Thwarting Inside Jamming Attacks on Wireless Broadcast Communications

Sisi Liu
Dept. of Electrical and Computer Engineering
University of Arizona
Tucson, AZ, USA
sisimm@ece.arizona.edu

Loukas Lazos
Dept. of Electrical and Computer Engineering
University of Arizona
Tucson, AZ, USA
llazos@ece.arizona.edu

Marwan Krunz
Dept. of Electrical and Computer Engineering
University of Arizona
Tucson, AZ, USA
krunz@ece.arizona.edu

ABSTRACT

We address the problem of jamming-resistant broadcast communications under an *internal threat model*. We propose a time-delayed broadcast scheme (TDBS), which implements the broadcast operation as a series of unicast transmissions, distributed in frequency and time. TDBS does not rely on commonly shared secrets, or the existence of jamming-immune control channels for coordinating broadcasts. Instead, each node follows a unique pseudo-noise (PN) frequency hopping sequence. Contrary to conventional PN sequences designed for multi-access systems, our sequences exhibit high correlation to enable broadcast. Moreover, their design limits the information leakage due to the exposure of a subset of sequences by compromised nodes. We map the problem of constructing such PN sequences to the 1-factorization problem for complete graphs. Our evaluation results show that TDBS can maintain broadcast communications in the presence of inside jammers.

Categories and Subject Descriptors

C.2.0 [**Computer - Communication Networks**]: General—*Security and Protection*

General Terms

Security, reliability, algorithms, design

Keywords

Jamming, broadcast communications, denial-of-service, wireless networks, graph factorization, security.

1. INTRODUCTION

Wireless communications are vulnerable to intentional interference attacks, typically referred to as jamming. In the simplest form of jamming, the adversary interferes with the signal reception by transmitting a continuous jamming waveform [24] or several short jamming pulses [18]. Conventional anti-jamming techniques rely extensively on spread spectrum (SS) communications, such as direct sequence spread spectrum (DSSS) and frequency hopping spread spectrum (FHSS) [1, 24]. SS provides bit-level protection by spreading bits according to a secret pseudo-random noise (PN) code, known only to the communicating parties. In the case of broadcast communications, the sender's PN code must be shared by all (potentially non-trustworthy) receivers. The disclosure of such a secret due to the compromise of any receiver nullifies the gains due to SS [16, 20].

Several researchers have studied the problem of anti-jamming broadcast communications under an internal threat model [4, 8, 9, 14, 16, 20, 21, 25, 26]. Methods in [4, 14, 16, 21] eliminate the dependency of broadcast on shared secrets. Baird et al. proposed the encoding of "indelible marks" at specific locations within each broadcasted message [4]. Assuming that an active jamming attack cannot flip a bit '1'to a bit '0', it was shown that a jammer cannot erase packets from the wireless channel (but can inject arbitrary packets). Pöpper et al. [20] proposed a method called Uncoordinated DSSS (UDSSS), in which broadcast transmissions are spread according to a PN code, randomly selected from a public codebook. At the receiving end, nodes decode received messages by exhaustively applying every PN code in the public codebook. Liu et. al. proposed RD-DSSS, a randomized differential DSSS scheme that also relies on randomly selected PN codes [16]. Compared to UDSSS, the RD-DSSS scheme provides resilience to reactive jammers.

Note that when the spreading PN code is not known a priori, broadcast transmissions must be repeated several times to synchronize the receiver [20]. Moreover, DSSS exhibits a threshold behavior to interference. It rejects the interfering signal as long as the interference remains below the jamming margin, but the throughput becomes practically zero if this margin is surpassed [19,24]. On the other hand, FHSS exhibits a graceful degradation in performance with the increase of interference. Due to this dual behavior, DSSS and FHSS find applications on different domains. The former is typical in the commercial domain (e.g., [12]) where moderate interference levels are caused by users operating on the same spectrum, while the latter finds applications in adversarial settings where the interference is likely caused by a powerful jammer. Because the adversarial model assumed in this work is of a powerful jammer, we develop anti-jamming methods that adopt a FHSS design.

Our Contributions: We study the problem of anti-jamming broadcast communications in the presence of inside jammers. We propose the *Time-Delayed Broadcast Scheme* (TDBS) for anti-jamming broadcast communications, based

WiSec'11, June14–17, 2011, Hamburg, Germany.

on FHSS communications. TDBS differs from classical FHSS designs in that two communicating nodes do not follow the same FH sequence, but are assigned unique ones that have high correlation properties. Unlike the typical broadcast operation where every receiver is eventually tuned to a common broadcast channel, TDBS implements the broadcast operation as a series of unicast transmissions spread both in frequency and time. To ensure resilience to inside jammers, the locations of the unicast transmissions, defined by a frequency band/slot pair, are only partially known to any subset of receivers. Because the jammer can only interfere with a limited set of frequency bands per time slot, a subset of the unicast transmissions are interference-free, thus propagating broadcast messages.

The problem of FH sequence design, is mapped to a 1-factorization problem in complete graphs. While a broad class of scheduling algorithms are known to employ 1-factors (perfect matchings) (e.g., [7,11,22,23,27]), they are, in general, concerned with unicast communications in a benign setting. They also typically require coordination via the exchange of broadcast messages [7, 11]). TDBS is specifically designed to facilitate broadcasting in the presence of jammers and in the absence of a coordination channel.

Note that TDBS is not meant to be a permanent replacement of the conventional broadcast mechanism in a benign setting. Broadcasting on a common frequency band achieves the optimal communication efficiency (one slot) in the absence of any jammer. TDBS is designed as an emergency mechanism for temporarily restoring communications until the jammer is physically removed from the network. Therefore, its primary focus is resilience to inside jammers.

Paper Organization: The remainder of the paper is organized as follows. In Section 2, we state the system and adversarial model assumptions. In Section 3 we present an overview of TDBS. Section 4 describes TDBS for single-hop networks. In Section 5, we extend the TDBS operation to multi-hop networks. The security and performance of TDBS are evaluated in Section 6. In Section 7, we present related work, and in Section 8, we conclude the paper.

2. SYSTEM AND ADVERSARIAL MODELS

Network Topologies: We consider two types of network topologies. In the topology of figure 1(a), a set of nodes form a single-hop broadcast group. Any node may initiate a broadcast transmission to its neighbors. This single-hop topology is typical in wireless LAN and wireless personal area networks, where a group of devices is assumed to be in close range (e.g., bluetooth devices), and in military scenarios where a set of mobile nodes move in a team-coordinated fashion. In figure 1(b), we consider a multi-hop network connected in ad hoc mode. To make TDBS scalable with the network size, we assume that the network is partitioned to clusters which form cliques [13,28]. Broadcast transmissions occurring under this architecture may be limited within a cluster, or may propagate to other clusters.

System Model: Nodes communicate over a set $\mathcal{C} = \{f_1, \ldots, f_K\}$ of K distinct frequency bands (e.g., $K = 79$ for the bluetooth standard). Each node is equipped with a single half-duplex transceiver. Hence, a node can only listen to or transmit over one band at a time. We assume that all nodes are synchronized to a time-slotted system. Nodes are capable of hopping between frequency bands. Without loss of generality, we assume that frequency hopping occurs on a per-slot basis. For simplicity, the duration of one time slot is assumed sufficient for the transmission of one message unit.

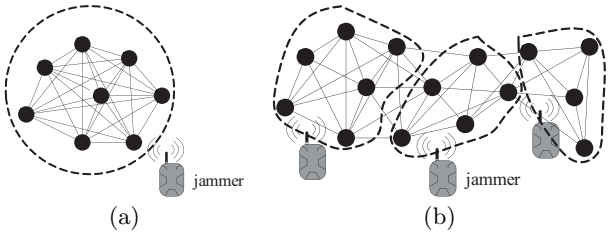


Figure 1: (a) A WPAN architecture in which devices located within one-hop form a broadcast communication group, (b) a multi-hop architecture in which communicating nodes span several hops.

The network is initialized by a trusted authority, which is responsible for pre-loading relevant parameters such as PN FH sequences and other cryptographic secrets. For multi-hop topologies, we assume a static network topology, known to the trusted authority. Broadcast communications can be either public (transmitted in an unencrypted form) or private. In the latter case, confidentiality and authenticity of the communication is achieved via resource-efficient public key operations. Once the network is initialized, the trusted authority is no longer needed.

Adversary Model: The goal of the adversary is to prevent the sender(s) from communicating with all, or a subset of the intended receivers. For this purpose, the adversary deploys a set of jamming devices at locations of his own choosing, which can be centrally coordinated. These devices are capable of collectively jamming any J frequency bands of the adversary's choosing, by adding interfering signals to the selected frequencies. Wireless transmissions over any of the jammed frequency bands are assumed to be "irrecoverably" corrupted. We do not impose any particular power constraint on the adversary, but assume that the jammed frequency bands become unavailable in the entire network (single-hop, or multi-hop). The jamming devices can switch between frequency bands on a per-slot basis.

