SCISPACE
formerly Typeset

Open access • Proceedings Article • DOI:10.1145/1514274.1514299

Mitigating control-channel jamming attacks in multi-channel ad hoc networks

— Source link

Loukas Lazos, Sisi Liu, Marwan Krunz

Institutions: University of Arizona

Published on: 16 Mar 2009 - Wireless Network Security

Topics: Control channel, Wireless ad hoc network, Optimized Link State Routing Protocol, Jamming and Frequency-hopping spread spectrum

Related papers:

- The feasibility of launching and detecting jamming attacks in wireless networks
- