c,t) is in the R-chains of victim nodes. Thus, victim nodes will not be able to get any R-alibis. To cope with the slander attacks, R-chains need to have certain randomness, which will be discussed next.

B. One-way R-chains

The basic idea to introduce randomness into an R-chain while still making it verifiable is based on the concept of one-way chains. One-way chains are widely used cryptographic primitive such as in Tesla [11]. A one-way chain is generated based on a one-way hash function F. To generate a one-way chain of length l, we first randomly pick the last element of the chain e_l. Then, we generate the whole chain by repeatedly applying the function F l times (i.e. $e_{l-1} = F(s_l)$, $e_{l-2} = F(e_{l-1})$ and so on). Finally, e_0 is the commitment to the entire one-way chain. e_0 can always be used to verify whether an element belongs to the chain i.e., any e_i belongs to a chain if and only if $F^i(s_i) = e_0$. The chain is released in the order from e_0 to e_l.

There are several key properties of one-way chain that will be used to solve our problem. First, each element e_i in the one-way chain can be considered as a random value uniformly drawn from the output space of one-way hash function F.

Second, once the first element of the chain e_0 is released to the network, any later element of the chain e_i ($i > 0$) cannot be changed and can be verified by checking whether $F^i(e_i) = e_0$. Third, due to the property of one-way hash function F, the knowledge of element e_i does not reveal any information about e_j for any $j > i$. Lastly, elements of a one-way chain have to be generated by applying the pre-selected one-way hash function F and cannot be generated arbitrarily.

In our alibi framework, a one-way chain is used to generate a one-way R-chain as follows. Time is divided into epochs of l time slots. An R-chain has a length of l. Each node generates its R-chain at the beginning of each epoch. To generate an R-chain of length l, a node i randomly selects a value s_l^i. The whole chain is then generated from s_l^i by repeatedly applying F in the same way to generate a one-way chain of length l.

For a node i, at a time slot t of an epoch ($1 \le t \le l$), it uses element s_t^i of the chain to calculate the channel c_t^i on which it will become an R-defendant at time slot t. Specifically,

$$c_t^i = s_t^i \mod (\lceil |\mathcal{C}|/p_i^{witness} \rceil) \tag{5}$$

, where $p_i^{witness}$ is the probability of being a R-defendant. So, at time slot t if $c_t^i \le |\mathcal{C}|$, node i becomes an R-defendant on channel c_t^i and does not become a witness otherwise. If the hash function is uniform, the probability of being a witness for node i is $\frac{|\mathcal{C}|}{(\lceil |\mathcal{C}|/p_i^{witness} \rceil)} \approx p_i^{witness}$. Note that the imprecision of the probability comes from the ceiling operation. However, the imprecision is bounded by

$$\frac{|\mathcal{C}|}{(|\mathcal{C}|/p_i^{witness})} - \frac{|\mathcal{C}|}{(|\mathcal{C}|/p_i^{witness}+1)} = \frac{|\mathcal{C}|}{(|\mathcal{C}|/p_i^{witness}) \times (|\mathcal{C}|/p_i^{witness}+1)} =$$

$$= \frac{p_i^{witness}}{(|\mathcal{C}|/p_i^{witness}+1)} < \frac{1}{|\mathcal{C}|+1}.$$

Thus, by using Eq. 5, we basically emulate the probability $p_i^{witness}$ of being witness of node i

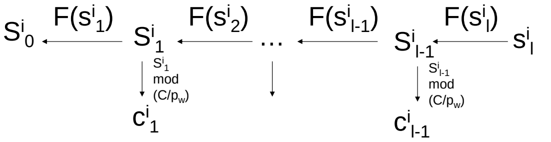


Fig. 4. An illustration of one-way R-chain

Figure 4 illustrates the whole process of generating an one-way R-chain of node i at the beginning of an epoch of l time slots. The chain $s_0^i, \dots, s_l^i$ is generated in advance. Each s_j^i ($j = 1..l$) is used to generate c_j^i - the channel that node i will hop to at time slot j in the epoch.

Similar to Section III-D, the content of a proof of node i at time slot t on channel c now has to include s_t^i. That means, the proof message m is $(t, c, \mathbf{s_0^i}, \mathbf{s_t^i}, H(PKT_t^c))$, where PKT_t^c is the received packet content at time t on channel c. With this scheme, any recipient (including G) of a proof message of node i can verify whether node i follows its one-way chain by checking whether $F^t(s_t^i) = s_0^i$ and $c_t^i = s_t^i \mod (|\mathcal{C}|/p_w)$. If either check failed, the proof message is invalid and will not be considered for generating alibi.

C. Analysis of one-way R-chains

As shown in the previous section, because $(s_t^i \mod (\lceil |\mathcal{C}|/p_w) \rceil)$ is uniform in $[0, \lceil |\mathcal{C}|/p_w \rceil]$, honest node i still becomes an R-defendant with probability $p_i^{witness}$ and behaves like in the non-colluding scheme.

For attackers, under one-way R-chains, collusion is limited. Specifically, two attackers i and j can collude at the overlaps of their R-chains (i.e. at any time slot t, where $c_t^i = c_t^j$). Thus, if attacker i jams at time t and attacker j becomes an R-defendant on the channel $c_t^j = c_t^i$ at time t, node j will get an R-alibi. Furthermore, if node j shares this alibi to node i, they can achieve both jamming and collecting alibis at the same time. That means such a pair of attackers that colludes in the way just described is undetectable under the one-way R-alibi scheme. We refer to this strategy as *safe-jam strategy*.

The success of the safe-jam strategy depends on the threshold k_{alibi} that the number of witnesses on the same channel at the same time slot has to be greater than to get an R-alibi. The maximum value of k_{alibi} is $(k+1)$ because there are k attackers. Smaller value of k_{alibi} will make both the honest nodes and the attackers to get alibis easier. We now will analyze the safe-jam strategy under the threshold k_{alibi}.

Parameter	Values
TDMA slot size	100ms
Number of channels	10
Number of nodes n	$[10-40]$
Number of CBR flows	$n/2$
Number of attackers	$[1-9]$
Jamming rate p^{jam}	$[0.1-1.0]$
γ	0.001
Simulation time	200 seconds

TABLE I

SIMULATION PARAMETERS

The probability that x attackers select the same particular channel is the binomial distribution with the probability of successful trial $q = 1/|\mathcal{C}|$,

$$P[\text{x attackers select the same channel}] = \binom{|\mathcal{C}|}{x} q^x (1-q)^{|\mathcal{C}|-x}$$

. If $x = k_{alibi}$, the selected channel can be jammed safely. Furthermore, if $x > k_{alibi}$, then the remaining $x - k_{alibi}$ attackers can safely jam any other channels because they already have the shared alibis from the k attackers that stay on the same channel. Therefore, given that $x \geq k_{alibi}$, the number of channels can be jammed safely is $x - k_{alibi} + 1$. Thus, the expected number of channels that can be jammed under the safe-jam strategy is

$$U_{safejam} = \sum_{x=k_{alibi}} P[\text{x attackers select the same channel}]$$

$$\times (x - k_{alibi} + 1) = \sum_{x=k_{alibi}}^{|\mathcal{J}|} \binom{|\mathcal{C}|}{x} q^x (1-q)^{|\mathcal{C}|-x} (x - k_{alibi} + 1) \tag{6}$$

.

Figure 5 shows $U_{safejam}$, calculated from Eq. 6, for the case of $|\mathcal{C}| = 11, 23$ and $k_{alibi} = 2$. The x-axis denotes $U_{safejam}$ - the expected number of channels that can be safely jammed. It is shown that $U_{safejam}$ is around 30% of the total number of channels.

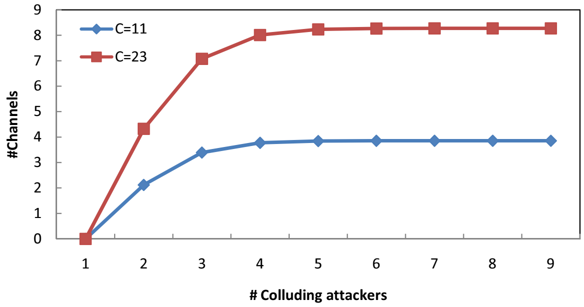


Fig. 5. Numerical results for the number of jammed channels $U_{safejam}$ under the safe-jam strategy