The adversary is capable of physically compromising network devices and recovering stored information including cryptographic keys, PN codes, certificates, etc. Moreover, the adversary is aware of the methods used to protect broadcast transmissions (in our case the specifics of the PHY layer implementation and the TDBS algorithm). Note that similar adversary models have been considered in [14–16, 20].

3. OVERVIEW OF TDBS

To achieve jamming-resistant communications in the presence of insiders, TDBS realizes broadcast as a series of unicast transmissions distributed in frequency and time, thus avoiding the convergence of all nodes to a common frequency band. The locations of the unicast transmissions, defined by a frequency band/slot pair (f, s), are only partially known to each node (every node is aware of his own schedule). Therefore, the compromise of a node reveals only the set of locations assigned to that node, while keeping the locations of other communicating nodes secret.

For this purpose, nodes are divided into pairs scheduled to communicate over frequency bands which are selected at random. It is possible to partition the set of nodes to

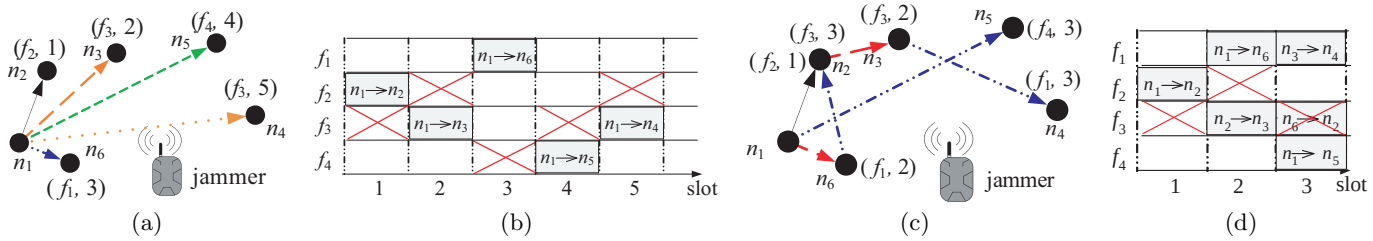


Figure 2: (a) Operation in the SU mode. Broadcast is realized as a series of unicasts. The pair (f, s) denotes the frequency band and time slot where the unicast takes place, (b) the timeline of the unicast transmissions of n_1 for the SU mode. The "x" marks denote frequencies jammed by the adversary, (c) operation in the AB mode. A broadcast transmission is relayed by several nodes at separate frequency bands, (d) the timeline of the unicast transmissions for the AB mode.

groups of size larger than two for more efficient broadcast communication at the expense of reduced resilience to node compromise. Because we are primarily concerned with the jamming-resistance property, we consider the case of node pairs. The communicating pairs and assigned frequency bands change on a per-slot basis thus realizing a FH system. TDBS differs from traditional FH designs in that: (a) communicating nodes do not synchronize to the same FH sequence, but follow unique hopping patterns and, (b) these patterns have a high correlation to lower the number of slots required to complete a broadcast transmission. Moreover, TDBS differs from rendezvous systems that have been proposed for coordinating multi-channel access (e.g. [3, 5]), in that it focuses on the broadcast operation as opposed to rendezvous for unicast communications.

Two modes of operation are proposed for TDBS: the sequential unicast mode (SU) and the assisted broadcast mode (AB). In the SU mode, the sender sequentially relays information to intended receivers. This more inefficient mode is appropriate when receivers do not have relaying capabilities, or are not trusted to relay the broadcast message. In the AB mode, any node that receives a broadcast message can act as a relay for that message.

Figure 2 shows an example of the two modes. In figure 2(a), node n_1 operates in the SU mode. It sequentially unicasts a broadcast message to nodes $n_2 - n_6$. Figure 2(b), depicts the timeline of transmissions of figure 2(a). The broadcast is completed after five slots. The "x" marks denote the frequency band jammed by the adversary at each time slot. Figure 2(c), shows the operation in the AB mode. Node n_1 initiates a broadcast in slot 1, by transmitting a message m to n_2. In slot 2, n_1 and n_2 relay m to n_6 and n_3, respectively, using frequency bands f_1 and f_3 in parallel. In slot 3, the broadcast is completed with the relay of m from n_1, n_3 and n_6 to n_5, n_4 and n_2, respectively. The timeline of the transmissions taking place in the AB mode is shown in figure 2(d). Observe that in this scenario the broadcast is completed despite the jamming of the transmission between n_6 and n_2 in slot 3.

The main challenge of TDBS is to design the FH sequences of individual nodes such that the following requirements are met: (a) hopping sequences are pseudo-random, (b) compromise of a subset of nodes (insiders) limits the information leakage relevant to the sequences of uncompromised nodes, and (c) every node has the same opportunity to perform a broadcast (fairness). In the next section, we develop algorithms for constructing hopping sequences for TDBS-SU and TDBS-AB that satisfy the above requirements. We first illustrate our algorithms for single-hop topologies and then extend our results to multi-hop topologies.

4. TDBS FOR SINGLE-HOP TOPOLOGIES

To achieve resilience to jamming, we randomly distribute unicast transmissions both in frequency and in time. This problem can be viewed as a link scheduling problem for avoiding collisions in multi-channel networks, under the node-exclusive interference model. A large body of literature treats this type of scheduling as various instances of a matching problem in general graphs [7, 11, 22, 23, 27]. However, pre-existing methods are not immediately applicable to our setup for the following reasons.

In link scheduling problems, the goal is to maximize the aggregate network throughput, realized as the sum of individual traffic flows. We are concerned with the dissemination of one message to a specified set of receivers (the members of a broadcast group) over unpredictable frequency band/slot locations, and in the presence of adversaries. This desired property is not necessarily satisfied by maximum throughput designs, which optimally schedule link transmissions in the entire network (centralized approaches) [27]. Moreover, decentralized solutions implementing distributed matching algorithms require the local exchange of coordination messages between nodes, over a commonly agreed channel [7, 11]. Clearly, such a channel cannot be available in our setup due to the presence of an inside jammer.

To ensure the broadcast property, we map the problem of constructing FH sequences to the problem of producing 1-factorizations in complete graphs. 1-factorizations realize a series of perfect matchings (1-factors), which span the all edges of a complete graph [30]. Hence, a broadcast from any node will be communicated to all other nodes. We first present relevant preliminaries from graph theory. Interested readers are referred to [17,30] for an in-depth treatise of the problem of 1-factorization.

4.1 Definitions and Useful Theorems

Consider a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ denotes the vertex set and $\mathcal{E}$ denotes the edge set. Assume that $|\mathcal{V}| = 2n$ where n is a positive integer (a dummy node can be added otherwise).

DEFINITION 1. **Complete graph:** *$\mathcal{G}(\mathcal{V}, \mathcal{E})$ is said to be complete if each pair of vertices is connected by an edge. We denote such a graph by K_{2n}, where $|\mathcal{V}| = 2n$.*

DEFINITION 2. **1-factor:** *A 1-factor or a perfect matching F of a graph $\mathcal{G}$ is a subset of $\mathcal{E}$ that partitions $\mathcal{V}$, i.e., F*

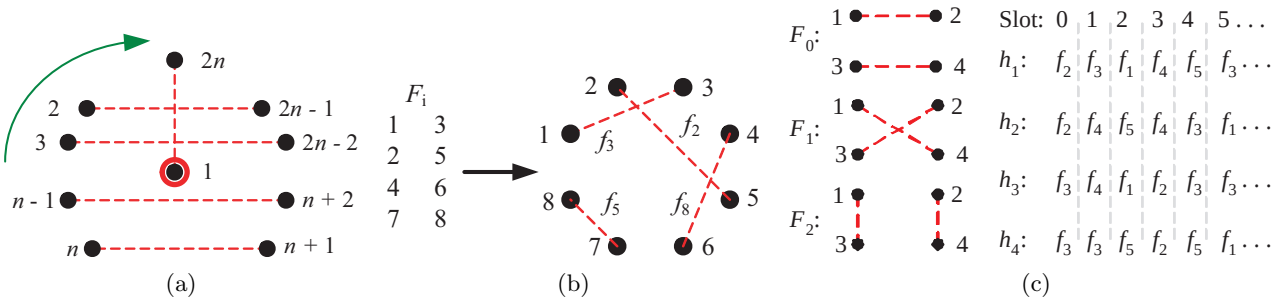


Figure 3: (a) Algorithm for constructing a 1-factorization $\mathcal{F} = \{F_0, \ldots, F_{2n-2}\}$. To obtain a factor F_i, every node is rotated by i positions to the left. Node 1 remains fixed, (b) mapping of a 1-factor to unicast transmissions. Paired nodes concurrently communicate on separate frequency bands, (c) construction of hopping sequences for sequential unicast based on 1-factorization for a group of four nodes.

is a set of pairwise disjoint edges of $\mathcal{G}$ that covers all vertices of $\mathcal{V}$.

