Mitigating Jamming Attacks in Wireless Broadcast Systems

Qi Dong
The University of Texas at Arlington
qi.dong@mavs.uta.edu
and
Donggang Liu
The University of Texas at Arlington
dliu@uta.edu
and
Matthew Wright
The University of Texas at Arlington
mwright@cse.uta.edu

Wireless communications are vulnerable to signal jamming attacks that can block mission-critical messages from being received. Spread spectrum techniques mitigate these attacks by spreading normal narrowband signals over a much wider band of frequencies and forcing jammers who do not know the spreading pattern to expend much more effort to launch the attack. In broadcast systems, however, jammers can easily find out the spread pattern by compromising just a single receiver. Several group-based approaches have been proposed to deal with *insider jammers* who compromise receivers in broadcast systems; they can tolerate up to t malicious receivers as long as the system can afford $2t$ additional copies for each broadcast message. In this paper, we introduce a novel jamming-resistant broadcast system that organizes receivers into multiple *channel-sharing broadcast groups* and isolates malicious receivers using adaptive re-grouping. By letting receivers in different groups partially share their channels, this scheme reduces the extra communication cost from $2t$ to $(2-\rho)t$ copies, where ρ is the channel sharing factor ($0<\rho<1$); this is much closer to optimal given the previously proven lower bound of t additional copies. In addition, we propose two extensions to the scheme to further improve the system performance: a *sequential test-based scheme* that saves communication costs without reducing security and an *unpredictable channel assignment scheme* that removes the requirement of operating on multiple channels at the same time, greatly reducing the required hardware complexity. We evaluate our schemes with analysis and simulation.

Categories and Subject Descriptors: C.2.0 [**Computer and Communication Networks**]: General—*Security and Protection*; C.2.2 [**Network Protocols**]: Applications

General Terms: Design, Security

Additional Key Words and Phrases: jamming attack, security, wireless networks

1. INTRODUCTION

In conventional wireless communication, the information-bearing baseband signal is modulated onto a proper high-frequency carrier for transmission. The modulated signal occupies a region of radio spectrum centered at the carrier frequency. If the attacker (or *jammer*) injects sufficient interfering signals into the same spectral region, he can significantly reduce the signal-to-noise ratio (SNR) at the receiver and thus interrupt the wireless communication [Poisel 2003].

Spread spectrum has long been an effective technique to mitigate jamming attacks. Examples include *Frequency Hopping* (FH) and *Direct Sequence Spread Spectrum* (DSSS) [Poisel 2003]. Their idea is to spread the signal over a much larger bandwidth to make it extremely expensive for an adversary to block the communication. In general, the greater the bandwidth, the better the resistance against jamming attacks. *Orthogonal Frequency Division Multiplexing* (OFDM) and *Software Defined Radio* were proposed for efficient spectrum management in cognitive radio networks [Akyildiz et al. 2006]. Researchers have studied the combination of spread spectrum and OFDM to improve the schemes' efficiency and flexibility in their use of the spectrum [Fazel and Kaiser 2008] as well as to improve their resistance against jamming [Cheun et al. 1999; Lance and Kaleh 1997; Zhou et al. 2002; Zhang and Li 2003].

These jamming-resistant techniques can be modeled as a virtual multi-channel communication system, where the *virtual (or logical) channel* denotes the signal coordinates determined by either the spreading code (DSSS), the frequency-hopping pattern (FH), the sub-carriers (OFDM), or their combinations. In such a system, the available spectrum contains a large number of orthogonal channels and only a small subset of them will be used for communication. If the jammer does not know which subset is in use, he will be forced to either jam a large number of channels with negligible interference in each or jam only a few of them and leave many others, very likely including those actually used for transmission, free of interference [Poisel 2003; Simon et al. 2001]. In case the jammer successfully blocks some of the signals, forward error correction (FEC) can be used to add redundancy and ensure robust transmission in most cases with low overhead [Luby 2002; Shokrollahi 2006; Karlof et al. 2004].

These mechanisms work well for one-to-one communication. However, in broadcast communication, there are many receivers. Once the attacker compromises a single receiver, he can discover which channels are in use and directly block those channels without wasting any effort. This *insider jammer* effectively leverages additional knowledge to render the defenses ineffective. The sender could bypass an insider jammer by using jamming-resistant one-to-one communication to send a separate copy of each message to each receiver. However, this introduces significant cost and delay, especially when a large amount of data (e.g., multimedia data) needs to be disseminated or the data is time-sensitive (e.g., in battlefields or other emergency situations).

Group-based schemes have been proposed to combat insider jammers in broadcast systems [Desmedt et al. 2001; Chiang and Hu 2008; Dong et al. 2008]. The idea is to organize receivers into multiple broadcast groups and use different channels for different groups. This ensures that a compromised receiver can only affect the members in the same group. A "divide and conquer" strategy is then used to isolate malicious receivers. However, these schemes require the sender to send a separate copy of each broadcast message to every group, causing a lot of communication overhead. The sender needs to send at least $2t$ extra copies of messages for each broadcast to deal with t compromised receivers.

We propose a solution based on partial channel sharing to reduce the extra communication overhead. Instead of sending messages using one channel [Desmedt et al. 2001; Chiang and Hu 2008; Dong et al. 2008], we divide such channel into multiple smaller ones and let different groups partially share their channels. In this way, the data sent over the shared channels can reach more than one group, saving substantial communication cost overall. The challenge with this approach is that a malicious receiver in one group may try to jam those shared channels to affect the message reception in other groups. Moreover, if the attackers can predict the channel assignment, they can protect themselves from being detected. In this paper, we develop an efficient solution to reduce the communication overhead without sacrificing resilience against malicious receivers.

The contributions of this paper are three-fold. First, we propose a novel jamming-resistant broadcast system that organizes receivers into multiple *channel-sharing broadcast groups* and isolates malicious receivers using adaptive re-grouping. Compared to existing approaches, this scheme reduces the communication cost from $2t$ to $(2-\rho)t$ extra messages, where t is the number of compromised receivers and ρ is the channel sharing factor ($0<\rho<1$). As a result, the proposed scheme is much closer to the optimal overhead of t additional messages (as proven in [Chiang and Hu 2008]) in realistic scenarios.

Second, we propose a *sequential test based scheme* to further improve the performance of our jamming-resistant broadcast system so that the sharing factor ρ can be set larger to save more communication cost without reducing security. Our analytic and simulation results show that this approach greatly pushes the performance limit of jamming-resistant broadcast systems towards optimal.

Finally, we also propose an *unpredictable channel assignment* technique. This technique dynamically forms broadcasting groups in an *unpredictable* way such that insiders cannot determine when they can attack without being detected. Basically, each receiver has a certain probability of sharing a common channel with other receivers. If a given channel is jammed and this channel is only assigned to one receiver, this receiver will be considered as one of the insiders. Thus, no matter when the insider chooses to jam the channel, there is a chance that he will be detected. Clearly, the more channels that the insider jams, the higher the probability that he will be detected. Compared with the previous two schemes, this technique only requires each receiver to listen to one channel, instead of multiple channels at the same time, thereby reducing the hardware cost greatly.

The rest of the paper is organized as follows. The next section presents the network and adversary models used in this paper. In Section 3, we describe our first jamming-resistant broadcast protocol and evaluate its performance. In Section 4, we discuss an improved approach using Lai's Bayes sequential tests. In Section 5, we present and evaluate our unpredictable channel assignment technique. Section 6 reviews related work on jamming-resistant communication. Section 7 concludes this paper and points out some future directions.

2. NETWORK AND ADVERSARY MODELS

Table I lists some frequently used notations. In this paper, we assume that a benign sender needs to broadcast messages to a set of r receivers, $\{R_1, R_2, \cdots, R_r\}$. We assume that the system contains totally n orthogonal *virtual channels*, which are

Table I. Frequently Used Notations

R	set of receivers, i.e., $R=\{R_1,R_2,\cdots,R_r\}$
R'	set of compromised receivers
C_G	set of broadcast channels assigned to group G
C'_G	set of jammed channels in C_G
t	number of compromised receivers, i.e., $t=\lvert R'\rvert$
m	number of broadcast channels assigned to each receiver
n	total number of available channels in the system
j	number of channels that can be jammed at one time
η	fraction of corrupted channels we can tolerate
ρ	channel sharing factor, i.e., fraction of shared channels

determined by spreading codes, frequency-hopping patterns, sub-carriers, or some combination of these. The proposed techniques in this paper can be considered as *group-based* approaches. In a group-based approach, we organize the receivers into multiple groups and assign m ($m \ll n$) channels to each group for broadcast communication. The receivers in the same group listen to and receive messages from the m channels assigned to the group. We assume that the sender and the receivers are time synchronized, as is necessary for any technique for creating virtual channels, and every receiver knows when to start and stop the transmission at any given channel.

Let C_G be the set of *active* channels assigned to group G ($|C_G| = m$). Every broadcast message M will be processed according to a packet-level forward error correction (FEC) code [Luby 2002; Shokrollahi 2006; Karlof et al. 2004] and divided into m packets such that M can be recovered from any set of $(1-\eta)\times m+1$ correct packets. These m packets will be transmitted on the active channels of every group, one for each channel. Thus, a receiver in group G can always recover the original message M unless $\eta \times m$ or more channels in C_G are jammed.

We assume that each receiver shares a secret key with the sender; this key is used to select the secret channels for the jamming-resistant one-to-one communication with the sender. Hence, this *private and jamming-resistant channel* allows a receiver to send feedback to the sender for making more informed decisions. In addition, we also assume that the sender can detect packet loss on every active channel. This can be achieved by: (1) having the sender monitor the active channels, (2) adding extra monitoring nodes in the network, or (3) asking receivers to report the channel condition.

The attacker's goal is to prevent as many receivers from receiving broadcast messages as possible. We assume that the attacker knows all technique specifics, including configuration parameters. We consider both *outsider attackers* and *insider attackers*. An outsider attacker does not know which channels are used. Thus he only randomly selects channels to jam. We assume that outsider attackers have power constraints that only allow them to block at most j channels at one time.

An insider attacker can compromise some receivers and learn all their secrets, including their assigned channels. In this paper, we focus on detecting an *active* malicious node, called the *traitor*, whose channel information is currently being used by the attacker to launch jamming attacks. For stealthy or selective attackers that behave normally most of the time, our goal is to catch them after they jam

Table II. Decision Making Criteria in TG examination.

Observation		Decision
E_0	$\lvert C'_{TG}\rvert < \eta m$	Accept H_0: $R' \cap TG = \emptyset$
E_1	$\lvert C'_{TG}\rvert \geq \eta m$	Accept H_1: $R' \cap TG \neq \emptyset$

the communication for more than a small number of times. We let t denote the total number of compromised receivers.

3. SCHEME I: ADAPTIVE RE-GROUPING WITH PARTIAL CHANNEL SHARING

In this section, we describe and analyze a scheme based on adaptive re-grouping with partial sharing of channels.

3.1 Protocol Description

For each broadcast message, the goals of jamming-resistant broadcast are: (i) the sender sends as few copies of this message as possible and (ii) the message can be correctly delivered to as many receivers as possible in the presence of insider jammers. The former indicates the need for a small number of groups. However, the latter requires small group size since a traitor can block the messages to all other receivers in the same group. This usually leads to a large number of groups. In Scheme I, we address this dilemma by proposing an *adaptive re-grouping* technique to isolate traitors without increasing the number of groups and a *partial channel sharing* idea to reduce the number of active channels needed to deliver broadcast messages.

Specifically, we classify the groups into *trusted groups* (TG) and *suspicious groups* (SG). There is only one trusted group but there can be multiple suspicious groups. The idea is to adaptively re-group the receivers if the attacker launches the jamming attack so that the benign nodes are more likely to be merged into the trusted group, and the traitors are more likely to be included in a number of small suspicious groups.

The high-level description of our protocol is given below. The trusted group includes all receivers *currently* believed (by our protocol) to be trustworthy. It is possible that the trusted group becomes untrusted later when more observations about the channel condition are available. Once this happens, we split the group into two suspicious groups (or a *suspicious group pair*) and set the trusted group to be empty. We let this pair of suspicious groups *partially* share their channels for lower communication cost, instead of using completely different set of channels. A group is said to be suspicious if we cannot *currently* determine whether it contains traitors; we need more observations about the channel condition to make the decision. Once we determine that one of the group pair contains traitors, we split it into two smaller suspicious groups for further processing. Meanwhile, if we can not determine the other group contains traitors at that time, we will believe it to be trustworthy and merge it into the trusted group. In other words, a suspicious group is merged into the trusted group only when its peer is determined to be untrustworthy. The above procedure continues until all traitors are isolated or the jamming attack stops.

Given the high-level protocol description, the remaining issues are: (1) how to

Table III. Decision Making Criteria in SGP examination.