VI. EVALUATION

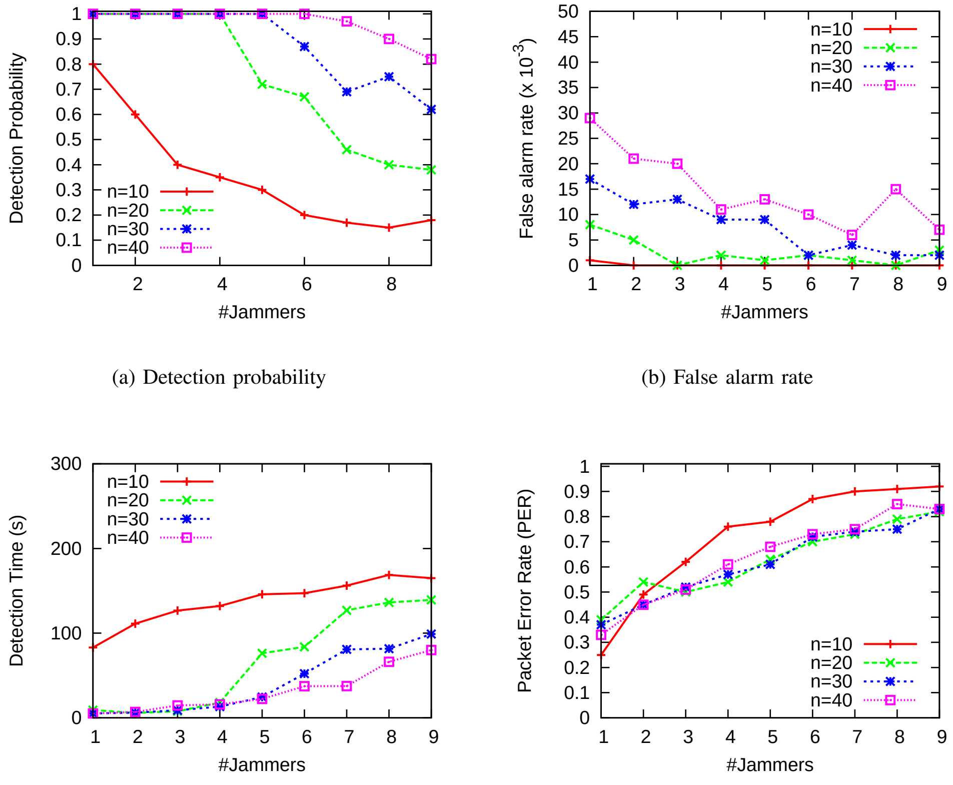
A. Simulation Setup

We evaluate the proposed protocols in TOSSIM. SSCH is implemented by modifying the existing MAC. Each node has 4 channel seeds (similar to the implementation in [3]). Each channel seed can be either "sending", "receiving" or "idle". A node only becomes a witness in the time slots where one of its idle channel seeds is used. There are $n/2$ CBR traffic flows established randomly and uniformly between pairs of nodes. In a CBR flow, the sender will send a data packet to the selected receiver in every $100ms$. In the simulation, all jammers use the same jamming probability p^{jam}. The simulation parameters are listed in Table I. In each scenario, we calculate the average detection probability, average false alarm rate, average detection time and packet error rate. We also repeat each scenario 10 times to get the confident statistics.

B. Simulation Results

Figures 6 and 7 show the performance of the proposed system for the case of non-colluding attackers. Specifically, Figure 6 shows the results in which the network size is varied from 10 to 40 and the jamming probability of all attackers are set to 0.6. Figure 6(a) shows that the detection probability increases when the network size increases. This is because more nodes with more traffic will give create more chances for honest nodes to get alibis. Figure 6(a) also shows that more attackers will make it harder to identify them, especially those with low jamming probability. This is because when there are more attackers, any attacker with low jamming probability can have more chances to get alibis from the other attackers' jamming actions. Figure 6(b) shows that the false alarm rate is maintained within the expected false alarm rate γ. Figure 6(c) further shows the detection time which can be similarly explained as in Figure6(a). Figure 6(d) shows the average packet loss. As the number of attackers increase, the packet loss rate also increases.

Figure 7 shows the results in which the network size n is set to 40 and the jamming probability p^{jam} is varied from 0.1 to 1.0. Figure 7(a) shows the detection probability of the proposed system. The detection probability increases when the jamming rate increases. This behavior shows the correctness of the principle of alibis: the more the attackers jam, the easier to detect them. It also shows that more attackers will make it harder to identify them, especially those with low jamming probability. This is because when there are more attackers, any attacker with low jamming probability can have more chances to get alibis from the other attackers' jamming actions. For attackers with high jamming probability, there will be no difference because they are always busy jamming. Figure 6(b) shows the correct false alarm rate according to the threshold γ. Figures 7(c) and 7(d) further show the detection time and the packet error rate, respectively.