DEFINITION 3. **1-factorization:** *A 1-factorization $\mathcal{F}_{2n} = \{F_0, F_1, \ldots, F_{2n-2}\}$ of a graph $\mathcal{G}$ is a partition of its edge set $\mathcal{E}$ to $(2n-1)$ 1-factors.*

THEOREM 1. **1-factorization of K_{2n}:** *A complete graph K_{2n} is 1-factorable [30].*

Construction of 1-factorizations of K_{2n}: 1-factorizations of K_{2n} can be systematically constructed using well-known algorithms (e.g., [10,17,29,30]). These methods typically rely on the selection of a "starter" 1-factor, based on which the entire 1-factorization can be derived. A simple method for constructing a 1-factorization of K_{2n} is to select a starter 1-factor and apply a shift-and-rotate operation to it [30]. This method is illustrated in figure 3(a). A 1-factorization is initialized by the 1-factor F_0. Node 1 remains fixed. To obtain the 1-factor F_i, nodes in the perimeter are rotated clockwise by i steps.

4.2 Mapping to the 1-factorization Problem

In this section, we map the problem of constructing hopping sequences for TDBS into the problem of generating 1-factorizations of complete graphs. In our mapping, the vertex set $\mathcal{V}$ of K_{2n} represents the node set $\mathcal{N}$ of the single-hop network, and an edge $(x, y) \in \mathcal{E}$ represents a unicast transmission between nodes x and y. A 1-factor corresponds to partitioning of the $2n$ nodes into n communicating pairs. These pairs are scheduled to communicate in parallel over separate frequency bands. A 1-factorization $\mathcal{F}_{2n}$ partitions the set of edges $\mathcal{E}$ into $(2n-1)$ disjoint 1-factors, where each edge appears exactly once. In a schedule constructed according to $\mathcal{F}_{2n}$, every node has the opportunity to communicate with all remaining $(2n-1)$ nodes, thus achieving the sequential relay of a broadcast message.

An example of the mapping to the 1-factorization problem is shown in figure 3(b). A group of eight nodes is partitioned into four pairs, which are scheduled to communicate over four frequency bands. According to the 1-factor F_i, the communicating pairs during slot i are $\{(1,3),(2,5),(4,6),(7,8)\}$, communicating over frequency bands f_3, f_2, f_8 and f_5, respectively. Figure 3(c) shows a feasible set of hopping sequences h_j for four nodes, $j = 1, \ldots, 4$, based on the 1-factorization of K_4. Communication of all pairs of nodes is completed in three slots. We now present algorithms for constructing FH sequences.

Algorithm 1 TDBS-SU: Sequential Unicast Mode

```
1: Generate F_2n of K_2n
2: repeat
3: for i = 0 to (2n − 2) do
4:   for j = 1 to ⌈n/K⌉ do
5:     π = rand(perm(C))
6:     for w = 1 to min{n, K} do
7:       h_{F((j−1)K+w,1)} = h_{F((j−1)K+w,2)} = π(w)
8:     end for
9:   end for
10: end for
11: end repeat
```

4.3 TDBS-SU: Sequential Unicast Mode

In the SU mode, a sender sequentially unicasts the broadcast message to $(2n-1)$ intended receivers. The hopping sequences are constructed as follows.

Step 1: Construct a 1-factorization $\mathcal{F}_{2n}$ of K_{2n}, where $\mathcal{F}_{2n} = \{F_0, F_1, \ldots, F_{2n-2}\}$.
Step 2: For all $F_i \in \mathcal{F}_{2n}$, $0 \le i \le 2n-2$, repeat Steps 3–5.
Step 3: Obtain a random permutation π of the set of frequency bands $\mathcal{C}$.
Step 4: Assign frequency bands in π to $\min\{n, K\}$ edges of F_i in the order of occurrence of the edges.
Step 5: Repeat Steps 3 and 4 until all pairs in F_i are assigned a frequency band.
Step 6: Repeat Steps 1-5.

The pseudo-code of the hopping sequence construction for the SU mode is shown in Algorithm 1. In figure 3(c), we show an example of the application of Algorithm 1 to a group of four nodes. The set of available channels is $\mathcal{C} = \{f_1, \ldots, f_5\}$, $(K = 5)$. Because $K \ge n$, the n pairs corresponding to a 1-factor can communicate in parallel in one slot. In slot 0, pairs communicate according to factor F_0. The random permutation of $\mathcal{C}$ is $\pi = \{f_2, f_3, f_5, f_1, f_4\}$. Pair $(1,2)$ is assigned band $\pi(1) = f_2$ and pair $(3,4)$ is assigned band $\pi(2) = f_3$. The process is repeated for factors F_1, and F_2. Note that condition $K \ge n$ is not necessary for the correct operation of our algorithm. When the number of frequency bands is smaller than the pairs of communicating nodes, transmissions corresponding to one factor are split in multiple slots, as shown in Steps 3-5. However, for single hop networks, it is expected that $K >> n$. We now show that Algorithm 1 constructs random FH sequences.

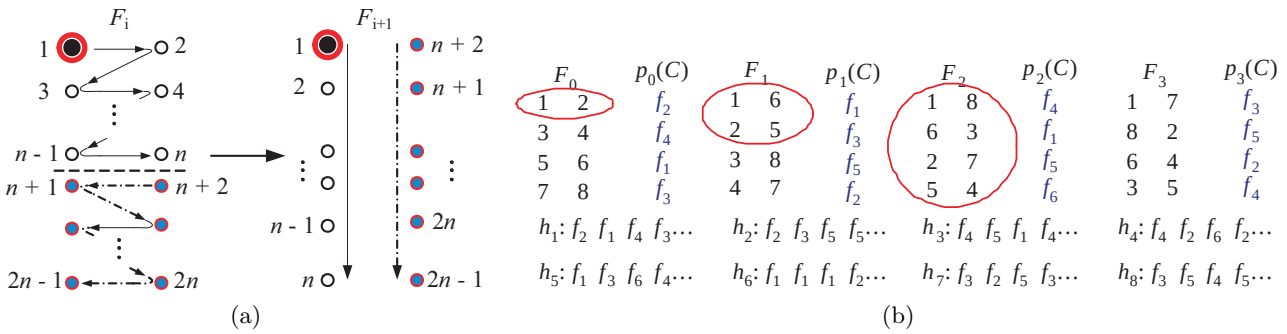


Figure 4: (a) Splitting algorithm used to obtain the 1-factor F_{i+1} from the 1-factor F_i. The first n nodes of F_i are obtained in a "zigzag" fashion and are placed on the first column of F_{i+1}. The last n nodes of F_i are obtained in an "inverse zigzag" fashion and are placed in the second column of F_{i+1}, (b) the first four 1-factors for a group of eight nodes and the corresponding hopping sequences.

PROPOSITION 1. *The FH sequences constructed by Algorithm 1 are random.*

PROOF. Let $h_j = \{X_1, X_2, \ldots\}$ denote a FH sequence constructed by Algorithm 1 for a node j, where X_i is a random variable denoting the frequency band used at slot i. Random variables X_i form an i.i.d. with each variable being randomly distributed (frequency bands at Step 4 are randomly and independently selected). Hence, h is random. □

4.4 TDBS-AB: Assisted Broadcast Mode

Algorithm 2 TDBS-AB: Assisted Broadcast Mode

1: Generate random F_0 of K_{2n}
2: initialize $i = 0$
3: **repeat**
4: **for** $j = 1$ to $\lceil \frac{n}{K} \rceil$ **do**
5: $\pi = rand(\text{perm}(\mathcal{C}))$
6: **for** $w = 1$ to $\min\{n, K\}$ **do**
7: $h_{F_i((j-1)K+w,1)} = h_{F_i((j-1)K+w,2)} = \pi(w)$
8: **end for**
9: **end for**
10: $F_{i+1} = \texttt{split}(F_i)$
11: $i++$
12: **end repeat**

In the AB mode, any node that has already received a broadcast message operates as a broadcast relay. To construct hopping sequences for the AB mode, the 1-factors F_i are selected and arranged in such a way that the *number of nodes that can relay a broadcast transmission in each 1-factor is maximized*. This property minimizes the delay until the broadcast is completed, while increasing resilience to jamming. We first define the notion of the *relay set*.

DEFINITION 4. **The Relay Set** R_j^i *of node j in a 1-factor F_i is defined as the set of nodes that can relay a transmission that originated from j.*

The main idea of our hopping sequence construction algorithm is to maximize the size of the relay set R_j^i, for every node j and in every 1-factor F_i. Note that in the AB mode, it is not necessary that the series of 1-factors form a 1-factorization (i.e., that all pairs of nodes communicate directly), because nodes can receive the broadcast transmission via multiple hops. The hopping sequences assigned to each node are constructed as follows.