	Observation		Decision
E_2	$\vert C'_{SG_1}\vert < \eta m$ and $\vert C'_{SG_2}\vert < \eta m$		Accept H_2: No traitors
E_3	$\vert C'_{SG_1}\vert \geq \eta m$ or $\vert C'_{SG_2}\vert \geq \eta m$	$\vert EC'_1\vert < \vert EC'_2\vert$	Accept H_3: SG_2 has traitors
E_4		$\vert EC'_1\vert = \vert EC'_2\vert$	Accept H_4Both have traitors
E_5		$\vert EC'_1\vert > \vert EC'_2\vert$	Accept H_5: SG_1 has traitors

detect the existence of traitors in the trusted group, (2) how to partially share channels between a suspicious group pair, (3) how to detect the untrustworthy group in a suspicious group pair, and (4) how to identify and remove traitors. We answer these problems in the following.

Detecting the existence of traitors in the trusted group: Initially, all receivers belong to the same trusted group TG; they share the same m channels. Note that if the trusted group contains no traitors, it is very difficult for the attacker to jam enough (at least ηm) channels to block the communication, as we will show in Section 3.2. Thus, if the number of jammed channels exceeds ηm, i.e., the receivers cannot recover the broadcast message, then it is very likely that the trusted group has traitors at this moment. Hence, we simply monitor whether $|C'_{TG}|$, i.e., the number of jammed channels in TG, exceeds ηm. If not, the group is still a trusted group; otherwise, we consider it as untrusted and immediately split it into two suspicious groups. (The trusted group becomes empty in this case.) Table II lists the criteria for making these decisions.

Group splitting with partial channel sharing: When a group is determined to be untrusted, we randomly split it into two equal-sized suspicious groups, i.e., a suspicious group pair. We focus on a given suspicious group pair SGP during the discussion. We let SG_1 and SG_2 denote the two groups of SGP. We assign m randomly selected channels to each of these two groups. Let C_{SG_1} and C_{SG_2} be the channels assigned to SG_1 and SG_2, respectively. In our scheme, SG_1 and SG_2 share a random fraction ρ of channels. Let SC be the shared common channels and EC_1 and EC_2 be the private (exclusive) channels for groups SG_1 and SG_2, respectively.

In each round of re-grouping, a receiver only receives its channel set. The sender will never leak any information about which channels belong to SC, EC_1, or EC_2. Clearly, if both groups in SGP have traitors, the attacker can find out the channel assignment and figure out all channel sharing information; this impact will be evaluated in Section 3.2.

Determining the untrusted group in SGP:

The goal here is to determine which group (SG_1, SG_2, or both) contains traitors. Let us consider an example when the sender notices that $|C'_{SG_1}| \geq \eta m$. There are three cases. First, when only SG_1 contains traitors, the attacker can easily select and jam ηm or more channels in C_{SG_1}. Second, when only SG_2 contains traitors, the attacker can jam some channels in C_{SG_2} expecting to jam some channels in SC and then randomly jam the channels in all other available channels expecting to jam the channels in EC_1. It is possible that $|C'_{SG_1}| = |SC'| + |EC'_1| \geq \eta m$. Third, both SG_1 and SG_2 contains traitors. In this case, the attacker knows the

Algorithm 1 ImmediateDetection

```
Require: $SPG = \{SG_1, SG_2\}$
Ensure: $H_3$, $H_4$, $H_5$
$H_3 \leftarrow F$, $H_4 \leftarrow F$, $H_5 \leftarrow F$
if $(|C'_{SG_1}| \geq \eta m) \vee (|C'_{SG_2}| \geq \eta m)$ then
    if $|EC'_1| < |EC'_2|$ then $\{E_3 = T\}$
        $H_3 \leftarrow H$
    else if $|EC'_1| = |EC'_2|$ then $\{E_4 = T\}$
        $H_4 \leftarrow H$
    else $\{E_5 = T\}$
        $H_5 \leftarrow H$
    end if
end if
```

channel assignment and can arbitrarily jam the channels in C_{SG_1}. An effective scheme is thus needed to distinguish these three cases given the observations about the channel condition.

Algorithm 2 Jamming-Resistant Broadcast

```
$TG \leftarrow R$
$C_{TG} \leftarrow$ Randomly select $m$ channels
$\{SGP\} \leftarrow \emptyset$, $\widehat{R'} \leftarrow \emptyset$
repeat
    Broadcast according to $C_{TG}$ and $\{C_{SGP}\}$ and monitor the jammed channels $C'$.
    if $TG \neq \emptyset$ then {begin $TG$ examination:}
        if $|C'_{TG}| \geq \eta m$ then $\{E_1 = T\}$
            $SplitGroup(TG)$\{Accept $H_1$\}
        else
            Assign new channels to replace the jammed channels.
        end if
    end if
    if $\{SGP\} \neq \emptyset$ then {begin SGP examination:}
        for $k = 0$ to $|\{SGP\}|$ do
            $ImmediateDetection(SGP_k)$
            if $H_4 = T$ then
                $SplitGroup(SG_{k,1})$, $SplitGroup(SG_{k,2})$
            else if $H_3 = T$ then
                $SplitGroup(SG_{k,2})$, $TG \leftarrow SG_{k,1} \cup TG$
            else if $H_5 = T$ then
                $SplitGroup(SG_{k,1})$, $TG \leftarrow SG_{k,2} \cup TG$
            else
                Assign new channels to replace the jammed ones.
            end if
        end for
    end if
until finish broadcast
```

Our scheme takes advantage of the fact that the channel assignment of a traitor-free group is always kept secret. In other words, the sender only sends C_{SG_1} to the

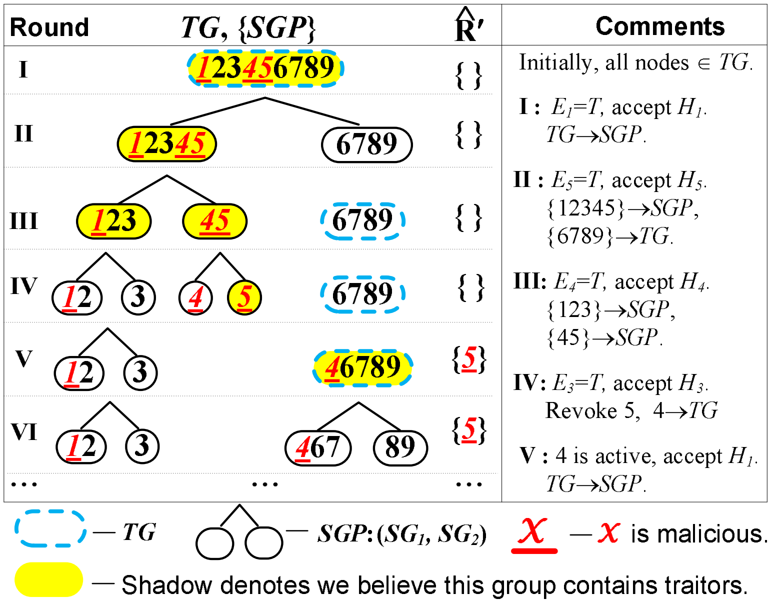
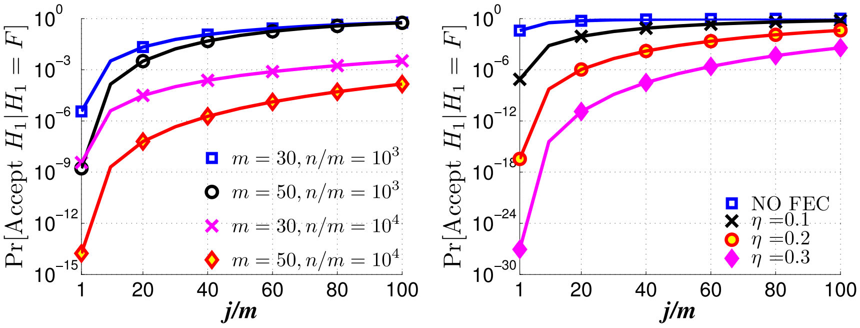

Round
TG, {SGP}
R̂′
Comments
I
123456789
{ }
II
12345
6789
{ }
III
123
45
6789
{ }
IV
12
3
4
5
6789
{ }
V
12
3
46789
{5}
VI
12
3
467
89
{5}
• • •
Initially, all nodes ∈ TG.
I : E1=T, accept H1. TG→SGP.
II : E5=T, accept H5. {12345}→SGP, {6789}→TG.
III: E4=T, accept H4. {123}→SGP, {45}→SGP.
IV: E3=T, accept H3. Revoke 5, 4→TG
V : 4 is active, accept H1. TG→SGP.
— TG
— SGP: (SG1, SG2)
X — X is malicious.
— Shadow denotes we believe this group contains traitors.

Fig. 1. Illustration of procedure 2.

receivers in SG_1, but does not leak anything about which channels belong to EC_1 or SC. Hence, we have the following observation: *if only one suspicious group in SGP has traitors, the probability that its private channels are jammed is usually higher than the probability that the private channels of the peer (traitor-free) group in SGP are jammed.* The reason for this is that the adversary does not know which channels are private to the peer group and can only jam a randomly selected subset of all the available channels. Therefore, we can determine which group has traitors by studying the jammed channels.

More specifically, if any receiver in the suspicious group pair SGP is blocked from receiving broadcast messages, we mark the group which has more jammed private channels as the one that contains traitors. Note that it is possible that both SG_1 and SG_2 contain traitors. In this case, the attacker can identify EC_1, EC_2 and SC and thereby jam exactly the same number of channels in both EC_1 and EC_2. This is, however, unlikely to happen in the case where at most one group contains the traitor. Therefore, if $|EC_1| = |EC_2|$, we mark both SG_1 and SG_2 as untrusted. Table III lists the decision criteria on determining untrusted groups. We provide a detailed analysis of this technique in Section 3.2.

Detecting and revoking the traitor: If a traitor keeps attacking the broadcast system, the size of its group will be reduced continuously. Once we detect a group with only one member that has the traitor, we can directly revoke this member from the system. Procedure 2 shows the pseudocode of our jamming-resistant broadcast scheme. Figure 1 shows an example of our jamming-resistant broadcast scheme.



(a) $\eta = 0.1$

(b) $m = 40$ and $n = 10^3 \times m$

Fig. 2. The decision error rate of traitor detection on a traitor-free TG.

3.2 Performance Analysis

There are two traitor detection modules in Scheme I, one for detecting the existence of traitors in TG and the other for detecting which group in SGP cannot be untrusted. Traitor detection is used to decide one of the hypotheses in Tables II and III based on the channel condition. In our analysis, we will focus on the *decision error rate* Pr[Accept $H|H = F$], where F represents FALSE. We will also discuss the impact of system parameters (m, N, η, and ρ) and the attacker-related parameter (j) on our protocol. We assume $\rho > \eta$.

3.2.1 *Performance of Traitor Detection for the Trusted Group.* **Pr[Accept $\boldsymbol{H_0}|\boldsymbol{H_0}$=$\boldsymbol{F}$]:** If $|C'_{TG}| < \eta m$, the nodes in TG can recover the broadcast message and the attacker does not block the communication. Although some receivers may have been compromised, there are no traitors actively involved in jamming the communication. It is therefore reasonable to believe that Pr[Accept $H_0|H_0 = F] = 0$. Nevertheless, we can always determine the existence of malicious receivers once the attacker starts the jamming attack.

Pr[Accept $\boldsymbol{H_1}|\boldsymbol{H_1}$=$\boldsymbol{F}$]: If TG contains no traitors, the attacker does not know which m channels belong to C_{TG}, he can only randomly select j channels from all n channels to jam. Thus, $|C'_{TG}|$, the number of TG's jammed channels, follows the hypergeometric distribution $f(i; n, m, j)$:

$$f(i; n, m, j) = \binom{m}{i}\binom{n-m}{j-i}\Big/\binom{n}{j}. \qquad (1)$$

Thus, Pr[Accept $H_1|H_1 = F$] can be calculated by

$$\Pr[\text{Accept } H_1|H_1 = F] = \sum_{i=\eta\times m}^{m} f(i; n, m, j) \qquad (2)$$

Figure 2 shows the decision error rate for traitor detection on a traitor-free TG under different settings. Parameter n is determined by factors like the design of spreading code, the channel hopping patterns, and the available spectrum in a specific channelization scheme. Intuitively, a larger n means more resistance against jamming attacks, as seen in Figure 2(a). In multi-carrier systems, m is often used for resource allocation. If a node requires more bandwidth, we can assign more channels. On the other hand, we can increase m to improve the resistance against jamming attacks as shown in Figure 2(a). η is determined by the forward error correction (FEC) scheme. Figure 2(b) shows that we can enhance the resistance against jamming by increasing η. Based on Figure 2, we choose $n/m = 10^3$, $j/m = 1 \sim 100$, and $\eta = 0.1, 0.2, 0.3$ in all our later analysis.

3.2.2 *Performance of Detecting the Untrusted Group in SGP.* $\mathbf{Pr[Accept}\ \boldsymbol{H_2}|\boldsymbol{H_2}\mathbf{=}\boldsymbol{F}\mathbf{]}$**:** Similar to the analysis in the last subsection, it is reasonable to believe that Pr[Accept $H_2|H_2 = F] = 0$ in this case, since the communication is not blocked.