(a) Detection probability

(b) False alarm rate

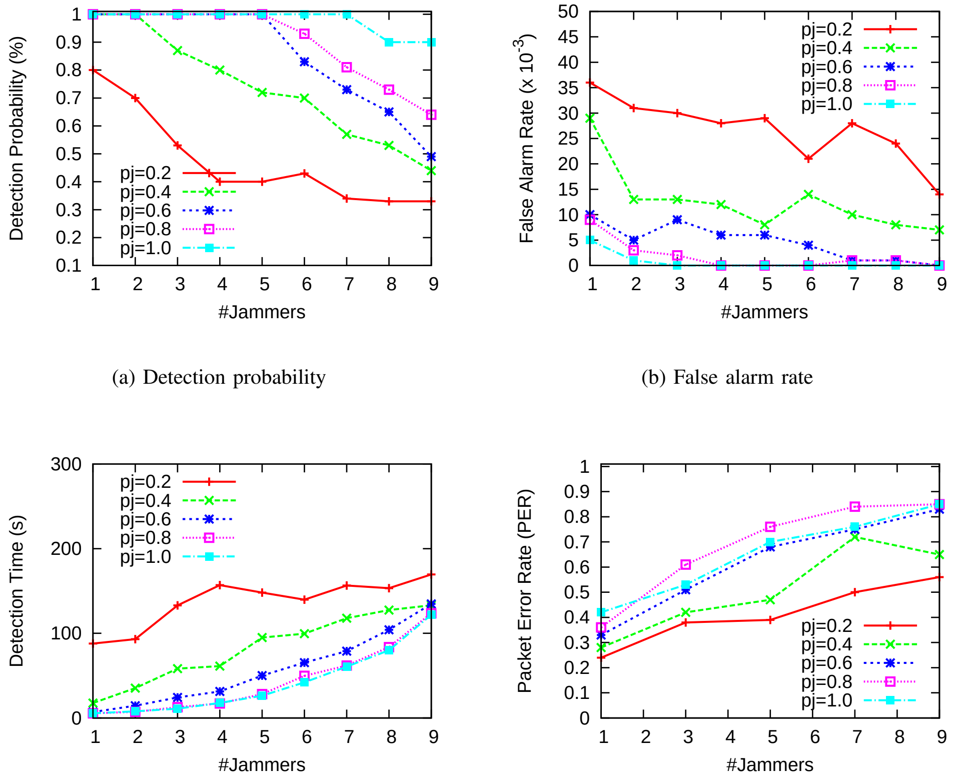
(c) Detection Time

(d) Packet error rate

Fig. 6. Non-colluding attackers: The impact of network size n ($p^{jam} = 1$).

VII. 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 (e.g. continuous jamming [8][12]). In reactive jamming strategy, the attackers only jam when they detect on-going communication on the targeted channels [12][13][14][15][16][17][18]. They may jam "important" packets such as



(a) Detection probability

(b) False alarm rate

(c) Detection Time

(d) Packet error rate

Fig. 7. Non-colluding attackers: The impact of jamming probability p^{jam} ($n = 40$).

control packets because corrupted control packets can drastically reduce the effective throughput of the communication channel [16][17].

Due to the dangers of various jamming attacks, jamming defenses have gained much attention from researchers. One of the most effective jamming mitigation is the spread spectrum technique. By hopping the carrier frequency (frequency-hopping spread spectrum - FHSS) or spreading its signal in time (direct-sequence spread spectrum - DSSS), the network can force the jammer to spend several-fold more power than if spread spectrum were not used [1]. However, the spread spectrum does not work if jammers know the hopping-pattern (HP) of FHSS or the pseudo-noise chip (PN) sequence of DSSS. Once the attacker knows such knowledge, he can jam the channel very effectively. For example, in 802.11 DSSS the PN is a common knowledge and the attacker can easily obtain it. By just using the COTS 802.11 cards, the attacker can easily modify the firmware to have an effective 802.11 jammer [12]. That said, the "outsider" attack (i.e., no knowledge of the HP or PN) can be defended effectively with spread spectrum technology while the "insider" attack is still a problem.

Indeed, dealing with the insider-based attacks, where the "shared secret" such as shared HP or PN is compromised, is a challenging problem. In [19], the authors proposed a tree of shared secrets of single shared secret to identify compromised nodes. This idea is an extension from the well-known hierarchical key management. Recently, there have been proposals of zero-shared knowledge communication under the jamming situation by using concurrent code [20], uncoordinated FHSS [10][21] and zero shared-secret DSSS [22][23].