Algorithm 3 Splitting Algorithm `split`

1: $F_{i+1}(1,1) = F_i(1,1)$
2: **if** n even **then**
3: $F_{i+1}(1,2) = F_i(\frac{n}{2}+1, 2)$
4: **else**
5: $F_{i+1}(1,2) = F_i(\lceil \frac{n}{2} \rceil, 2)$
6: **end if**
7: **for** $j = 2$ to n **do**
8: $F_{i+1}(j,1) = F_i(\lceil \frac{j}{2} \rceil, 2)$, if j even
9: $F_{i+1}(j,1) = F_i(\lceil \frac{j}{2} \rceil, 1)$, if j odd
10: **if** n even **then**
11: $F_{i+1}(j,2) = F_i(\lceil \frac{n+j}{2} \rceil, 1)$, if j even
12: $F_{i+1}(j,2) = F_i(\lceil (\frac{n+j}{2} \rceil, 2)$, if j odd
13: **else**
14: $F_{i+1}(j,2) = F_i(\lceil \frac{n+j}{2} \rceil, 2)$, if j even
15: $F_{i+1}(j,2) = F_i(\lceil (\frac{n+j}{2} \rceil, 1)$, if j odd
16: **end if**
17: **end for**

Step 1: Obtain an arbitrary 1-factor F_0 of K_{2n}. Set $i = 0$.
Step 2: Obtain a random permutation π of the set of frequency bands $\mathcal{C}$.
Step 3: Assign frequency bands in π to $\min\{n, K\}$ edges of F_i in the order of occurrence of the edges.
Step 4: Repeat Steps 2 and 3 until all pairs in F_i are assigned a frequency band.
Step 5: Construct 1-factor F_{i+1} according to the *splitting algorithm*. Set $i = i + 1$.
Step 6: Repeat Steps 2 and 5.

The pseudo-code of TDBS-AB is shown in Algorithm 2. The pseudo-code of the splitting algorithm employed to generate F_{i+1} from F_i is shown in Algorithm 3, and illustrated in figure 4(a). Every pair of nodes that communicate according to the 1-factor F_i are placed in adjacent rows in the 1-factor F_{i+1}. The propagation of this property in subsequent 1-factors minimizes the broadcast delay by maximizing the size of the relay set R_j^i for any j and for every 1-factor.

To illustrate the application of Algorithm 2, consider a network of eight nodes. The first four 1-factors that are generated by our algorithm and the corresponding hopping sequences assigned to various nodes are shown in figure 4(b). Node 1 initiates a broadcast transmission of message m following the 1-factor F_0. The circles mark the nodes that receive message m after the completion of the unicasts corre-

sponding to various 1-factors. In fact, one can verify from the 1-factors shown in Fig. 4(b) that any broadcast transmission initiated under 1-factor F_0 is completed by 1-factor $F_{\log_2(8)-1} = F_2$. In section 6, we prove that this property holds for any broadcast initiated at any time slot. Note that TDBS-AB uses the same mechanism as TDBS-SU (Steps 2-4) for assigning frequency bands to communicating pairs. Therefore, Proposition 1 holds for the hopping sequences generated by TDBS-AB. These sequences are uniformly distributed over the set of available channels, thus minimizing the success of an external jammer in guessing the frequency bands of future communications based on past observations. Moreover, compromise of sequences limits the information leakage regarding other sequences.

5. TDBS IN MULTI-HOP NETWORKS

In this section, we extend the operation of TDBS to multi-hop networks. In this scenario, the FH sequence design can be viewed as a global scheduling problem. While several distributed methods have been proposed for distributed scheduling (e.g., [7,11]), we note that these methods require coordination via a commonly accessible channel. However, such a channel can be blocked by an inside jammer. We, therefore, develop a scalable solution based on clustering, that does not require node coordination.

We partition the network into clusters where each cluster forms a clique [13,28]. Clique clustering produces a network partition where every node belongs to a single cluster and the members of each cluster are within one hop. We then divide the broadcast operation into two phases: (a) the intra-cluster phase, and (b) the inter-cluster phase. During the intra-cluster phase, communication is limited within each cluster. In the inter-cluster phase, messages are exchanged between border nodes of adjacent clusters. The two phases are interleaved in time.

5.1 Intra-cluster Phase

In the intra-cluster phase, a broadcast message propagates to all nodes within a cluster. Because the nodes of a cluster form a clique, the SU or AB operation modes for single-hop networks are employed. To avoid interference between adjacent clusters, the set of available frequency bands $\mathcal{C}$ is divided into four mutually exclusive sets which are assigned to each cluster according to the four color theorem [2].

One such assignment is shown in figure 5(a). The shading pattern of each cluster denotes a separate set of frequency bands. In this example, 10 frequency bands are assigned to each cluster, yielding a $K = 40$. Note that the number of available frequency bands K is expected to be be much larger than the number of nodes within the same collision domain (i.e., cluster size). In any case, the algorithms outlined in Sections 4.3 and 4.4, produce FH sequences for any relation between K and n. The steps for deriving FH sequences for the intra-cluster phase are as follows.

Step 1: Color each cluster based on the four-color theorem.
Step 2: For each distinct cluster size $2n$, construct a 1-factorization $\mathcal{F}_{2n}$ of K_{2n}.
Step 3: For each cluster, pick the 1-factorization corresponding to its cluster size and construct FH sequences for the cluster nodes following the SU mode or the AB mode.
Step 4: Repeat Steps 2 and 3 until all clusters are processed.

In Step 2, it is sufficient to produce distinct 1-factorizations for every possible cluster size. Two clusters of the same size can use the same 1-factorization, dictating the rendezvous of its cluster members, respectively. However, due to the random permutation assignment of frequency bands in Step 3, the pairs of nodes of each cluster will communicate at different frequency bands, thus ensuring the randomness of the pairwise communication among pairs.

5.2 Inter-cluster Phase

In the inter-cluster phase, border nodes in adjacent clusters relay broadcast messages that are intended to propagate beyond the boundaries of each cluster. To do so, while avoiding collisions between adjacent transmissions, we exploit the cluster labeling produced by the application of the four-color theorem. During this phase, every time-slot is marked with one of the four colors indicating the set of clusters that are allowed to transmit on that slot. As an example, in figure 5, clusters A and D are allowed to transmit on slot 0, clusters C and F on slot 1, clusters B and E on slot 2 and cluster G on slot 3, with this sequence repeating modulo four (slot numbers indicate assignment before the interleaving with the intra-cluster phase) . After the slot coloring, the FH sequences of individual nodes are generated as follows.

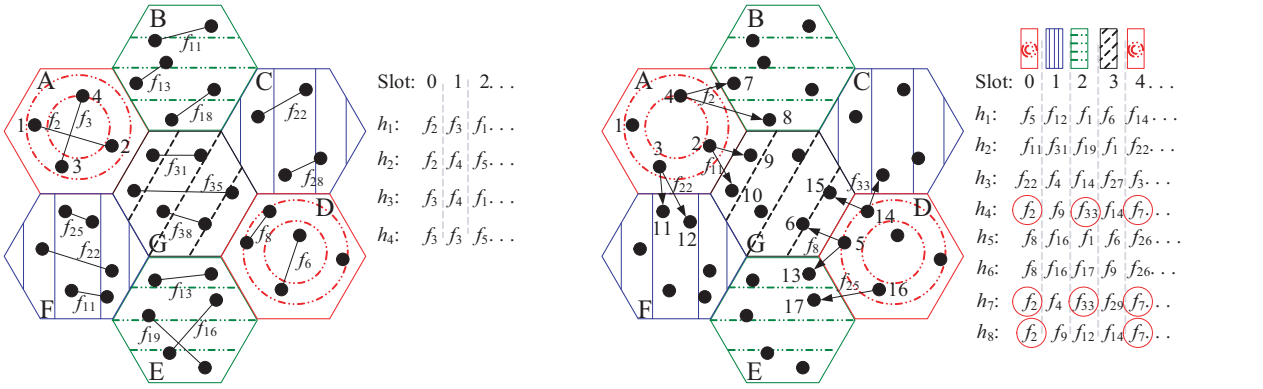
Step 1: For each cluster x, find the nodes in x bordering adjacent clusters. Place this nodes to a set $\mathcal{A}$.
Step 2: For each node $i \in \mathcal{A}$, find the neighbors of i in adjacent clusters. If a neighbor is common to two nodes in x, assign it to the node with the fewer neighbors. Break ties arbitrarily (e.g., considering the node with the lowest id). Merge nodes assigned to the same i to a single vertex and place vertices to set $\mathcal{B}$. Create a bipartite graph $G(\mathcal{A} \cup \mathcal{B}, \mathrm{E})$, where an edge (x,y) exists if nodes corresponding to y are assigned to x. G forms a 1-factor F_x.
Step 3: For each slot colored with x's color, obtain a random permutation π of the set of frequency bands $\mathcal{C}$.
Step 4: Assign frequency bands in π to $\min\{n, K\}$ edges of F_x in the order of occurrence of the edges.
Step 5: Repeat Steps 3 and 4 until all pairs in F_x are assigned a frequency band.
Step 6: Repeat Steps 1-5, until all clusters are processed.