$\mathbf{Pr[Accept}\ \boldsymbol{H_x}|\boldsymbol{H_x}\mathbf{=}\boldsymbol{F}\mathbf{]}|_{\boldsymbol{x}\mathbf{=3,4,\ or\ 5.}}$**:** Since our scheme has the same performance when $x = 3$ or $x = 5$, we only discuss the cases when $x = 3$ or $x = 4$. According to the definitions of the hypotheses in Table III, we have

$$\neg H_3 \Leftrightarrow (\neg H_4 \wedge H_5) \vee H_2$$
$$\neg H_4 \Leftrightarrow (\neg H_4 \wedge H_5) \vee (\neg H_4 \wedge H_3) \vee H_2.$$

For simplicity, we define the following four probabilities:

$$\begin{cases} P_1: & \text{Pr[Accept } H_3|\neg H_4 \wedge H_5 = T], \\ P_2: & \text{Pr[Accept } H_3|H_2 = T], \end{cases}$$
$$\begin{cases} P_3: & \text{Pr[Accept } H_4|\neg H_4 \wedge H_5 = T], \\ P_4: & \text{Pr[Accept } H_4|H_2 = T], \end{cases}$$

where T represents TRUE. Thus, we have

$$\text{Pr[Accept } H_3|H_3 = F] \leq \text{MAX}(P_1, P_2)$$
$$\text{Pr[Accept } H_4|H_4 = F] \leq \text{MAX}(P_3, P_4)$$

P_1 is the probability that event E_3 occurs, given the condition that only SG_1 contains traitors. In this case, the attacker only knows C_{SG_1} and cannot distinguish EC_1 from SC. To interrupt the communication at SG_2, the attacker may first jam a number of channels in C_{SG_1}, expecting to hit some channels in SC, and then jams $(j - |C'_{SG_1}|)$ channels randomly selected from the other available $(n - |C'_{SG_1}|)$ channels, expecting to hit some channels in EC_2. Event E_3 describes the case where (i) at least one of $|C'_{SG_1}|$ and $|C'_{SG_2}|$ exceed $\eta \times m$ and (ii) $|EC_1| < |EC_2|$. These limit the ranges of $|C'_{SG_1}|$, $|EC'_1|$, and $|EC'_2|$. Clearly, $|C'_{SG_1}| \neq m$ since otherwise $|EC'_1| = (1 - \rho) \times m$ and $|EC'_1| \geq |EC'_2|$. Thus, we have

$$0 \leq |C'_{SG_1}| \leq m - 1. \tag{3}$$

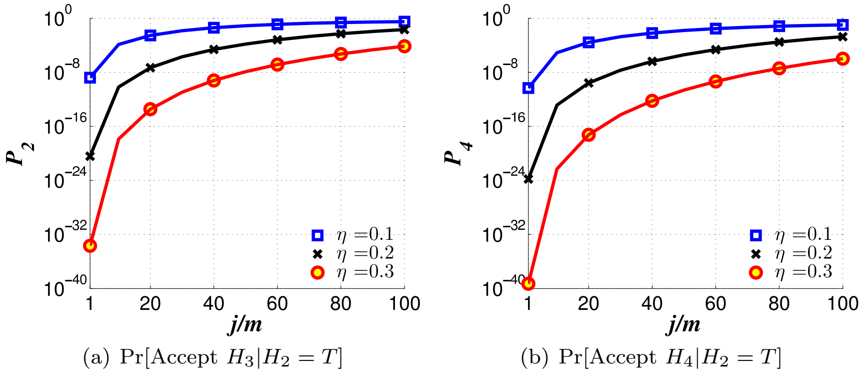
Consequently, we also have

$$0 \leq |EC'_1| \leq \text{MAX}(|C'_{SG_1}|, (1 - \rho)m - 1), \tag{4}$$

If $|C'_{SG_1}| < \eta m$, the attacker has to make $|EC'_2|$ greater than $\eta m - |SC'_1|$ to ensure that $|C'_{SG_2}| \geq \eta m$ in order to interrupt the communication at SG_2. Since

Table IV. Probabilities that the attacker succeeds in making particular events occur.

Pr[Event | Condition]			Parameters in equation 1.			
	Events	Conditions	N_t	N_d	N_s	i
P_1	$E_3 = T$	$\neg H_4 \wedge H_5 = T$	$\lvert C_{SG_1}\rvert = m$ $\lvert C^c_{SG_1}\rvert = n - m$	$\lvert EC_1\rvert = (1-\rho)m$ $\lvert EC_2\rvert = (1-\rho)m$	$\lvert C'_{SG_1}\rvert \in [0, m-1]$ $j - \lvert C'_{SG_1}\rvert$	See Inequality 4. See Inequality 5 and 6.
P_2		$H_2 = T$	$\lvert C\rvert = n$ $\lvert C_{SG_2}\rvert = m$ $\lvert {C'_{SG_2}}^c\rvert = n - \lvert C'_{SG_2}\rvert$	$\lvert C_{SG_2}\rvert = m$ $\lvert EC_2\rvert = (1-\rho)m$ $\lvert EC_1\rvert = (1-\rho)m$	j $\lvert C'_{SG_2}\rvert$ $j - \lvert C'_{SG_2}\rvert$	$\lvert C'_{SG_2}\rvert \in [\eta m, m]$ $\lvert EC'_2\rvert \in [1, \mathrm{MIN}(\lvert C'_{SG_2}\rvert, (1-\rho)m)]$ $\lvert EC'_1\rvert \in [0, \lvert EC'_2\rvert - 1]$
P_3	$E_4 = T$	$\neg H_4 \wedge H_5 = T$	$\lvert C_{SG_1}\rvert = m$ $\lvert C^c_{SG_1}\rvert = n - m$	$\lvert EC_1\rvert = (1-\rho)m$ $\lvert EC_2\rvert = (1-\rho)m$	$\lvert C'_{SG_1}\rvert \in [\eta m, m]$ $N_{s,2} \in [0, j - \lvert C'_{SG_1}\rvert]$	$\lvert EC'_1\rvert \in [0, \mathrm{MIN}(\lvert C'_{SG_1}\rvert, (1-\rho)m)]$ $\lvert EC'_2\rvert = \lvert EC'_1\rvert$
P_4		$H_2 = T$	$\lvert C\rvert = n$ $\lvert C_{SG_2}\rvert = m$ $\lvert {C'_{SG_2}}^c\rvert = n - \lvert C'_{SG_2}\rvert$	$\lvert C_{SG_2}\rvert = m$ $\lvert EC_2\rvert = (1-\rho)m$ $\lvert EC_1\rvert = (1-\rho)m$	j $\lvert C'_{SG_2}\rvert$ $j - \lvert C'_{SG_2}\rvert$	$\lvert C'_{SG_2}\rvert \in [\eta m, m]$ $\lvert EC_2\rvert \in [0, \mathrm{MIN}(\lvert C'_{SG_2}\rvert, (1-\rho)m)]$ $\lvert EC'_1\rvert = \lvert EC'_2\rvert$



(a) Pr[Accept $H_3|H_2 = T$]

(b) Pr[Accept $H_4|H_2 = T$]

Fig. 3. The decision error rate in traitor detection on a traitor-free *SGP* ($m = 50$, $n/m = 10^3$, and $\rho = 0.5$).

$|SC'_1| = |C'_{SG_1}| - |EC'_1|$, we have

$$|EC'_2| \geq \begin{cases} \text{MAX}(|EC'_1| + 1, \eta m - |C'_{SG_1}| + |EC'_1|) \\ \qquad\qquad\qquad\qquad \text{if } |C'_{SG_1}| < \eta m, \\ |EC'_1| + 1 \qquad\qquad\quad \text{if } |C'_{SG_1}| \geq \eta m. \end{cases} \tag{5}$$

The upper bound of $|EC'_2|$ is given by:

$$|EC'_2| \leq \text{MIN}((j - |C'_{SG_1}|), (1-\rho)m), \tag{6}$$

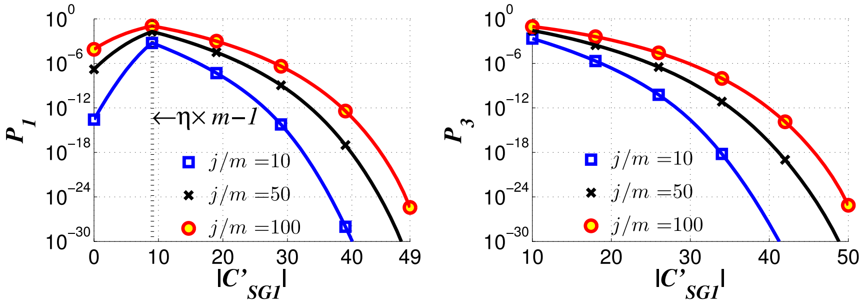
Therefore, P_1 can be estimated by:

$$P_1 = \sum_{|EC'_2|} \sum_{|EC'_1|} f(|EC'_1|; m, (1-\rho)m, |C'_{SG_1}|) \times f(|EC'_2|; n - m, (1-\rho)m, j - |C'_{SG_1}|),$$

where $f(i : N_t, N_d, N_s)$ is the hypergeometric distribution (i.e. Equation 1). The ranges of $|C'_{SG_1}|$, $|EC'_1|$, and $|EC'_2|$ are given by Inequalities 3,4, and 5, respectively. Similarly, we can calculate P_2, P_3, and P_4. The results are summarized in Table IV.

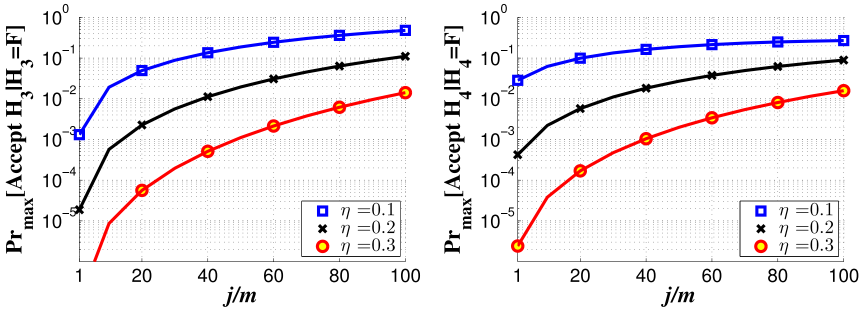
3.2.3 *Performance under Worst-Case Scenarios.* In this subsection, we analyze the worst-case performance of our protocol when the attacker takes the best possible strategy (i.e., he launches various types of jamming attacks to introduce the maximum number of wrong decisions). The analysis is conducted in the following three cases: (1) no group contains traitors; (2) only one suspicious group contains traitors; (3) both of the suspicious group pair contain traitors. We skip the case where the trusted group contains traitors, because once the attacker blocks the communication, we will notice the existence of traitors.

No group in SGP contains traitors: In this case, the attacker is not aware of any channel assignment, and can only jam randomly selected channels. His best strategy is to jam as many channels as he can. The attacker's energy limitations determine his jamming ability, i.e. the more channels he can block at the same time (or the greater the value of j), the higher probability he can interrupt the communication. This is shown in Figures 2 and 3.



(a) $\Pr[\text{Accept } H_3|\neg H_4 \wedge H_5 = T]$ (b) $\Pr[\text{Accept } H_4|\neg H_4 \wedge H_5 = T]$

Fig. 4. The decision error rate in traitor detection on SGP under different attack strategies ($m = 50$, $n/m = 10^3$, $\rho = 0.5$ and $\eta = 0.2$).



(a) $\text{MAX}(P_1, P_2)$. (b) $\text{MAX}(P_3, P_4)$.

Fig. 5. The decision error rate in traitor detection on SGP when only one group contains traitors ($m = 50$, $n/m = 10^3$, and $\rho = 0.5$).

Only one group in SGP contains traitors: Suppose only SG_1 has traitors and the attacker intends to fool us into believing that SG_2 contains traitors. Let us consider P_1 and P_3. In addition to j, there is another parameter that can affect the attacker's jamming performance, i.e., $|C'_{SG_1}|$. Unlike j, which is limited by the attacker's energy supply, $|C'_{SG_1}|$ can be controlled by the attacker.

Figure 4 shows the impact of $|C'_{SG_1}|$. From Figure 4(a), we know that P_1 reaches its maximum value $P_{max,1}$ when $|C'_{SG_1}| = \eta m - 1$. This means that the best jamming strategy is to jam $\eta m - 1$ channels in C_{SG_1} and spend the rest of his energy on jamming the channels randomly selected from $C^c_{SG_1}$. Similarly, from Figure 4(b), $P_3 \leq P_{max,3} = P_3|_{|C'_{SG_1}|=\eta m}$. Naturally, the insider jammer has more impact than the outsider attacker. Given the same configuration, we have $P_{max,1} > P_2$ and $P_{max,3} > P_4$, which can be seen from comparing Figure 3(a) with Figure 4(a) and comparing Figure 3(b) with Figure 4(b). As a result, $\Pr[\text{Accept } H_3|H_3{=}F] \leq P_{max,1}$ and $\Pr[\text{Accept } H_4|H_4{=}F] \leq P_{max,3}$. The worst-case performance is illustrated in Figure 5. From Figures 2, 3, and 5, we can see that the performance degrades with a larger j, but can be improved dramatically by increasing η.