VIII. Conclusion

We have shown how the alibi framework copes with the insider-based jammers. The framework is built from the core concept of "alibi", a form of defense whereby a defendant attempts to prove that he or she was elsewhere when the crime in question was committed. Even though started from such a simple concept, alibi framework has to deal with various challenging scenarios such as lossy channels, non-colluding multiple attackers and colluding multiple attackers. We have shown detailed study of properties of alibi framework including accuracy, detection time and network performance, by both simulation and analysis. The overall results show promising research direction of alibi framework to cope with insider-based jamming attacks.

REFERENCES

[1] R. Negi and A. Perrig, “Jamming analysis of MAC protocols,” Carnegie Mellon, Tech. Rep., 2003.

[2] H. Nguyen, T. Pongthawornkamol, and K. Nahrstedt, “Identifying insider-based jammers in single-hop wireless networks,” in *IEEE MILCOM*, 2009.

[3] P. Bahl and R. Chandra, “SSCH: Slotted seeded channel hopping for capacity improvement in IEEE 802.11 ad-hoc wireless networks,” in *ACM Mobicom*, 2004, pp. 216–230.

[4] J. Mo, H.-S. W. So, and J. Walrand, “Comparison of multi-channel MAC protocols,” in *ACM MSWIM*, 2009.

[5] A. Chan, X. Liu, G. Noubir, and B. Thapa, “Control channel jamming: Resilience and identification of traitors,” in *IEEE ISIT*, 2007.

[6] L. Lazos, S. Liu, and M. Krunz, “Mitigating control-channel jamming attacks in multi-channel ad hoc networks,” in *ACM WiSec*, 2009, pp. 169–180.

[7] K. Jamieson and H. Balakrishnan, “PPR: partial packet recovery for wireless networks,” in *ACM SIGCOMM*, 2007.

[8] A. D. Wood, J. A. Stankovic, and G. Zhou, “DEEJAM: defeating energy-efficient jamming in ieee 802.15.4-based wireless networks,” in *IEEE SECON*, 2007.

[9] A. D. Wood, J. A. Stankovic, and S. H. Son, “JAM: a jammed-area mapping service for sensor networks,” in *IEEE RTSS*, 2003.

[10] M. Strasser, C. Pöpper, S. Capkun, and M. Cagalj, “Jamming-resistant key establishment using uncoordinated frequency hopping,” in *IEEE Symposium on Security and Privacy (Oakland)*, 2008, pp. 64–78.

[11] A. Perrig, R. Canetti, J. D. Tygar, and D. Song, “The TESLA broadcast authentication protocol,” in *RSA CryptoBytes*, 2002.

[12] D. J. Thuente and M. Acharya, “Intelligent jamming in wireless networks with applications to 802.11b and other networks,” in *IEEE MILCOM*, 2006.

[13] 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, pp. 64–78.

[14] M. Li, I. Koutsopoulos, and R. Poovendran, “Optimal jamming attacks and network defense policies in wireless sensor networks,” in *IEEE INFOCOM*, 2007, pp. 1307–1315.

[15] V. Navda, A. Bohra, S. Ganguly, and D. Rubenstein, “Using channel hopping to increase 802.11 resilience to jamming attacks,” in *IEEE INFOCOM*, Anchorage, AK, May 2007.

[16] T. X. Brown, J. E. James, and A. Sethi, “Jamming and sensing of encrypted wireless ad hoc networks,” in *ACM MobiHoc*, 2006, pp. 120–130.

[17] P. Kyasanur and N. H. Vaidya, “Detection and handling of MAC layer misbehavior in wireless networks,” in *IEEE DSN*, 2003, p. 173.

[18] S. Radosavac, J. S. Baras, and I. Koutsopoulos, “A framework for MAC protocol misbehavior detection in wireless networks,” in *ACM WiSe*, 2005, pp. 33–42.

[19] J. T. Chiang and Y.-C. Hu, “Dynamic jamming mitigation for wireless broadcast networks,” in *IEEE INFOCOM*, 2008.

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

[21] T. Jin, G. Noubir, and B. Thapa, “Zero pre-shared secret key establishment in the presence of jammers,” in *ACM Mobihoc*, 2009.

[22] C. Popper, M. Strasser, and S. Capkun, “Jamming-resistant broadcast communication without shared key,” in *USENIX security Symposiump*, 2009.

[23] Y. Liu, P. Ning, H. Dai, and A. Liu, “Randomized differential DSSS: Jamming-resistant wireless broadcast communication,” in *IEEE INFOCOM*, 2010.