The inter-cluster phase is illustrated in Figure 5(b). According to their color, clusters A and D are scheduled to broadcast during slot 0. Nodes 2,3, and 4 belong to set $\mathcal{A}$ of cluster A since they can communicate with nodes of adjacent clusters. For slot 0, the communicating pairs are $\{2-9, 10\}$ $\{3-11, 12\}$ and $\{4-7, 8\}$, and are assigned frequency bands f_{11}, f_{22}, and f_2, respectively. Similarly, for cluster D and slot 0, the communicating pairs are $\{5-6, 13\}$ $\{14-15\}$ and $\{16-17\}$, and are assigned frequency bands f_8, f_{33}, and f_{25}, respectively. Note that during the inter-cluster phase, all channels in $\mathcal{C}$ are available for assignment to the communications of adjacent pairs of nodes.

The intra-cluster and inter-cluster slots are interleaved in the FH design, to allow for both single hop and multi-hop broadcast transmissions are achieved.

6. PERFORMANCE AND SECURITY EVALUATION

In this section, we evaluate the performance of TDBS and



(a) (b)

Figure 5: (a) The intra-cluster phase, (b) the inter-cluster phase.

analyze its security properties. As a performance/security metric, we use the broadcast delay, defined as follows.

DEFINITION 5. **The Broadcast Delay** *D is the number of slots required for the completion of a broadcast operation, i.e., until all intended recipients have received a copy of the broadcasted message.*

6.1 Performance in the Absence of Jammers

In this section, we evaluate the broadcast delay for the two TDBS modes in the absence of jammers. This analysis is provided to facilitate the evaluation of the broadcast delay when jammers are assumed to be present.

PROPOSITION 2. *The broadcast delay of TDBS-SU is $D = \lceil \frac{n}{K} \rceil (2n-1)$ slots.*

PROOF. The proof is provided in Appendix A. □

Next, we evaluate the broadcast delay in the AB mode.

PROPOSITION 3. *The broadcast delay for TDBS-AB is $D = \lceil \frac{n}{K} \rceil \lceil \log_2(2n) \rceil$ slots.*

PROOF. The proof is provided in Appendix B. □

6.2 Security Analysis

We first analyze the resilience of TDBS to external and internal jammers for single-hop networks.

6.2.1 Resilience to External Jammers

Under an external threat model, the hopping sequences assigned to various nodes remain secret. For this scenario, we assume that the adversary deploys multiple jamming devices that can jam up to J frequency bands per time slot, with $J < K$. For convenience, we assume $K \geq n$ so that all node pairs corresponding to a 1-factor can communicate in parallel in one time-slot. This is typical in wideband communications where K is much larger than the expected number of nodes within the same collision domain. Our results can be extended to the $K < n$ case in a straightforward manner. Suppose that a jammer attempts to jam the broadcast of a single node j. To compute D, we evaluate the average number of 1-factorizations needed to complete the broadcast, in the presence of the external jammer, and for each mode.

PROPOSITION 4. *In the presence of an external jammer, the expected number $\mathrm{E}[Z]$ of 1-factorizations needed to complete a broadcast operation in the SU mode is*

$$\mathrm{E}[Z] = (1-p)^{2n-1} + \sum_{i=2}^{\infty} i(1-p^{i-1})^{2n-1} \times \sum_{k=1}^{2n-1} \binom{2n-1}{k} \left(\frac{p^{i-1}(1-p)}{1-p^{i-1}}\right)^k, \quad (1)$$

where $p = \frac{J}{K}$ denotes the jamming probability.

PROOF. The proof is provided in Appendix C. □

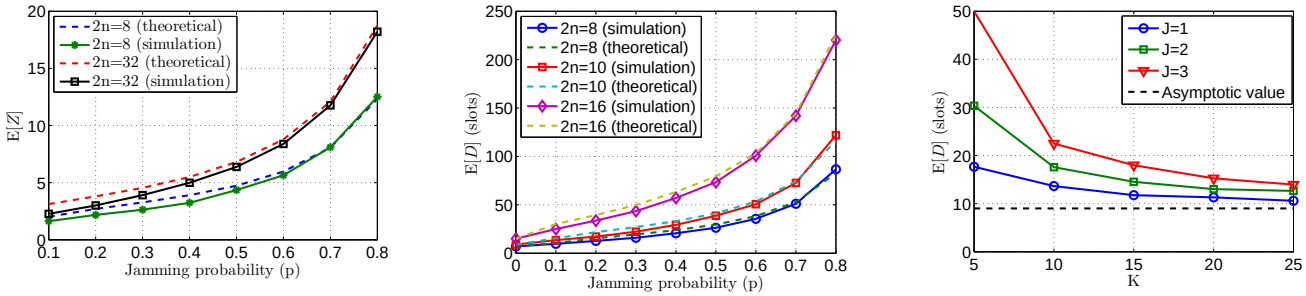
In figure 6(a), we compare the theoretical value of $\mathrm{E}[Z]$ with the simulated one. For our simulations, we generated sequences of size $1,000$ hops for different values of n and K according to Algorithm 1. We also randomly selected J channels to be jammed per time-slot. All results were averaged over 100 runs. We measured $\mathrm{E}[Z]$ as a function of the jamming probability $p = \frac{J}{K}$. We observe that the simulation values agree with the theoretical ones.

Based on Proposition 4, the expected broadcast delay $\mathrm{E}[D]$ is equal to the expected number of 1-factorizations needed for the completion of a broadcast, times the number of slots needed for the completion of each 1-factorization. The first $(E[Z]-1)$ 1-factorizations require $(2n-1)$ slots, while the last 1-factorization requires, on average, $\frac{2n-1}{2}$ slots (the last successful transmission takes place on any of the 1-factors of the last 1-factorization with equal probability). Therefore, $\mathrm{E}[D] = (2n-1)\left(\mathrm{E}[Z] - \frac{1}{2}\right)$.

Figure 6(b), shows the theoretical and simulated value of $\mathrm{E}[D]$ as a function of the jamming probability p. We observe that even when the adversary jams 80% of the available channels, nodes are still capable of completing their broadcast transmissions at the expense of some delay. Nevertheless, the broadcast communication is maintained. In figure 6(c), we show $\mathrm{E}[D]$ as a function of the number of available channels K for various values of J. $\mathrm{E}[D]$ decreases with K, approaching the asymptotic value of K, obtained in the absence of a jammer, i.e., $\mathrm{E}[D] = 2n-1$.

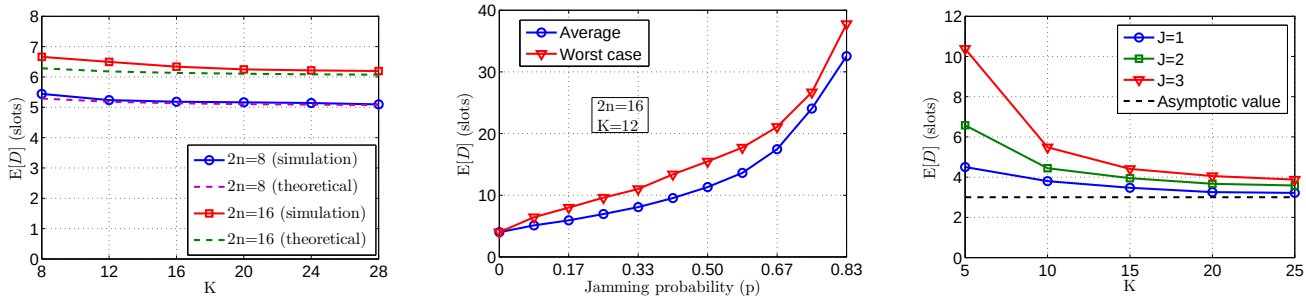
For the AB mode, $\mathrm{E}[D]$ does not have a simple closed-form expression but involves complex summation formulas. However, we can derive a useful formula for $J = 1$.

PROPOSITION 5. *After the first successful relay of a broadcast message m, the broadcast delay until m is received by*



(a) (b) (c)

Figure 6: (a) E[Z] as function of the jamming probability p, (b) E[D] as a function of jamming probability p. (c) E[D] as a function of K when $2n = 10$.



(a) (b) (c)

Figure 7: (a) E[D] as a function of K when $J = 1$, for the AB mode of the worst case. The theoretical value is computed based on (3), (b) E[D] as a function of p, for the AB mode. The average and worst case are shown, (c) E[D] as a function of K and for various J. The asymptotic value is equal to $\lceil \log_2(2n) \rceil$.

$(2n-2)$ nodes (all nodes, but one) is bounded by

$$\lceil \log_2(2n) \rceil - 1 \leq D \leq \lceil \log_2(2n) \rceil. \quad (2)$$

PROOF. The proof is provided in Appendix D. □

Proposition 5 allows us to estimate the expected broadcast delay for the AB mode. Let D_1 denote the expected delay until the first success, D_2 the delay until $(2n-2)$ nodes receive message m and D_3 the delay until the last node receives m. The expected broadcast delay is bounded by

$$\begin{aligned} \mathrm{E}[D] &= \mathrm{E}[D_1 + D_2 + D_3] \\ &\leq \frac{K}{K-1} + \lceil \log_2(2n) \rceil + \frac{K}{K-1}. \end{aligned} \quad (3)$$

In (3), we have used the fact that it takes, on average, $\frac{K}{K-1}$ slots for the first successful relay when $p = \frac{1}{K}$. Moreover, after the first success, $\lceil \log_2(2n) \rceil$ slots are needed in the worst case until $2n-2$ nodes receive m. The last node receives m after $\frac{K}{K-1}$ slots, on average.

We also studied the performance of the AB mode via simulations. For our simulations, we generated sequences of size $1,000$ hops for different values of n and K according to Algorithm 2. We also randomly selected J channels to be jammed per time-slot. All results were averaged over 100 runs. Figure 7(a) shows E[D] as a function of K for $J = 1$. We observe that the theoretical value derived using Proposition 5 agrees with the simulation. In figure 7(b), we show the average and worst-case broadcast delay, as a function of p. We observe that even when $p = 0.83$, the average and worst-case delays differ by less than six slots. This is due to the "relay explosion" effect of the splitting algorithm. The AB mode is significantly more resilient to jamming than the SU mode, due to the larger number of broadcast relays. Even when 83% of the frequency bands are jammed, the AB mode requires only 38 slots to complete a broadcast, compared to 228 slots needed with the SU mode. In figure 7(c), we show E[D] as a function K for different values of J. We observe that with the increase of K, E[D] asymptotically approaches the performance of the AB mode in the absence of jammers.

6.2.2 Resilience to Internal Jammers

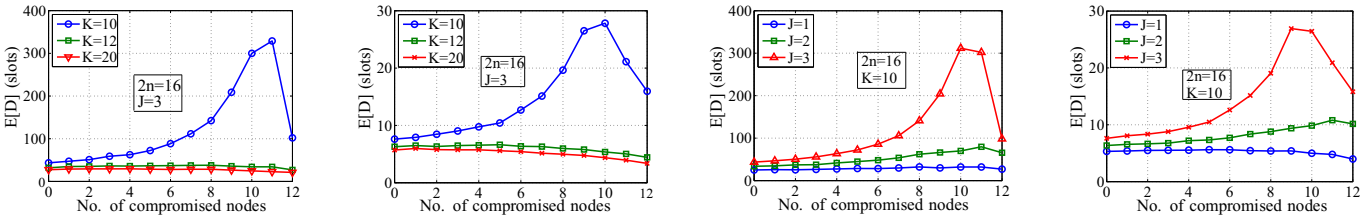
Assume now that the adversary has compromised r nodes and recovered their FH sequences. We are interested in determining the broadcast delay until the remaining $(2n-r-2)$ *legitimate* nodes receive a broadcast message m. Knowledge of the r FH sequences reduces the adversary's uncertainty with respect to the frequency locations of legitimate unicasts. This is because the space of $\mathcal{C}$ for the selection of the uncompromised FH sequences is reduced. The exact value of E[D] depends on the selection of the 1-factorization that is used to construct the hopping sequences and the specific arrangement of the compromised nodes on that 1-factorization. The jamming probability p varies on a slot-by-slot basis and is given in the following proposition.

PROPOSITION 6. *Under the compromise of r nodes, the jamming probability p is bounded by*

$$\min\{1, \frac{J}{K - \lceil \frac{r}{2} \rceil}\} \leq p \leq \min\{1, \frac{J}{K-r}\}. \quad (4)$$

PROOF. The proof is provided in Appendix E. □

We further used simulation to investigate the impact of



(a) SU mode (b) AB mode (c) SU mode (d) AB mode

Figure 8: (a), (b) E[D] as a function of the number of compromised nodes for various values of K, when $J = 3$, (c), (d) E[D] as a function of the number of compromised nodes for various values of J, when $K = 10$.

node compromise on the broadcast delay. For our simulations, we generated FH sequences of length $1,000$ hops for different values of n and K. We randomly selected r of these sequences to be exposed to the adversary. At each time slot, the adversary randomly jammed J bands, excluding the exposed ones. A broadcast was deemed successful, when all legitimate nodes obtain a message copy. All results were averaged over 100 runs. Figures 8(a) and 8(b) show E[D] as a function of the number of compromised nodes when $J = 3$ and $K = 10, 12, 20$, for the SU and AB modes, respectively. We observe that legitimate nodes complete their broadcast transmissions even when more than 50% of the nodes are compromised. The AB mode exhibits significantly lower delay compared to the SU mode, due to the use of multiple relays. Note that when K is small and several nodes are compromised, the jammers have a high chance of jamming legitimate pairs. This fact can be seen from the sharp increase of E[D] when $K = 10$.

In figure 8(c) and 8(d), we show E[D] as a function of the number of compromised nodes when $K = 10$ and for various values of J, under the SU and AB modes, respectively. Even with the increase of J, legitimate nodes are able to complete their broadcast transmissions in both modes, with the AB mode being the most efficient. Note that E[D] decreases when a large number of nodes is compromised, since fewer legitimate nodes need to receive a unicast message for completing a broadcast transmission.

6.3 Evaluation of Multi-hop Scenarios

In this section, we evaluate TDBS for multi-hop networks. We focus on the jamming-resistance of the inter-cluster phase, since for the intra-cluster phase, the security analysis for single-hop networks holds. We define the following performance metrics for the inter-cluster phase:

- **Flooding Delay** D_f**:** the number of slots needed until all clusters adjacent to a cluster i, have received a broadcast that originated in i, directly from a node in i.
- **Escape Delay** D_e**:** the number of slots needed until a broadcast message m originating at a cluster i, reaches any node in any adjacent cluster.

Escape diversity DIV**:** the fraction of adjacent clusters that receive a broadcast m directly from a cluster i, when some border nodes in i are compromised.

We first analytically evaluate the average flooding delay E[D_f] in the presence of external jammers. Assume a cluster with N_C adjacent clusters. Let N_L denote the number of "bridge links" between two adjacent clusters.

PROPOSITION 7. *In the presence of an external jammer,* E[D_f] *is equal to*

$$\mathrm{E}[D_f] = (1-\tilde{p})^{N_C} + \sum_{i=2}^{\infty} i(1-\tilde{p}^{i-1})^{N_C} \times \sum_{k=1}^{N_C} \binom{N_C}{k} \left(\frac{\tilde{p}^{i-1}(1-\tilde{p})}{1-\tilde{p}^{i-1}}\right)^k, \quad (5)$$

where $\tilde{p} = \left(\frac{J}{K}\right)^{N_L}$ *denotes the probability that all* N_L *links to an adjacent cluster are jammed at a particular slot*

PROOF. The proof of Proposition 7 follows the same steps as the proof of Proposition 4, by substituting $p = \frac{J}{K}$ with $\tilde{p} = \left(\frac{J}{K}\right)^{N_L}$. Due to space limitations we refer to the proof provided in Appendix C. □

We also verified Proposition 7 via simulations. In our setup, we generated a multi-hop topology consisting of 50 nodes, organized in clusters. We then generated FH schedules for all nodes in the network for the inter-cluster phase, according to the algorithm described in Section 5.2. At each time slot, the jammer was assumed able to block J random frequency bands across the entire network. Results were averaged over all clusters in the network. Figure 9(a) shows E[D_f] as a function of the jamming probability p. We denote the number of "bridge links" between two adjacent clusters to be N_L. We observe that, even when 80% of the available frequency bands are jammed, only 13 inter-cluster slots are needed until all neighboring clusters directly receive a broadcast. Once the message propagates to adjacent clusters, the intra-cluster phase follows. We also evaluate the expected escape delay E[D_e] under the compromise of r border nodes.

PROPOSITION 8. *Under the compromise of* r *border nodes of a cluster* i, E[D_e] *is given by*

$$\mathrm{E}[D_e] = \frac{1}{1-\left(P_c^{N_L} + \sum_{i=1}^{N_L}\binom{N_L}{i}\left(\frac{J(1-P_c)}{K-r}\right)^i\right)^{N_C}}, \quad (6)$$

where $P_c = \frac{r}{N_C \times N_L}$ *denotes the compromise probability.*

PROOF. The proof is provided in Appendix F. □

The expected escape diversity E[DIV] is evaluated in the following proposition.

PROPOSITION 9. *Under the compromise of* r *nodes,* E[DIV] *is given by*

$$\mathrm{E}[DIV] = 1 - P_c^{N_L}. \quad (7)$$

PROOF. The proof is provided in Appendix G. □

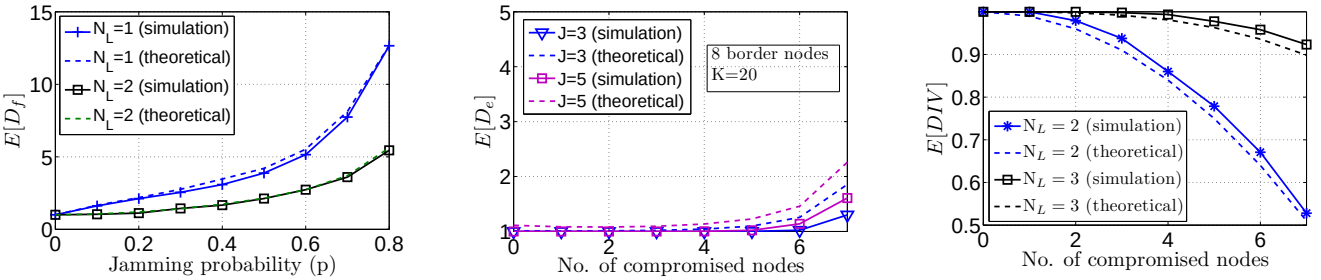


Figure 9: (a) E$[D_f]$ as a function of the jamming probability p, (b) E$[D_e]$ as a function of the number of compromised nodes r for various J, (c) E$[DIV]$ as a function of r for various N_L.