Both groups of SGP contain traitors: In this case, the attacker knows the channel assignment and sharing information. However, this does not mean that the attacker can evade our protocol and block the broadcast messages arbitrarily. To prevent a victim R_i from receiving the broadcast message, the attacker has to jam at least ηm channels assigned to R_i, which means one of the events E_3, E_4, or E_5 must occur. Correspondingly, we will accept either H_3, H_4, or H_5, and split the groups to isolate traitors. Thus, the attacker's best strategy here is to "sacrifice" the traitors in one group to hide the traitors in the other group. Note that the hidden traitors will be assigned into the trusted group, and have the chance to launch another attack. However, we note that the hidden traitor will be detected and revoked at the moment when its knowledge is reused for launching jamming attacks in the future.

Overall impact of compromised receivers: In the following, we analyze how well this protocol performs under various types of jamming attacks including the selective attacks, where the insider jammers do not launch attacks continuously but choose to only attack at selected times. We consider the following questions: given t compromised receivers, how many benign receivers will lose the broadcast message in each round of jamming attacks, and how many rounds of jamming attacks can the attacker launch?

From Figure 1, we notice that the adaptive re-grouping leads to a tree-like structure. Each group that is believed to contain traitors is split into two suspicious groups, which can be considered its children. If a traitor R'_i continues to jam, its group will be split continuously until R'_i becomes the leaf of the tree, i.e., it becomes the only member of its group. In this case, if R'_i keeps acting maliciously, it will be removed from the system. Thus the attacker will keep R'_i inactive at certain point so that the sender has to keep sending messages to its group, wasting the sender's energy.

When we split an untrusted group into two suspicious groups, it is possible that both contain traitors but later we only detect one of them being untrustworthy. In other words, the attacker can hide the traitors in one of the groups by simply not using their secrets to jam the communication, and thus this group will be merged into the trusted group. The attacker may take advantage of this and make use of the secrets of its traitors one by one to maximize the impact of his attack.

Let R'_x denote the x-th active traitor whose secret is used to launch jamming attacks. In the beginning, the attacker keeps using R'_1's secret for jamming until R'_1 becomes a leaf, producing two suspicious groups and one trusted group. This introduces $\lceil \lg(|R|) \rceil$ rounds of attack. After that, the attacker will make use of the secret at another traitor R'_2 for jamming until R'_2 becomes a leaf, producing four suspicious groups and one trusted group. This introduces $\lceil \lg(|R|-2) \rceil$ rounds of attacks. This procedure continues until all traitors become leaves, producing $2t$ suspicious groups and one trusted group. Hence, R'_x can be used $\lceil \lg(|R|-2(x-1)) \rceil$ times.

Note that the number of receivers affected by the active traitor R'_x varies in each round of attack. This is because the size of R'_x's group is reduced by half after each round of attack. Therefore, the maximum numbers of receivers affected by R'_x in each round of attack are $\{|R|-2(x-1)-1, \lceil \frac{|R|-2(x-1)}{2} \rceil - 1, \lceil \frac{|R|-2(x-1)}{4} \rceil - 1, \cdots 1\}$. Overall, the maximum rounds of jamming attacks can be estimated by

$\sum_{x=1}^{t} \lceil \lg(|R| - 2(x-1)) \rceil$.

Communication overhead and impact of ρ:

Obviously, it is not possible to merge all the trusted receivers into a single group unless all the traitor nodes can be exactly pinpointed. However, if a malicious receiver becomes the single member in its own group, it may choose to hide himself and never attack again. There is no effective mechanism to detect such non-active attacker, and the sender has to send extra copies to guarantee that no legitimate receiver will lose the message. The existing solutions require $2t$ extra copies [Chiang and Hu 2008]. According to the previous analysis, our approach reduces such extra overhead to $(2-\rho)t$ copies. If $\rho = 0.5$, the communication overhead is $1.5t$ extra messages. Apparently, a larger ρ means fewer copies we need to send for each broadcast and thus less communication overhead. By properly configuring ρ, we can have much less communication cost than the previous methods.

However, we note that the value of ρ impacts P_1, P_2, P_3, and P_4. Thus, it also impacts the decision error rate. Figure 6 plots the maximal decision error rates on testing hypotheses H_3 and H_4 with different parameters.

We first look at $\Pr_{max}$[Accept $H_4|H_4{=}F$]. Based on the previous analysis, we know that

$$\Pr_{max}[\text{Accept } H_4 | H_4 = F] = f(0; m, (1-\rho)m, \eta m),$$

which apparently increases with ρ. Figure 6(b) also confirms that the decision error rate on H_4 increases with ρ.

We then look at $\Pr_{max}$[Accept $H_3|H_3{=}F$]. Suppose only group SG_1 has traitors. Then the attacker's best strategy is to jam $\eta m - 1$ channels in C_{SG_1} and as many channels in ${C_{SG_1}}^c$ as possible. If the attacker hits no channel in EC_1 and one channel in EC_2, he can fool us into believing that SG_1 is traitor-free but SG_2 is not. The probability of this happening can be estimated by

$$\Pr_{max}[\text{Accept } H_3 | H_3 = F] = \Pr[EC_1' = 0] \times \Pr[EC_2' > 1].$$

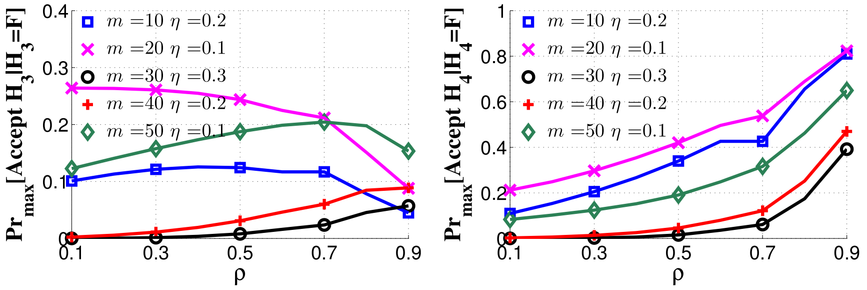
We note that this is not a monotonic function. The reason is that

$$\Pr[EC_1' = 0] = f(0; m, (1-\rho)m, \eta m - 1).$$

increases with ρ, but

$$\Pr[EC_2' \geq 1] = 1 - f(0; n-m, (1-\rho)m, j - \eta m + 1),$$

decreases with ρ. In addition, we also need to consider other parameters such as m, n, j, and η. Moreover, since $\Pr[EC_2' \geq 1]$ is usually small, the decision error rate on testing H_3 is still relatively low even when the decision error rate on testing H_4 with the same parameter set is very high, which can be seen from Figure 6(a) and Figure 6(b). From the above discussion, we believe that given a reasonable system configuration and reasonable limits on the attacker's jamming ability, our system can resist jamming attacks with very low decision error rates. However, we also note that if we have very limited resources (i.e., very small m, n and η) and strong attackers (i.e., a very large j), our system will generate a high decision error rate in order to save more cost in terms of communication. This is shown in Figure 6. In the next section, we will present a scheme to cope with resource



(a) Impact of ρ on testing H_3.

(b) Impact of ρ on testing H_4.

Fig. 6. The impact of ρ in resource-constraint systems ($n/m = 10^3$, and $j/m = 50$).

limitations and more powerful attackers and also to improve performance so that we can set a larger ρ to save more communication cost.

4. SCHEME II: SEQUENTIAL TEST BASED DETECTION

To reduce the decision error rate, we propose to get more observations about the channel condition before making any decisions. Generally, the more observations, the lower the decision error rate (and the larger ρ we can set). However, waiting for more observations increases the cost and the decision delay. For example, the sender will need to assign new channels to replace the jammed ones.

We propose to use a risk function to capture the impact of detection errors and delays. Without loss of generality, we consider a loss of 1 if the decision is wrong, and a loss of 0 if the decision is correct. We then introduce c to represent the ratio of the cost of waiting for one more observation over the cost of making a wrong decision. Thus, the problem becomes to find a solution to minimize the following risk function:

$$z = c \times E[S] + \Pr[\text{The decision is wrong}], \tag{7}$$

where variable S denotes the total number of channel observations we have collected at the moment when we stop the sequential test and make a decision.

4.1 Problem Statement

As discussed before, when we detect jamming attacks in SGP (i.e., $\mathrm{MAX}(|C'_{SG_1}|, |C'_{SG_2}|) \geq \eta m$), we know that at least one of them contains traitors. Note that if both groups contain traitors, we can always identify at least one untrusted group. The undetected group that also contains traitors can be simply handled in the future. As a result, in this section, we focus only on the case when only one of these two groups contain traitors.

The main problem is to tell which group is more likely to contain traitors given the observations about the channel condition. In the following, we will convert such traitor detection problem into an estimation problem of two random variables X and Y that are given by

$$X = \begin{cases} 1, & \text{if } |EC'_1| < |EC'_2|, \\ 0, & \text{if } |EC'_1| \geq |EC'_2|. \end{cases} Y = \begin{cases} 1, & \text{if } |EC'_1| > |EC'_2|, \\ 0, & \text{if } |EC'_1| \leq |EC'_2|. \end{cases}$$

Since only one group contains traitors, we can see that X and Y will both follow a Bernoulli distribution with a mean of P_x and P_y, respectively. Specifically, we have

$$\begin{cases} Pr[X=1] = P_x = 1 - Pr[X=0] \\ Pr[Y=1] = P_y = 1 - Pr[Y=0] \end{cases}$$

Let $P_{th} = \Pr_{max}[\text{Accept } H_3 | H_3 = F]$. We also have $P_{th} = \Pr_{max}[X = 1 | SG_2 \text{ has no traitors}]$. In other words, if SG_2 has no traitors, we have $P_x \leq P_{th}$. This implies:

$$P_x > P_{th} \Rightarrow H_3 : SG_2 \text{ has traitors.} \tag{8}$$

Similarly, we have

$$P_y < 1 - P_{th} \Rightarrow H_3 : SG_2 \text{ has traitors.} \tag{9}$$

$$P_y > P_{th} \Rightarrow H_4 : SG_1 \text{ has traitors.} \tag{10}$$

$$P_x < 1 - P_{th} \Rightarrow H_4 : SG_1 \text{ has traitors.} \tag{11}$$

From 8, 9, 10, and 11, we know that we can use P_x and P_y for detecting the group that contains traitors. We can then apply the Bayes sequential test described by Lai [1988] to address the problem.

4.2 Detection Based on Lai's Bayes Sequential Test

We do not use popular sequential test methods such as the fictitious optimal fixed sample size test or Wald's sequential probability ratio test since they require the knowledge of fixed P_x and P_y to achieve the optimal results. In our case, the attacker can arbitrarily change these values.

In the Bayes sequential test scheme [Lai 1988], we test the composite hypothesis: $H : P < P_0$ versus $K : P > P_0$. Lai shows that the risk z (i.e., Equation 7) of this sequential test is asymptotically equivalent to that of the fictitious optimal fixed sample size test that assumes the knowledge of P. The stopping rule of Lai's Bayes sequential test is

$$S = \inf\{s \geq 1 : I(\overline{P}_s, P_0) \geq \frac{h_0(c \times s)}{2c \times s^2}\},$$

where $\inf\{s\}$ denotes the infimum of set {s}; $I(\widetilde{P}, P)$ is the Kullback-Leibler information number given by $I(\widetilde{p}, p) = \widetilde{p} \log(\widetilde{p}/p) + (1 - \widetilde{p}) \log((1 - \widetilde{p})/(1 - p))$. When $\widetilde{p} = 0$ or 1, we define $I(\widetilde{p}, p) = \log 2$; $h_0(\cdot)$ is a function given in [Lai 1988], which is also listed below:

$$h_0(x) = \begin{cases} (2/\pi)^{1/2}(x^{-1/2} - 5x^{-5/2}/48\pi)/4, & \text{if } x \geq 0.8, \\ \exp(-0.69x - 1), & \text{if } 0.1 \leq x < 0.8, \\ 0.39 - 0.015x^{-1/2}, & \text{if } 0.01 \leq x < 0.1, \\ (t(2\log(1/x) + \log\log(1/x) & \\ - \log 4\pi - 3\exp(-0.016x^{-1/2})))^{1/2} & \text{if } x < 0.01. \end{cases}$$

The terminal decision rule (the final decision) is to accept H or K according to $\overline{P}_S > P_0$ or $\overline{P}_S < P_0$.

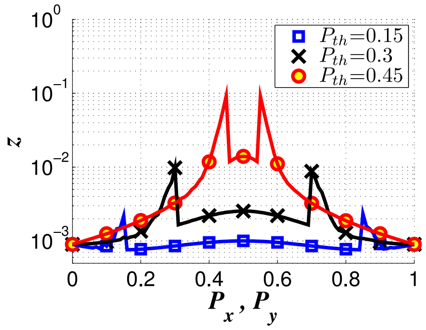


Fig. 7. The risk z v.s. P_x (or P_y), for $c = 10^{-4}$.

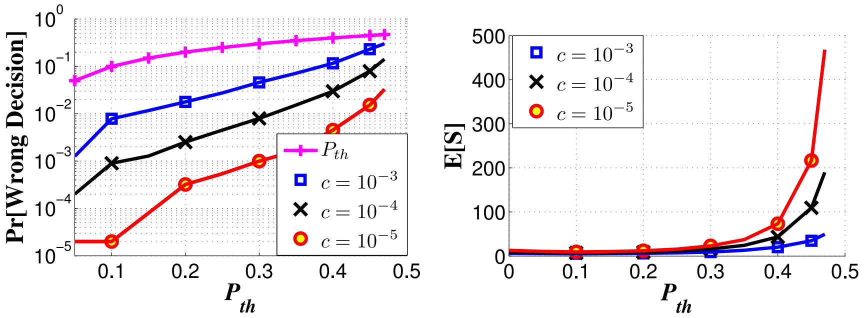
Algorithm 3 SequentialDetection

```
Require: $SPG = \{SG_1, SG_2\}$, $c$, $s$, $X$, $Y$, $P_{th}$
Ensure: $H_3$, $H_4$, $H_5$
$H_3 \leftarrow F$, $H_4 \leftarrow F$, $H_5 \leftarrow F$
if $(|C'_{SG_1}| \geq \eta m) \vee (|C'_{SG_2}| \geq \eta m)$ then
    $s \leftarrow s + 1$
    if $|EC'_1| < |EC'_2|$ then $\{E_3 = T\}$
        $X \leftarrow X + 1$
    else if $|EC'_1| > |EC'_2|$ then $\{E_5 = T\}$
        $Y \leftarrow Y + 1$
    end if
    if $(I(\frac{X}{s}, P_{th}) \geq \frac{h_0(c \times s)}{2c \times s^2}) \wedge (\frac{X}{s} > P_{th})$ then
        $H_3 \leftarrow T$
    else if $(I(\frac{X}{s}, 1 - P_{th}) \geq \frac{h_0(c \times s)}{2c \times s^2}) \wedge (\frac{X}{s} < 1 - P_{th})$ then
        $H_5 \leftarrow T$
    end if
    if $(I(\frac{Y}{s}, P_{th}) \geq \frac{h_0(c \times s)}{2c \times s^2}) \wedge (\frac{Y}{s} > P_{th})$ then
        $H_5 \leftarrow T$
    else if $(I(\frac{Y}{s}, 1 - P_{th}) \geq \frac{h_0(c \times s)}{2c \times s^2}) \wedge (\frac{Y}{s} < 1 - P_{th})$ then
        $H_3 \leftarrow T$
    end if
    if $(H_3 = H) \wedge (H_5 = H)$ then
        $H_4 \leftarrow T$
    end if
end if
```

Based on Lai's stopping rule and terminal decision rule, we propose our Bayes sequential test based detection in Procedure 3, which can be used to replace Line 15 in Procedure 2.

4.3 Performance Analysis

Procedure 3 requires two system parameters, c and P_{th}. We introduce c to balance the detection error and delay. P_{th} is a function of the system parameters (m, n, ρ, and η) and the attacker's jamming ability (j), as discussed in Section 3.2.3. Figure 6(a) shows that $P_{th} = \Pr_{max}[\text{Accept } H_3 | H_3 = F] < 0.5$ holds for a wide



(a) Wrong decision rate.

(b) Decision making speed.

Fig. 8. Performance of sequential test based detection in the worst-case scenario. (P_{th} is the probability of making a wrong decision of accepting H_3 using our first scheme in the worst-case scenario.)

range of parameters. Since the cost of making a wrong decision is very high, c is usually very small. We thus set $c = 10^{-4}$, as in [Lai 1988]. We then use a simple simulation to evaluate the performance of our scheme under different values of P_x and P_y, which can be controlled by the attacker. Specifically, for each value of P_x or P_y, we use MATLAB to generate 1000 values, which are either 0 or 1 under the given probability (P_x or P_y). The results in Figure 7 are based on 50,000 rounds of such simulation. It shows that our Bayes sequential test based approach can achieve a low risk no matter what strategy the attacker takes.

Figure 7 also shows that the best strategy for the attacker is to let $P_x = P_{th}$ or $P_y = P_{th}$. Figure 8 plots both of the expected number of lost messages before making a decision and the decision error rate. It shows that our scheme can achieve very good performance even in the worst case scenario where the attacker always takes the best strategy. It also shows that we can configure c according to the application to balance the decision delay and the error rate.

Figure 8(a) also plots P_{th}, the probability of making a wrong decision of accepting H_3 using our first scheme when the attacker takes the best jamming strategy. We can see that our detection scheme based on Bayes sequential test dramatically reduces the decision error rate with very small number of samples (less than 10 in most cases). Unlike the first scheme, the second scheme can achieve small decision error rates even if ρ is set to a large value, such as 0.9. As a result, this scheme enables the network operator to further reduce communication overhead by configuring ρ to be larger.

5. SCHEME III: UNPREDICTABLE CHANNEL ASSIGNMENT

The above techniques improve upon prior approaches using dynamic grouping [Chiang and Hu 2008; Dong et al. 2008] with partial channel sharing and reduce the extra communication cost from $2t$ to $(2 - \rho)t$ additional copies, where $(0<\rho<1)$. They are suitable for the situation where the receiver is able to process the signal from multiple channels simultaneously. In this section, we propose a lower-cost jamming defense for receivers with low-cost hardware that is incapable of listening to multiple channels at the same time. Specifically, we assign each receiver only one channel at a time, so that it can receive the completed messages by only monitoring that signal channel, instead of multiple channels simultaneously.

In this section, we also seek to prevent the attacker from predicting the channel assignments and achieving their maximal jamming impact without being detected. Specifically, in dynamic-group approaches, the sender isolates malicious receivers by splitting the suspicious groups. If the channel assigned to a two-member group is jammed, one of the group members must be malicious. However, we cannot determine which one is malicious. In previous approaches, the jammed two-member group will be split again and each member will get a new channel to receive messages. After splitting, the attacker will notice that he is the only user of the channel currently assigned to him. For example, in a network of 2^h (h is an integer) receivers. The attacker will realize he is the only member of his group if his channel has been changed h times. In this case, he will simply stop jamming to evade detection. Therefore, the attacker can go undetected and achieve his maximal impact by forcing the sender to send messages via $2t + 1$ channels.

In this section, we propose an improved technique based on *unpredictable channel assignment* to prevent insiders from knowing when to stop jamming. This scheme dynamically forms broadcasting groups in an *unpredictable* way such that insiders cannot achieve maximal jamming impact, but still keeps the extra communication cost as low as $(2 - \rho)t$ additional copies. The main change over the prior schemes relates to how we handle two-member groups. In other words, we apply the above dynamic-grouping technique at the beginning and switch to this approach when the group being jammed is a two-member group. We call the two nodes in a two-member group as a *suspicious node pair.*

In the following, we focus on one suspicious node pair to present and analyze our unpredictable channel assignment scheme. Specifically, we first discuss a basic approach and then, based the corresponding analysis, provide several enhancements.

5.1 The Basic Approach

The high-level idea of our approach is as follows. For each suspicious node pair, we reassign the channels periodically, such that the suspicious node pair has a certain probability to share the newly assigned channel. In our scheme, each receiver only knows which channel he should listen for broadcast messages but does not know whether that channel is only used by him or shared with the other receiver in the pair. In other words, the channel sharing information is unpredictable to the attacker. When the suspicious node pair shares the same channel, we can reduce the communication overhead by sending only one copy of each broadcast message, instead of two copies as required in previous approaches. When the two nodes use different channels, and the attacker launches a jamming attack, then the traitor can be detected since the jammed channel has only one user, i.e., the traitor.

5.1.1 *Protocol Description.* Figure 9 shows a flowchart of our proposed scheme. We now follow the flowchart and discuss our solution in detail.

Random Channel Assignment: The sender assigns the node pair new channels every q broadcast messages. For each channel assignment, the sender will assign these two nodes two different channels with probability p. In other words, they will share the same channel with probability $1 - p$ as shown in Figure 10.

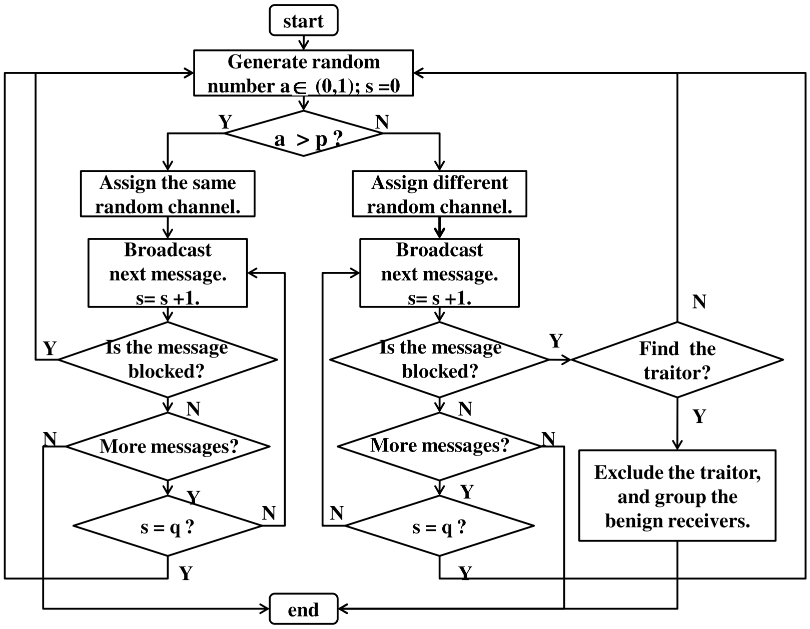

start
Generate random number a∈ (0,1); s =0
a > p ?
Y
N
Assign the same random channel.
Assign different random channel.
Broadcast next message. s= s +1.
Broadcast next message. s= s +1.
Is the message blocked?
Is the message blocked?
Find the traitor?
Exclude the traitor, and group the benign receivers.
More messages?
More messages?
s = q ?
s = q ?
Y
N
end

Fig. 9. The wireless broadcast system with insider jammer detection in Section 5.

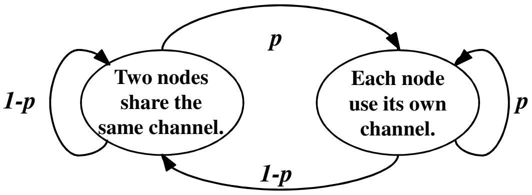

Two nodes share the same channel.
Each node use its own channel.
p
1-p
1-p
p

Fig. 10. The probability that each receiver is assigned different channel from each other is p. The probability that the suspicious node pair is assigned the same channel is $1 - p$.

The channels are always randomly selected from n available broadcast channels. The sender will update the new channel information to each receiver through their pairwise private channel. Every receiver only knows the channel that he should listen to for the broadcast messages; he does not know which channel is used by the other receiver or whether they are sharing the same channel. This creates a dilemma for the attacker: if he decides to jam the channel, he will be detected with some probability p each time; if he decides not to jam the channel, the sender's communication costs will be reduced due to channel sharing.

Message Broadcast and Channel Reassignment: After sending the new channel assignment information to each receiver in a suspicious node pair, the sender will maintain a counter s to track the number of broadcast messages that have been sent through that channel. If either of the following two conditions is met, the sender will reassign new channels to the node pair: (1) the sender has sent q messages through that channel, i.e., $s = q$; or (2) the message on the assigned channel is blocked due to jamming attack. Otherwise, the sender will continue to use the current channel to broadcast messages until no more broadcast messages exist in the queue, i.e., all broadcast messages have been delivered successfully.

Simple Traitor Detection: As shown in Figure 9, when a channel is blocked, the sender will need to determine which of the two nodes is the traitor. We will describe a simple but effective traitor detection scheme in this section and will discuss further enhancements to this scheme in Section5.2.

One simple traitor detection scheme is based on the following observations. If the attacker does not know which channel is used by a particular benign receiver, it is very hard for him to select and jam the right channel from a large number of potential channels. To increase his success rate, the attacker may simply jam the channel assigned to malicious receivers, expecting to affect the other benign receivers who are sharing that channels. However, if no one else is assigned to use a given channel, we can easily determine the traitor. Therefore, *if the sender assigns node u an unshared channel, and this unshared channel is blocked, then the sender will believe that u is malicious.*

Once the sender determines that one of the nodes in a suspicious node pair is malicious and blocks that node from the system, it will simply assume that the other node is benign. From then on, the sender will put this node into one of the groups that does not suffer from jamming attacks. Certainly, it is possible that the other node is an inactive malicious node, i.e., a compromised node that always behaves like a benign node. However, our approach can handle this very well. This is because once this node becomes active, it will be eventually isolated by using dynamic regrouping [Chiang and Hu 2008; Dong et al. 2008] and excluded from the system by using our scheme if the attacker chooses to maximize its jamming impact.

The above approach is also applicable to the other suspicious node pairs. We can directly apply this protocol on each of the suspicious node pairs. If the attacker keeps launching the jamming attack using the malicious nodes, we will eventually detect them. Otherwise, we can reduce the communication cost for each pair from two copies in the previous approaches to $1 + p$ copies, as we show in Section 5.1.2.