Figures 9(b) and 9(c) evaluate E$[D_e]$ and E$[DIV]$ as a function of the number of compromised border nodes. In our simulation, compromised border nodes do not relay messages and their FH sequences are assumed exposed. From figure 9(b), we observe that a small number of slots is sufficient for the first copy of a broadcast message to reach one adjacent cluster. From figure 9(c), we observe that more than 90% of neighboring clusters are guaranteed to receive the message when $N_L = 3$, while this value being reduced to 50% when $N_L = 2$.

7. RELATED WORK

The problem of jamming in wireless communications has been extensively studied under an external threat model (for example, see [1, 24] and the references therein). Jamming is typically mitigated by spreading the transmitted signal to a larger bandwidth following a secret PN code. Without knowledge of this code, the jammer has to expend several orders of magnitude more energy (typically 20-30 dB gain) to interfere with ongoing transmissions. However, in the case of broadcast communications, compromise of commonly shared PN codes suppresses the advantages of SS.

Recently, several researchers have considered the problem of jamming under an inside threat model. Chan et al. showed that a jammer that targets the broadcast control channel in GSM networks can reduce the required power for performing a DoS attack by several orders of magnitude [6]. Desmedt et al. proposed an anti-jamming scheme that protects broadcast communications from a small number of inside and colluding jammers [9]. Their method relies on combinatorial block designs to allow for partial sharing of secret information with respect to the location of the broadcast frequency bands. To protect control-channel traffic, the replication of broadcast transmissions over multiple channels whose location are cryptographically protected, was suggested in [6, 25, 26].

Alternative methods eliminate the dependence on shared secrets [4, 14, 16, 20]. Baird et al. proposed a keyless anti-jamming technique based on encoding of indelible marks at specific locations within each broadcasted message [4]. Pöpper et al. proposed a solution called Uncoordinated DSSS (UDSSS) [20]. In UDSSS, broadcast transmissions are spread according to a PN code, randomly selected from a public set of codes. Liu et. al. proposed RD-DSSS, a randomized differential DSSS scheme also relying on randomly selected PN codes [16]. Compared to UDSSS, the RD-DSSS scheme provides resilience to reactive jammers.

Several methods attempt to identify the compromised nodes that leaked information to the jammer. Lazos et al. proposed the assignment of unique frequency hopping sequences to each receiver, overlapping in a fixed subset of hops [14]. Using the uniqueness of the assigned sequences, compromised nodes whose sequences are used for jamming are identified. Tague et al. proposed the GUIDE scheme for identifying compromised nodes based on the set of control channels that are jammed. They formulated the identification problem as a maximum likelihood estimation problem [26]. Chiang and Yih-Chun Hu, developed a code-tree based approach for identifying compromised PN codes [8].

8. CONCLUSION

We proposed TDBS, a scheme for jamming-resistant broadcast communications in the presence of inside jammers. In TDBS, broadcast is realized as a series of unicast transmissions distributed in frequency and time. Because the adversary is limited in the number of channels he can jam, several unicast transmissions remain interference-free. We mapped the problem of constructing hopping sequences for TDBS to the problem of 1-factorization of complete graphs. We analytically evaluated the security properties of TDBS under an external and an internal threat model and showed that TDBS maintains broadcast communications even when multiple nodes are compromised. We verified our theoretical analysis using extensive simulations.

Acknowledgments

Part of this work was conducted while M. Krunz was a visiting researcher at the University of Carlos III, Madrid, and IMDEA Networks, Spain. This research was supported in part by NSF (under grants CNS-0844111, CNS-0721935, CNS-0904681, CNS-1016943, IIP-0832238), Raytheon, and the Connection One center. Any opinions, findings, conclusions, or recommendations expressed in this paper are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

9. REFERENCES

[1] D. Adamy. *EW 101: A first course in electronic warfare*. Artech House Publishers, 2001.

[2] K. Appel and W. Haken. Every planar map is four colorable: Part I. *Illinois Journal of Mathematics*, 21(3):491–567, 1977.

[3] P. Bahl, R. Chandra, and J. Dunagan. SSCH: slotted seeded channel hopping for capacity improvement in

IEEE 802.11 ad-hoc wireless networks. In *Proc. of MOBICOM*, pages 216–230, 2004.

[4] L. C. Baird, W. L. Bahn, M. D. Collins, M. C. Carlisle, and S. C. Butler. Keyless jam resistance. In *Proc. of the IEEE Workshop on Information Assurance United States Military Academy*, 2007.

[5] K. Bian, J. Park, and R. Chen. A quorum-based framework for establishing control channels in dynamic spectrum access networks. In *Proc. of MOBICOM*, pages 25–36, 2009.

[6] A. Chan, X. Liu, G. Noubir, and B. Thapa. Control channel jamming: Resilience and identification of traitors. In *Proc. of ISIT*, 2007.

[7] P. Chaporkar, K. Kar, X. Luo, and S. Sarkar. Throughput and fairness guarantees through maximal scheduling in wireless networks. *IEEE Transactions on Information Theory*, 54(2):572–594, 2008.

[8] J. T. Chiang and Y.-C. Hu. Dynamic jamming mitigation for wireless broadcast networks. In *Proc. of INFOCOM*, pages 1211–1219, 2008.

[9] Y. Desmedt, R. Safavi-Naini, H. Wang, C. Charnes, and J. Pieprzyk. Broadcast anti-jamming systems. In *Proc. of the IEEE International Conference on Networks (ICON)*, pages 349 – 355, 1999.

[10] J. H. Dinitz and D. R. Stinson. A hill-climbing algorithm for the construction of one-factorizations and room squares. *SIAM J. Algebraic Discrete Methods*, 8(3):430–438, 1987.

[11] A. Gupta, X. Lin, and R. Srikant. Low-complexity distributed scheduling algorithms for wireless networks. *IEEE/ACM Transactions on Networking (TON)*, 17(6):1846–1859, 2009.

[12] IEEE. IEEE 802.11 for wireless local area networks. `http://www.ieee802.org/11/`.

[13] H. Ishii and H. Kakugawa. A self-stabilizing algorithm for finding cliques in distributed systems. In *Proc. of the 21st IEEE Symposium on Reliable Distributed Systems (SRDS'02)*, pages 390–395, 2002.

[14] L. Lazos, S. Liu, and M. Krunz. Mitigating control-channel jamming attacks in multi-channel ad hoc networks. In *Proc. of WiSec*, pages 169–180, 2009.

[15] A. Liu, P. Ning, H. Dai, Y. Liu, and C. Wang. Defending DSSS-based broadcast communication against insider jammers via delayed seed-disclosure. In *Proc. of the Annual Computer Security Applications Conference (ACSAC'10)*, 2010.

[16] Y. Liu, P. Ning, H. Dai, and A. Liu. Randomized differential DSSS: Jamming-resistant wireless broadcast communication. In *Proc. of INFOCOM*, 2010.

[17] E. Mendelsohn and A. Rosa. One-factorizations of the complete graph-a survey. *Journal of Graph Theory*, 9(1):43–65, 1985.

[18] G. Noubir and G. Lin. Low-power DoS attacks in data wireless LANs and countermeasures. *Mobile Computing and Communications Review*, 7(3):29–30, 2003.

[19] R. Poisel. *Modern communications jamming principles and techniques*. Artech House on Demand, 2004.

[20] C. Popper, M. Strasser, and S. Capkun. Jamming-resistant broadcast communication without shared keys. In *Proc. of the USENIX Security Symposium*, 2009.

[21] C. Popper, M. Strasser, and S. Capkun. Anti-jamming broadcast communication using uncoordinated spread spectrum techniques. *IEEE Journal on Selected Areas in Communications*, 28(5), 2010.

[22] S. Sarkar and L. Tassiulas. End-to-end bandwidth guarantees through fair local spectrum share in wireless ad-hoc networks. *IEEE Transactions on Automatic Control*, 50(9):1246–1259, 2005.

[23] G. Sharma, C. Joo, and N. Shroff. Distributed scheduling schemes for throughput guarantees in wireless networks. In *Proc. of the 44th Annual Allerton Conference on Communications, Control, and Computing*, 2006.

[24] M. K. Simon, J. K. Omura, R. A. Scholtz, and B. K. Levitt. *Spread Spectrum Communications Handbook*. McGraw-Hill, 2001.

[25] P. Tague, M. Li, and R. Poovendran. Probabilistic mitigation of control channel jamming via random key distribution. In *Proc. of IEEE PIMRC*, pages 1–5, 2007.