5.1.2 *Performance Analysis.* In the following, we study the performance of the basic approach in terms of communication overhead and security.

Communication Cost: We focus on the additional communication cost to tolerate t malicious receivers. Specifically, we calculate the number of extra copies (denoted by e) to be sent for each broadcast message.

The dynamic regrouping schemes [Chiang and Hu 2008; Dong et al. 2008] confine the malicious nodes into suspicious node pairs. Therefore, a clever attacker will try to maximize its impact by making the number of suspicious node pairs as large as possible. This can be achieved by having each suspicious pair contain only one malicious node. In this case, the total number of suspicious node pairs is t, as shown in [Chiang and Hu 2008; Dong et al. 2008]. In previous methods, those suspicious node pairs will eventually lead to $2 \times t$ single-member groups. Therefore, for each broadcast message, the sender needs to send $2 \times t$ extra copies of each broadcast message. On the contrary, our approach can reduce the extra communication cost, as shown in the following theorem.

THEOREM 1. *In the proposed approach, the expected number of extra copies to be sent for each broadcast message is no more than* $(1+p) \times t$*, i.e.,* $E[e] \leq (1+p) \times t$*, where* t *is the number of malicious receivers.*

PROOF. In our protocol, the sender will reassign new channels to each suspicious node pair every q broadcast messages. Note that two nodes will use different channels with probability p. As a result, for each message, we have

$$\begin{aligned} &\Pr[\text{The sender sends 2 copies to each pair}] = p, \text{ and} \\ &\Pr[\text{The sender sends 1 copy to each pair}] = 1 - p. \end{aligned}$$

Since t malicious receivers can introduce at most t suspicious node pairs, we have

$$E[c] \leq (2 \times p + 1 \times (1-p)) \times t = (1+p) \times t.$$

□

Therefore, the proposed approach can reduce the extra communication cost of the existing solutions by $(1-p)/2$. The lower the probability p, the smaller the number of extra copies need to be sent, and thus the more the energy we can save. In addition, it is shown in [Chiang and Hu 2008] that the lower bound of the number of extra copies is t in the worst case. Thus, this approach actually pushes limit of jamming-resistant broadcast towards optimal by setting p to a small value. Certainly, a smaller p may lead to a longer delay in detecting malicious insiders. We will revisit this issue later.

Besides the communication cost of broadcasting messages, the sender also needs to periodically update the channel information for each receiver. More specifically, if the attacker stops launching jamming attacks, the sender sends new channel information to both nodes in a suspicious node pair every q broadcast messages. If the attacker keeps jamming the channels before q messages are broadcast, the sender needs to reassign the channels even more frequently. However, as will be shown later, if only one of the suspicious node pair is malicious, the attacker can not launch many jamming attacks, otherwise the traitor is caught. On the other hand, if both of the nodes are malicious, the attacker is not able to use the traitors' channel to jam any other benign nodes. Certainly, the traitors may just try to waste the system's energy by jamming their own channels and forcing the sender to reassign the channel repeatedly. We will present an enhancement protocol to detect and stop this attack quickly in Section 5.2.2.

Detection Rate and Detection Speed: The attacker will try to maximize its jamming impact on the system by letting each suspicious node pair contain one malicious node. Therefore, in the following, we evaluate the performance of traitor detection in a suspicious node pair consisting of one malicious node and one benign node. We are interested in *the probability* P_d *that we can identify the active malicious node whose assigned channel information is used by the attacker to launch at most* x *jamming attacks.* x is used to evaluate the damage caused by the malicious node. In other words, x implies the speed of detection, and P_d denotes the probability that a traitor u can block x messages at its benign peer v.

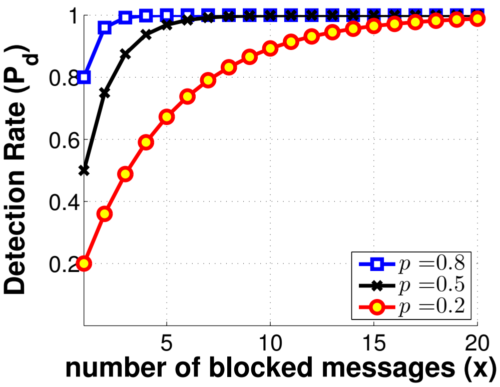


Fig. 11. The probability P_d that an insider jammer is detected v.s. x, the total number of messages it can blocked.

Theorem 2. *If a suspicious node pair consist of a benign node and a malicious node, then* $P_d = 1 - (1-p)^x$.

Proof. If the insider attacker jams the channel to block the broadcast messages at the benign node, he will be caught unless the traitor is assigned to the same channel as the benign node. According to the proposed protocol, the probability that the suspicious node pair share the same channel is $(1-p)$. Furthermore, the probability that the attacker succeeds in x rounds of attacks without being caught is $(1-p)^x$. Therefore, $P_d = 1 - (1-p)^x$. □

Figure 11 illustrates the relationship between P_d, p, and x. We can see that P_d increases with both p and x. Obviously, the more likely it is the two nodes in a suspicious pair use different channels (larger p), the more likely that the attacker will jam an unshared channel and be exposed as the traitor (larger P_d). In other words, a larger value of p implies a slower number of jamming attacks the jammer can launch before being caught. On the other hand, according to Theorem 1, a large p implies higher communication cost; thus, p should be small. Nevertheless, our scheme can actually detect traitors quickly even if p is small. According to Figure 11, we can see that the active traitor will be caught with very high probability before the 15th round of jamming, even if p is very small.

If both nodes in a suspicious node pair are malicious, the attacker can not jam the benign nodes. The only benefit is that He can keep the sender reassigning the channels frequently. We will make a slight modification of our protocol to address this problem in Section 5.2.2.

Please note we only analyze the performance of detecting the *active* traitor whose assigned channel is used by the attacker to block the broadcast message at the benign node. We do not try to detect the *passive* traitor, whose assigned channel is never used for jamming purpose. Indeed, there is no effective way to identify such malicious node as long as it behaves normally. However, once it becomes active, we can easily catch it.

False Alarm Rate: Theorem 2 shows that the proposed approach can identify the traitor with a very high probability even if the attacker only launches jamming attacks a few times. Thus, to avoid being caught, a cautious attacker may never jam the channels assigned to the malicious insiders. Instead, he will make blind guesses and randomly jam wireless channels, hoping to hit the channels used by some benign nodes and block their broadcast messages. He will succeed when the nodes in the suspicious pair are using different channels. In other words, if he happens to hit the channel used by the benign node in the pair, the sender will make a wrong decision and consider this benign node to be a traitor. Since every node will be assigned a new channel every q broadcast messages, we will study the false alarm rate P_f, i.e., the probability that a benign node is identified as a traitor by mistake during the period of transmitting q broadcast messages through its assigned channel.

Theorem 3. *The probability P_f of a benign node being identified as a traitor by mistake during the period of broadcasting q messages is*

$$P_f = p \times \sum_{i=1}^{q} P_{s,i}, \tag{12}$$

where $P_{s,i}$ is the probability that the unshared channel of the benign node is jammed for the first time at the i-th broadcast messages. $P_{s,i}$ can be estimated by

$$P_{s,i} = \frac{1 - \sum_{k=1}^{i-1} P_{s,k}}{\frac{n-t}{j} - (i-1)}, \tag{13}$$

where $P_{s,0} = 0$.

Proof. Let us calculate $P_{s,i}$ first. A clever attacker will avoid not only the t channels assigned to the t malicious receivers, but also the channels he has already jammed in the previous $i-1$ rounds. Given the attacker's ability of jamming j channels during the period of one message transmission, the probability that node u's unshared channel is jammed can be estimated by $\frac{j}{n-t-(i-1)\times j}$. On the other hand, the probability that u's unshared channel has not been jammed before is $1 - \sum_{k=1}^{i-1} P_{s,k}$. Thus, we have

$$P_{s,i} = \frac{j}{n-t-(i-1)\times j} \times \left(1 - \sum_{k=1}^{i-1} P_{s,k}\right).$$

Obviously, the probability that u's unshared channel is jammed during the period of broadcasting all q messages is $\sum_{i=1}^{q} P_{s,i}$. Moreover, the probability that u's channel is unshared is P. Therefore,

$$P_f = p \times \sum_{i=1}^{q} P_{s,i},$$

□

According to Equations 12 and 13, P_f is affected by n/j, t/j, p and q. In the following, we will discuss how these parameters affect P_f.

—Impact of n/j: This is the ratio between the number of total available channels in the system and the number of the channels the jammer can block in each round

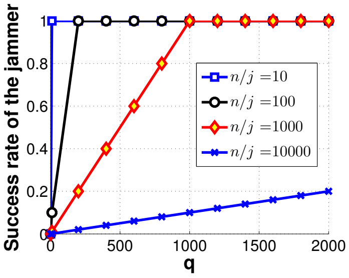


Fig. 12. The probability that the outsider attacker can successfully block the one-to-one communication system within the period of transmitting q messages.

of attack. As a special case in Equations 12 and 13, if $t = 0$, the attacker is an outsider attacker and does not compromise any receiver. In this case, receivers are all benign and share the same channel. This is equivalent to a *one-to-one* communication system. If p is set to be 1, P_f becomes the probability that the attacker can successfully jam a one-to-one communication channel within the period of transmitting q messages. Figure 12 shows the relationship between the probability that the jammer succeeds in blocking the one-to-one communication system and the possible jamming duration (q) with different values of n/j. Obviously, n should be large enough to combat the jammer. In practice, n is usually very large in any multi-channel based jamming-resistant communication system. Thus, in the following analysis, we will n/j to be 10^4.

—Impact of t/j: t is the number of compromised receivers in the system, and we usually have $n \gg t$. Thus, according to Equation 13, t/j will have much less impact on P_f than n/j. This is shown in Figure 13, which plots the relationship between the false alarm rate P_f and t/j. In the following analysis, we thus set t/j to be 1.

—Impact of p: p denotes the probability that the sender assigns each node in the node pair to a different channel. A smaller p implies a higher probability that the node pair uses the same channel. Since a cautious attacker does not want to expose the malicious node, he will not jam a channel assigned to any of the malicious nodes. For a given suspicious pair, the attacker needs to jam

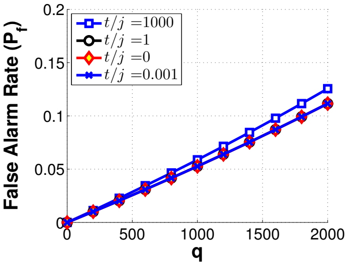


Fig. 13. The false alarm rate P_f, the probability that a benign node is identified as a traitor by mistake ($n/j = 10^4$ and $p = 0.5$).

the channel assigned only to the benign node to fool the sender into believing that this node is malicious. Clearly, the larger the value of p is, the higher the chance that the attacker can succeed is. According to Equation 12, the false alarm rate P_f increases along with p. This is also shown in Figure 14, which plots the relationship between P_f and p. Therefore, we should choose smaller p to get a lower false alarm rate P_f. As we showed earlier, we also choose a smaller p for lower communication cost e.

—Impact of q: q is the number of broadcast messages to be sent before the sender assigns a new channel. From Equations 12 and 13, and as shown in Figures 13 and 14, P_f increases with q. Thus, the sender needs to update the channels as frequently as possible. On the other hand, whenever the sender changes broadcast channels, it needs to send the new channel information to each receiver, which also increases the communication cost. Therefore, we need to make q as large as possible while still meeting the security requirement of a small P_f. In Section 5.2.1, we will present a modification to the scheme to guarantee a small P_f even if q is very large.

5.2 Enhancements

In this section, we will discuss some techniques to enhance the accuracy, reliability, scalability, and efficiency of the basic scheme presented in Section 5.1.

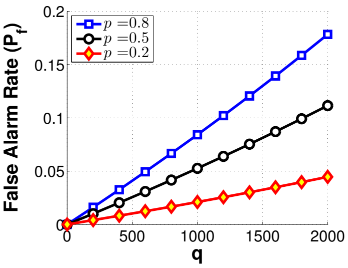


Fig. 14. The false alarm rate P_f v.s. q with different values of p ($j/t = 10^4$ and $j/t = 1$).

5.2.1 *Reducing The False Alarm Rate.* In the basic scheme, we monitor whether receiver u's unshared channel is jammed. Once this happens, u will be identified as a traitor. However, this decision could be wrong since the attacker may randomly jam wireless channels and happen to hit u's unshared channel. From our previous analysis, the false alarm rate will not be negligible if (1) the system has limited resources (i.e., a small number of n), (2) the attacker is very powerful (i.e., a large number of jammed channels j), or (3) the sender uses the same channel too many times (i.e, a large q). To reduce the false alarm rate, we propose to collect more observations before making a final decision. Thus, instead of making a decision based on a single jamming event, we monitor the frequency that the jamming event occurs.

Our intuition is that even if the attacker happens to jam benign node u's unshared channel once, he can hardly block u's unshared channel repeatedly in a short period of time. Note that we randomly reassign node u a new channel when u's channel is jammed. Thus, we have the following theorem.

Theorem 4. *The probability that the outsider attacker blocks u's channel for at*

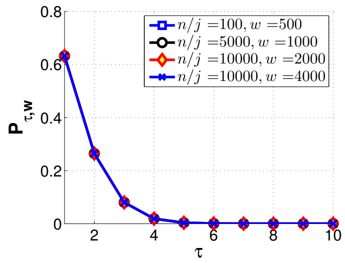


Fig. 15. The probability of getting at least τ successes in w trials ($j/t = 1$).

least τ times during w consecutive messages will not exceed $P_{\tau,w}$, which is given by

$$P_{\tau,w} = 1 - \sum_{i=0}^{\tau-1} \binom{w}{i} (P_J)^i (1-P_J)^{w-i}, \quad \textit{where}$$
$$P_J = \frac{j}{n - t - (q-1) \times j}.$$

Proof. According to Equation 13,

$$P_{s,i} < \frac{1}{\frac{n-t}{j} - (i-1)} < \frac{j}{n - t - (q-1) \times j}.$$

Thus, if node u is benign, the probability that its assigned channel is jammed during every broadcast message transmission should not exceed a threshold P_J, where

$$P_J = \frac{j}{n - t - (q-1) \times j}.$$

On the other hand, let us consider the random variable X, which follows the binomial distribution with parameters w and P_J, i.e., $X \sim B(w, P_J)$. The probability of getting at least τ successes in w trials is given by

$$P_{\tau,w} = 1 - \sum_{i=0}^{\tau-1} \binom{w}{i} (P_J)^i (1-P_J)^{w-i}.$$

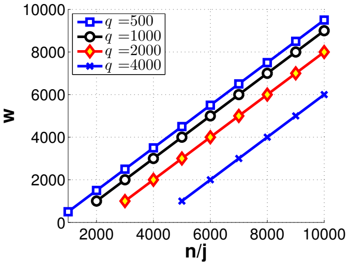


Fig. 16. The observation window w ($j/t = 1$).

Therefore, the probability that the outsider attacker blocks u's channel for at least τ times during w consecutive messages will not exceed $P_{\tau,w}$. $\square$

If we set $w = 1/P_J$, the expected number of times that an outsider attacker can block u's channel during w consecutive messages should be less than one. Therefore, the chance that the attacker blocks u's channel multiple times should be very small. In this case, $P_{\tau,w}$ can be estimated by

$$P_{\tau,w} = 1 - \sum_{i=0}^{\tau-1} \binom{w}{i} \left(\frac{1}{w}\right)^{i} \left(\frac{w-1}{w}\right)^{w-i}, \quad \text{where} \tag{14}$$

$$w = \frac{1}{P_J} = \frac{n-t}{j} - q + 1. \tag{15}$$

Let $w = 1/P_J$. Figure 15 and Figure 16 show $P_{\tau,w}$ under different system parameters. Figure 15 shows that $P_{\tau,w}$ decreases quickly with τ. For example, when $\tau = 5$, $P_{\tau,w}$ is as low as 0.0037, and it is very difficult for the outsider attacker to block u's channel for five times during w messages. Thus, if u's channel is jammed for five or more times during w messages, we can identify that u is malicious with a very high accuracy. Based on this observation, we modify the basic scheme as follows.

Protocol Modification: In the modified protocol, whenever node u's message is blocked, we reassign new random channels to u and its peer v and wait for more evidence. We set w according to Equation 15, and set τ to be a small number (e.g, five). We then keep track of how many times that u's unshared channel is jammed during the last w broadcast messages transmitted through u's unshared channel. If this number exceeds τ, u is considered to be malicious and will be removed from the system. As it is still possible though unlikely that node u could be a benign node, another reasonable option is to stop sending messages to u only for the next w messages. In other words, u will be temporarily removed from the system, and will be in the system again after w messages.

Performance Analysis: From previous analysis, the false alarm rate should not exceed $P_{\tau,w}$ given by Equation 14 and 15. Figure 15 also illustrates that $P_{\tau,w}$ is very small with a small value of τ, even if the available channels are limited or the attacker can jam many channels in one message transmission (i.e., a small value of n/j). This enhancement also allows us to reuse the same channel to transmit more messages than the basic scheme (i.e., a large q).

In addition, the proposed enhancement also limits the damage caused by malicious insiders. A compromised node u can only slightly effect its benign peer v, as shown in the following theorem.

Theorem 5. *Malicious node u can only cause its benign peer v to lose $\tau \times (1 - p)/p$ messages on average during every w broadcast messages.*

Proof. If u is malicious, the attacker is always aware of u's channel. To block the communication at its benign peer v efficiently, the attacker needs to jam the channel assigned to u so that he can succeed when u and v share the same channel. Let P_{attack} denote the probability that the attacker jams the channel assigned to u, $P_{unshared}$ denote the probability that u's unshared channel is jammed, and P_{shared} denote the probability that u's shared channel is jammed. We have $P_{unshared} = P_{attack} \times p$ and $P_{shared} = P_{attack} \times (1-p)$. Thus, $P_{shared} = P_{unshared} \times (1-p)/p$.

On the other hand, a careful attacker will not jam u's unshared channel τ times. Otherwise, the malicious node u will be identified. Therefore, the expected number of lost messages at v will not exceed $\tau \times (1-p)/p$. □

5.2.2 *Dealing with Other Attacks.* This section will discuss how to deal with the attacker whose goal is not to block the legitimate communications but to disrupt our protocol.

Attack Description: In our basic protocol, if a suspicious node pair's shared channel is jammed, the sender will simply reassign the new channels and continue to transmit the messages through the new channels, without any punishment. If the attacker puts two malicious nodes in the same pair, he will know whether those two malicious nodes are sharing their channel or not at any time. He may keep on jamming their shared channels and force the sender to reassign the channels frequently. Such an attack only blocks the malicious nodes' own channels and is not very interesting for the attacker who seeks to jam. However, the attacker may be interested in wasting the sender's energy by making him send new channel information continuously. We thus need to *monitor not only the jammed unshared channels but also the jammed shared channels.*

Intuition: In such an attack, the attacker will jam the shared channels with high frequency so that he can force the sender to send new channel information frequently. The attacker also needs to avoid jamming the unshared channels since otherwise the traitor will be detected. On the other hand, if only one of the suspicious node pair is malicious, the jammed channels will be very likely to include both unshared ones and shared ones. Also, among the channels assigned to a suspicious node pair, the ratio between the number of unshared channels and the number of shared channels is about $\frac{p}{1-p}$. Therefore, if the ratio between the number of jammed shared channels and the number of jammed unshared channels is much larger than $\frac{p}{1-p}$, it is very likely that both of the nodes in the suspicious node pair are compromised.

Protocol Modification: Our protocol should be robust and efficient in all of the following four situations: (i) M_0: neither of the node pair is malicious (i.e., the attacker is an outsider attacker); (ii) M_u: u is malicious; (iii) M_v: v is malicious; (iv) M_{uv}: both nodes in the node pair are malicious. (Please note that those four situations are not mutually exclusive. In fact, $\neg M_0 = M_u \cup M_v$ and $M_u \cap M_v = M_{uv}$.) Specifically, the outsider attacker (M_0) can not fool us into believing M_u, M_v, or M_{uv}. Only one compromised node (either $\neg M_{uv} \cap M_u$ or $\neg M_{uv} \cap M_v$) should not fool us into believing M_{uv}. To guarantee a low decision error rate, we propose to identify M_u, M_v or M_{uv} in the following two steps.

The fist step removes the possibility of M_0. Similar to our first enhancement technique, we make use of Theorem 4. Specifically, we keep tracking of the number $J_{s,w}$ of jammed shared channels of a suspicious node pair u and v in the most recent w (Equation 15) messages transmitted through their shared channels. If $J_{s,w} \geq \tau$ (τ is a small integer, e.g., five), at least one of the node pair is malicious. As shown before, it is very difficult for the outsider attack to fool us with this decision rule.

After we know at least one of u and v is malicious, the second step will be to determine M_u or M_v. Moreover, if both M_u and M_v turn out to be true, we know both u and v are malicious (M_{uv}). To identify M_u, let us analyze the situation where only node v is malicious (i.e., $\neg M_{u,v} \cap M_v$). Since we know at least one of u and v is malicious, if we observe an anomalous event that is impossible to occur when only v is malicious ($\neg M_{u,v} \cap M_v$), we can then conclude that u is malicious (M_u). To assist the decision making, we set a very small threshold ε. If the probability of a particular event happening is less than ε, we simply consider it as unlikely to occur. Therefore, ε serves as the maximum decision error rate that the system can tolerate.

Suppose only node v is malicious (i.e., $\neg M_{u,v} \cap M_v$). Let J_v denote the number of v's jammed unshared channels and J_s denote the number of jammed channels shared by u and v. J_s should follow the binomial distribution with parameters $J_v + J_s$ and $1 - P$, i.e., $J_s \sim B(J_v + J_s, 1 - P)$. The probability of getting exactly J_s shared channels is given by the probability mass function:

$$f(J_s; J_s + J_v, 1 - p) = \binom{J_s + J_v}{J_s} (1 - p)^{J_s} p^{J_v}.$$

Since our previous protocol will catch the single malicious node according to its jammed unshared channels, the traitor v will be careful and keep the number of its jammed unshared channels less than the threshold. However, the single malicious node does not know whether its channel is shared or not. Thus, it is quite difficult for it to jam many shared channels (i.e., large J_s) and only a few unshared channels (i.e., small J_v). In fact, when $\frac{J_s}{J_v} > \frac{1-p}{p}$, $f(J_s; J_s + J_v, 1-p)$ decreases with

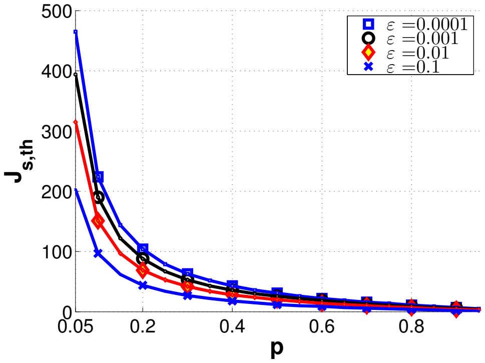


Fig. 17. The maximum number of shared channels that a malicious node pair can jam before being caught during the period, in which no more than 8 messages are jammed for w consecutive messages transmitted through the unshared channel.

increasing J_s. Based on this observation, we calculate $f(J_s; J_s + J_v, 1-p)$ whenever the shared channel is jammed. *If the probability* $f(J_s; J_s + J_v, 1-p) < \varepsilon$*, we believe it is unlikely that only* v *is malicious* ($\neg M_{u,v} \cap M_v$)*.* Since $J_{s,w} \geq \tau$ already shows that it impossible that both u and v are benign, we believe u *is malicious* (M_u)*.* Therefore, based on the above discussion, we have another criterion to identify malicious node u:

Theorem 6. *If (i)* $J_{s,w} \geq \tau$*, (ii)* $\frac{J_s}{J_v} > \frac{1-p}{p}$*, and (iii)* $f(J_s; J_s + J_v, 1-p) < \varepsilon$*,* u *is malicious. The false alarm rate of this decision rule is MIN*($P_{\tau,w}, \varepsilon$)*.*

Similarly, we also keep tracking of J_u, the number of u's jammed unshared channels. If $J_{s,w} \geq \tau$, $\frac{J_s}{J_u} > \frac{1-p}{p}$, and $f(J_s; J_s + J_u, 1-p) < \varepsilon$, we know that v is malicious (M_v). Furthermore, if both M_u and M_v are true, M_{uv} is true, i.e., both u and v are malicious.

In summary, the modification is simple. We just need to keep tracking of $J_{s,w}$, J_s, J_u, and J_v, and make decisions according to Theorem 6.

Performance Analysis: Since the attacker aims not to block the communication but to waste the sender's energy, we are interested in finding how many channels the insider attacker can jam before being caught, i.e., the number of channel assignment messages that a pair of malicious nodes can request from the sender. The jammed channels are either shared or unshared between the suspicious node pair. Let $J_{s,th}$ and $J_{un,th}$ denote the maximum number of shared and unshared channels that a malicious node pair can jam before being caught, respectively. Furthermore, $J_{un,th}$ includes $J_{u,th}$ and $J_{v,th}$, the maximum number of u's and v's jammed unshared channels before the traitor is caught, respectively. In other words, $J_{u,th} + 1$, $J_{v,th} + 1$, and $J_{s,th} + 1$ should be the minimum values that will trigger the traitor detection thresholds.

We seek to bound the maximum number of jammed unshared channels. Obviously, when u uses its unshared channel, v also uses its unshared channel; when u uses a shared channel, v also uses the exact same channel. If their unshared channels are jammed at the same time, the sender will reassign the new channel only once. To reach the maximal possible number of jammed unshared channel, a clever attacker should not jam both u's and v's unshared channel at the same time. Therefore, $J_{uv,th} = J_{u,th} + J_{v,th}$.

According to the traitor detection criterion in Section 5.2.1, neither J_u nor J_v should reach τ during every w consecutive messages transmitted through the unshared channels. Otherwise, either u or v will be identified as the traitor. Thus $J_{uv,th}$ should not exceed the threshold of $2 \times \tau - 2$ during every w consecutive messages transmitted through the unshared channels. In other words, $J_{uv,th}$ may keep increasing along with the total number of transmitted broadcast messages, but its maximum rate of growth is limited to $2 \times \tau - 2$ during every w consecutive messages transmitted through the unshared channels.

The detection criterion in this section mainly limits the number of jammed shared channels J_s. If J_s does not reach τ during w consecutive messages transmitted through the shared channels, it will not trigger the traitor detection mechanism. Such damage is very limited and an aggressive attacker may want J_s to exceed that threshold. However, he must be careful and keep $f(J_s; J_s + J_u, 1-p) \geq \varepsilon$ and $f(J_s; J_s + J_v, 1-p) \geq \varepsilon$. Otherwise, either malicious u or v will be identified. When $\frac{J_s}{J_u} > \frac{1-p}{p}$, $f(J_s; J_s + J_u, 1-p)$ decreases with J_s but increases with J_u. This implies that the more jammed unshared channels there are (either J_u or J_v), the more J_s the attacker can achieve. Thus, the attacker needs to increase the number of jammed unshared channel (J_{un}) to maximize J_s. We are looking for a value of $J_{s,th}$ that satisfies the following criterions:

$$J_{s,th} = \sup\{J_s : f(J_s; J_s + J_{uv,th}, 1-p) < \varepsilon\},$$

where $\sup\{J_s\}$ denotes the supremum of set $\{J_s\}$.

Similar to $J_{uv,th}$, $J_{s,th}$ may also increase along with the total number of transmitted broadcast messages. In other words, *if the attacker wants to jam more channels, he has to stop blocking the legitimate messages from time to time.* Obviously, the growth of $J_{s,th}$ is limited by the growth of $J_{uv,th}$.

Therefore, this upper bound is given by the following Theorem 7 and illustrated in Figure 17.

THEOREM 7. *The rate of growth of $J_{s,th}$ is limited by the rate of growth of $J_{uv,th}$. According to $J_{s,th} = \sup\{J_s : f(J_s; J_s + J_{uv,th}, 1-p) \geq \varepsilon\}$, where the growth of $J_{uv,th}$ is limited to $2 \times \tau - 2$ for w consecutive messages transmitted through the unshared channel.*

5.2.3 *Summary Of Unpredictable Channel Assignment.* In summary, the integrated traitor detection algorithm is given in Procedure 4, and its performance against different types of attackers is listed as follows:

—Outsider Attacker: In this case, neither u nor v is malicious. The probability that the attacker fools us into making a wrong decision is given by Theorem 4.

—Only u or v is malicious: Theorem 5 provides the expected number of jammed channels. The probability that the attacker fools us into believing both nodes are malicious is $\text{MIN}(P_{\tau,w}, \varepsilon)$, as shown by Theorem 6.

Algorithm 4 TraitorDetection

```
Require: suspicious node pair ($u$ and $v$), $p$, $\varepsilon$, $\tau$, $w$, $q$
Ensure: Identify the malicious nodes.
$J_s \leftarrow 0$, $J_u \leftarrow 0$, $J_v \leftarrow 0$, $NoTraitor \leftarrow$ TRUE
$J_{s,w} \leftarrow 0$, $J_{u,w} \leftarrow 0$, $J_{v,w} \leftarrow 0$
repeat
  $s \leftarrow 0$. Randomly assign new channel $C_u$ to $u$ and $C_v$ to $v$
  (Pr[$C_u = C_v$]$= p$).
  repeat
    For each broadcast message:
    $s \leftarrow s + 1$,
    if $C_u \neq C_v$ then
      Update $J_{u,w}$, $J_u$, $J_{v,w}$, and $J_v$
      if $J_{u,w} \geq \tau$ then
        $u$ is malicious! Stop procedure!
      end if
      if $J_{v,w} \geq \tau$ then
        $v$ is malicious! Stop procedure!
      end if
    else {$C_u = C_v$}
      Update $J_{s,w}$ and $J_s$
      if $J_{s,w} \geq \tau$ then
        $NoTraitor \leftarrow$ FALSE
      end if
      if $NoTraitor =$ FALSE then
        if $(f(J_s; J_s + J_u, 1-p) < \varepsilon) \wedge (\frac{J_s}{J_u} > \frac{1-p}{p})$ then
          $v$ is malicious! Stop procedure!
        end if
        if $(f(J_s; J_s + J_v, 1-p) < \varepsilon) \wedge (\frac{J_s}{J_v} > \frac{1-p}{p})$ then
          $u$ is malicious! Stop procedure!
        end if
      end if
    end if
  until $(s = q) \vee$(either $u$ or $v$ loses message)
until finish broadcast
```

—Both u and v are malicious: Theorem 7 provides the maximum number of channels that a malicious node pair can jam before being caught.

6. RELATED WORK

Jamming-resistant communication has long been an active research area [Poisel 2003; Xu et al. 2005; Lin and Noubir 2005; Li et al. 2007]. To provide jamming-resistant broadcast, several group-based schemes were studied [Desmedt et al. 2001; Chiang and Hu 2008; Dong et al. 2008]. In [Desmedt et al. 2001], each receiver is assigned to several groups and receives broadcast messages sent to these groups. This scheme requires knowledge of the number of compromised receivers before the group assignment. Two dynamic grouping schemes have been recently proposed [Chiang and Hu 2008; Dong et al. 2008]; they change the group membership dynamically using a "divide and conquer" strategy to counter jamming attacks.

However, given t compromised receivers, these two schemes require the sender to send $2t$ additional copies for each broadcast message to cope with t compromised receivers. Our scheme reduces such additional overhead to $(2-\rho)t$ copies, saving a fraction of $\rho/2$ communication cost. For example, when $\rho = 0.8$, we can save 40% communication cost.

In addition to jamming-resistant broadcast, researchers have studied jamming-resistant schemes to establish a secret key between two nodes or control channels among the nodes [Strasser et al. 2008; Strasser et al. 2009; Slater et al. 2009; Liu et al. 2010; Lazos et al. 2009; Chan et al. 2007; Tague et al. 2009]. These schemes usually involve a lot of communication overhead, long delay, or special hardware. They complement the scheme proposed in this paper, which focuses on how to isolate and detect compromised nodes in a broadcast system.

7. CONCLUSION AND FUTURE WORK

In this paper, we propose an adaptive jamming-resistant broadcast system with partial channel sharing. Compared to existing approaches, the proposed scheme significantly reduces the communication overhead without sacrificing security. Moreover, the sequential test based scheme allows us to further reduce the communication cost, greatly pushing the limit of jamming-resistant broadcast towards optimal. The unpredictable channel assignment scheme also provides an alternative solution for low-cost hardware communication platforms, and prevent the stealthy insider attackers from jamming without being detected.

In the future, we are particularly interested in developing our systems on real wireless communication platforms. It is also highly desirable to evaluate the performance of our approaches through field experiments to obtain more useful results. For example, the analysis on our jamming-resistant broadcast schemes assumes reliable communication if no jammer attacks the network. In other words, during the analysis, we believe that the packet loss is always caused by jammers. We believe that there exists insider jammers if a receiver cannot recover a message. Although this is often true in case of reliable communication, we may draw wrong conclusions in reality since the wireless communication is quite unreliable. The high channel loss may introduce a high decision error rate into our proposed approaches. It is thus important to conducting a thorough analysis on how channel loss rates impact our approaches.

We are considering the following two directions to further improve the unpredictable channel assignment. One is to regroup the suspicious nodes according to the jamming pattern. In this paper, we do not break a particular suspicious node pair until we identify the insider malicious node. However, even before we have enough evidence to identify the traitor, the existing jamming pattern may already give some clue about each node's likelihood of being malicious. Based on this likelihood, we can regroup all node pairs and rearrange them repeatedly, until the malicious nodes are isolated. Another potential improvement is to adaptively change the value of p, the probability that a pair of nodes use two different channels, according to the jamming pattern. This will make it even more difficult for the attacker to predict the channel assignment.

Acknowledgment:

This work is supported by the National Science Foundation (NSF) under grant number CNS-0916221. Any opinion, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

Akyildiz, I. F., Lee, W., Vuran, M. C., and Mohanty, S. 2006. Next generation/dynamic spectrum access/cognitive radio wireless networks: a survey. *Comput. Netw. 50,* 13, 2127–2159.

Chan, A., Liu, X., Noubir, G., and Thapa, B. 2007. Control channel jamming: Resilience and identification of traitors. *IEEE International Symposium on Information Theory. ISIT'07*.

Cheun, K., Choi, K., Lim, H., and Lee, K. 1999. Antijamming performance of a multicarrier direct-sequence spread-spectrum system. *Communications, IEEE Transactions on 47,* 12 (Dec), 1781–1784.

Chiang, J. and Hu, Y. 2008. Dynamic jamming mitigation for wireless broadcast networks. *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, 1211–1219.

Desmedt, Y., Safavi-Naini, R., Wang, H., Batten, L., Charnes, C., and Pieprzyk, J. 2001. Broadcast anti-jamming systems. *Comput. Netw. 35,* 2-3, 223–236.

Dong, Q., Liu, D., and Ning, P. 2008. Pre-authentication filters: Providing dos resistance for signature-based broadcast authentication in wireless sensor networks. In *Proceedings of ACM Conference on Wireless Network Security (WiSec)*.

Fazel, K. and Kaiser, S. 2008. *Multi-Carrier and Spread Spectrum Systems: From OFDM and MC-CDMA to LTE and WiMAX.* Wiley.

Karlof, C., Sastry, N., Li, Y., Perrig, A., and Tygar, J. 2004. Distillation codes and applications to dos resistant multicast authentication. In *Proc. 11th Network and Distributed Systems Security Symposium (NDSS)*.

Lai, T. 1988. Nearly optimal sequential tests of composite hypotheses. *The Ananals of Statistics 16,* 2, 856–886.

Lance, E. and Kaleh, G. 1997. A diversity scheme for a phase-coherent frequency-hopping spread-spectrum system. *Communications, IEEE Transactions on 45,* 9 (Sep), 1123–1129.

Lazos, L., Liu, S., and Krunz, M. 2009. Mitigating control-channel jamming attacks in multichannel ad hoc networks. *WiSec '09: Proceedings of the second ACM conference on Wireless network security*.

Li, M., Koutsopoulos, I., and Poovendran, R. 2007. Optimal jamming attacks and network defense policies in wireless sensor networks. In *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*. 1307–1315.

Lin, G. and Noubir, G. 2005. On link layer denial of service in data wireless lans. *Wirel. Commun. Mob. Comput. 5,* 3, 273–284.

Liu, Y., Ning, P., Dai, H., and Liu, A. 2010. Randomized differential dsss: Jamming-resistant wireless broadcast communication. In *Proceedings of the 29th IEEE International Conference on Computer Communications (INFOCOM '10)*.

Luby, M. 2002. Lt codes. In *FOCS '02: Proceedings of the 43rd Symposium on Foundations of Computer Science*. IEEE Computer Society, 271.

Poisel, R. 2003. *Modern Communications Jamming Principles and Techniques.* Artech House Publishers.

Shokrollahi, A. 2006. Raptor codes. *IEEE/ACM Trans. Netw. 14,* SI, 2551–2567.

Simon, M., Omura, J., Scholtz, R., and Levitt, B. 2001. *Spread spectrum communications handbook.* McGraw-Hill, Inc.

Slater, D., Tague, P., Poovendran, R., and Matt, B. 2009. A coding-theoretic approach for efficient message verification over insecure channels. In *WiSec '09: Proceedings of the second ACM conference on Wireless network security.* ACM, 151–160.

Strasser, M., Pöpper, C., Capkun, S., and Cagalj, M. 2008. Jamming-resistant key establishment using uncoordinated frequency hopping. In *SP '08: Proceedings of the 2008 IEEE Symposium on Security and Privacy (sp 2008).* IEEE Computer Society, 64–78.

STRASSER, M., PÖPPER, C., AND ČAPKUN, S. 2009. Efficient uncoordinated fhss anti-jamming communication. In *MobiHoc '09: Proceedings of the tenth ACM international symposium on Mobile ad hoc networking and computing*. ACM, 207–218.

TAGUE, P., LI, M., , AND POOVENDRAN, R. 2009. Mitigation of control channel jamming under node capture attacks. *IEEE Transactions on Mobile Computing*.

XU, W., TRAPPE, W., ZHANG, Y., AND WOOD, T. 2005. The feasibility of launching and detecting jamming attacks in wireless networks. In *Proceedings of ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*.

ZHANG, H. AND LI, Y. 2003. Anti-jamming property of clustered ofdm for dispersive channels. In *Military Communications Conference, 2003. MILCOM 2003. IEEE*. Vol. 1. 336–340 Vol.1.

ZHOU, S., GIANNAKIS, G., AND SWAMI, A. 2002. Digital multi-carrier spread spectrum versus direct sequence spread spectrum for resistance to jamming and multipath. *Communications, IEEE Transactions on 50,* 4 (Apr), 643–655.