[26] P. Tague, M. Li, and R. Poovendran. Mitigation of control channel jamming under node capture attacks. *IEEE Transactions on Mobile Computing*, 8(9):1221–1234, 2009.

[27] L. Tassiulas and A. Ephremides. Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. *IEEE Transactions on Automatic Control*, 37(12):1936–1948, 2002.

[28] P. Tosic and G. Agha. Maximal clique-based distributed group formation for autonomous agent coalitions. In *Proc. of the Third International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2004)*, 2004.

[29] W. Wallis. One-factorizations of complete graphs. *Contemporary Design Theory: A Collection of Surveys*, pages 692–731, 1992.

[30] W. Wallis. *One-factorizations*. Kluwer Academic Publishers, 1997.

APPENDIX

A. Proof of Proposition 2: To complete a broadcast in the SU mode, the sender must unicast the broadcast message to $(2n-1)$ receivers. The $(2n-1)$ transmissions correspond to the $(2n-1)$ 1-factors of $\mathcal{F}_{2n}$. Each factor requires $\lceil \frac{n}{K} \rceil$ time slots to be completed (here, all transmissions of a 1-factor are completed before transmissions of other 1-factors can proceed, in order to avoid schedule conflicts). Hence, the broadcast delay is equal to $\lceil \frac{n}{K} \rceil$ times the number of factors of $\mathcal{F}_{2n}$.

B. Proof of Proposition 3: We first prove that any broadcast transmission in the AB mode is completed after $\lceil \log_2(2n) \rceil$ 1-factors. Without loss of generality, assume that a broadcast is initiated by node $F_i(k,1)$, located in the kth row of F_i. With the completion of F_i, the relay set is $R_i^j = \{F_i(k,1), F_i(k,2)\}$. After the execution of Algorithm 3, nodes $F_i(k,1)$ and $F_i(k,2)$ appear in adjacent rows (due to the cyclic nature of Algorithm 3, rows 1 and 8 are considered adjacent) on the 1-factor F_{i+1}. This can be easily

verified by reversing the mapping from F_{i+1} to F_i in lines 8-15 of Algorithm 3. Because the pair $(F_i(k,1), F_i(k,2))$ appears on separate rows on F_{i+1}, each node will relay a broadcast to two new nodes, thus increasing R^j_{i+1} to four.

Further execution of Algorithm 3 divides the nodes in the relay set R^j_{i+1} to four adjacent rows. Since none of the nodes in R^j_{i+1} appears on the same row, the relay set after the completion of factor F_{i+1} increases to eight nodes. Following the recursive application of the splitting algorithm, the relay set after the completion of $\lfloor \log_2(2n) \rfloor$ 1-factors has a size of $2^{\lfloor \log_2 2n \rfloor}$. If $\lfloor \log_2(2n) \rfloor = \log_2(2n)$, the broadcast is complete since $2^{\log_2(2n)} = 2n$. Otherwise, one extra 1-factor is needed to relay the broadcast to the remaining $2n - 2^{\lfloor \log_2(2n) \rfloor}$ nodes. Because $2^{\lfloor \log_2(2n) \rfloor} > n$, the splitting algorithm places n nodes from the relay set into the n rows of the $\lfloor \log_2 2n \rfloor + 1 = \lceil \log_2(2n) \rceil$th 1-factor. These n relays complete the broadcast operation. Combining the two cases yields a required number of 1-factors that is equal to $\lceil \log_2(2n) \rceil$. Proposition 3 follows by noting that every 1-factor requires $\lceil \frac{n}{k} \rceil$ slots to be completed.

C. Proof of Proposition 4: Suppose that an arbitrary node j attempts a broadcast transmission in the presence of an external jammer. This broadcast is completed in a single 1-factorization if the jammer is unsuccessful in jamming the communication of j for $2n-1$ consecutive slots. Because h_j is random, a transmission of node j is successful with probability $\left(1 - \frac{J}{K}\right)$. Moreover, the events of a successful transmission of node j at slot i and slot w, $i \neq w$ are independent. Hence,

$$\Pr[Z = 1] = \left(1 - \frac{J}{K}\right)^{2n-1} = (1-p)^{2n-1}.$$

The broadcast is completed in two 1-factorizations if every receiver is jammed at most one time, and at least one receiver is jammed on the first 1-factorization. Taking into account all possible combinations,

$$\Pr[Z = 2] = \sum_{k=1}^{2n-1} \binom{2n-1}{k} (1-p)^{2n-1-k} p^k (1-p)^k.$$

Generalizing to the case of $Z = i$, it follows that

$$\begin{aligned} \Pr[Z = i] &= \sum_{k=1}^{2n-1} \binom{2n-1}{k} (1-p^{i-1})^{2n-1-k} \\ &\quad p^{(i-1)k} (1-p)^k, \\ &= (1-p^{i-1})^{2n-1} \sum_{k=1}^{2n-1} \binom{2n-1}{k} \\ &\quad \left(\frac{p^{i-1}(1-p)}{1-p^{i-1}}\right)^k. \end{aligned}$$

Proposition 4 follows from the definition of the expectation, i.e., $\mathrm{E}[Z] = \sum_i i \Pr[Z = i]$.

D. Proof of Proposition 5: The lower bound immediately follows from Proposition 3. The broadcast delay in the absence of a jammer is equivalent to the delay in the presence of an external jammer who is unsuccessful in jamming any communicating pair for $\lceil \log_2(2n) \rceil - 1$ slots. Hence, after the first successful relay, the lower bound on D follows.

To compute the upper bound on D, assume that an arbitrary node j wants to broadcast a message m to the remaining $(2n-1)$ nodes. Let a_i denote the size of the relay set in slot i. Initially, $a_0 = 2$, i.e., node j has completed its first successful relay. Once $a_i \geq 2$, the adversary can jam at most one of the pairs relaying m. The size of the relay set in this worst-case scenario grows according to the formula.

$$a_i = 2a_{i-1} - 1 = 2^i + 1, \; i \leq \lceil \log_2(2n) \rceil - 1, \qquad (8)$$

where a_i is computed recursively with $a_0 = 2$. To show the validity of (8), we refer to the proof of Proposition 3, where we showed that for $a_i \leq n$, the size of the relay set doubles with the increment of i. Because the adversary jams at most one frequency band per time slot, in the worst case, $a_i = 2a_{i-1} - 1$. This is true until $a_i \geq n$, in which case the size of the relay set can no longer double. In slot i, $i \leq \lceil \log_2(2n) \rceil - 1$, the relay set becomes larger than n for the first time. That is, it takes $i = \lceil \log_2(2n) \rceil - 1$ slots until more than half the nodes can relay message m. These $a_i \geq n$ relay nodes communicate with the remaining $2n - 2^i - 1 \leq n$ nodes that have not yet received m. Since only one frequency band is jammed, the number of nodes that have received m at the end of slot $(i+1)$ is equal to $(2n-2)$. In this worst case, only one node has not received m after $\lceil \log_2(2n) \rceil$ slots.

E. Proof of Proposition 6: Let x be the number of frequency bands over which the r compromised nodes are scheduled to communicate according to the 1-factor F. The number of bands over which legitimate communications take place in each slot is reduced to $K - x$. Hence, the jamming probability is increased to $p = \frac{J}{K-x}$. To derive bounds on p, we consider the lowest and highest values of x. If the compromised nodes are scheduled to communicate with each other at 1-factor F, then $x = x_{\min} = \lceil \frac{r}{2} \rceil$, where the ceiling function is used to account for an odd r. This value of x yields the lower bound on p. On the other hand, if all r nodes are scheduled to communicate with legitimate ones (appear on separate rows in F), then $x = x_{\max} = r$, and p attains its maximum value. Note that $p \leq 1$ and hence, $r \leq K - J$. When r is larger than $K - J$, there are 1-factors where all transmissions are jammed with certainty.

F. Proof of Proposition 8: At each time slot, the probability that an adjacent cluster fails to receive a broadcast is due to: (a) all N_L links are shared with compromised border nodes, and (b) the links shared with uncompromised border nodes are jammed by the adversary. So the probability that a neighboring cluster fails to receive a broadcast is

$$P_{fail} = P_c^{N_L} + \sum_{i=1}^{N_L} \binom{N_L}{i} \left(\frac{J(1-P_c)}{K-r}\right)^i.$$

The probability that at least one of the neighboring clusters successfully receive the broadcast at a time slot is

$$P_{success} = 1 - P_{fail}^{N_C}.$$

The broadcast among adjacent nodes forms a Bernoulli trial with a success probability $P_{success}$, so the average delay until the first success is $1/P_{success}$, which leads to our result.

G. Proof of Proposition 9: For any neighboring cluster, the probability that it can not receive a broadcast is that all N_L links are shared with compromised border nodes. This probability is $P_c^{N_L}$. So the expected number of neighboring clusters that can get a broadcast is $N_C \cdot (1 - P_c^{N_L})$. Dividing this value with N_C, yields $\mathrm{E}[DIV]$.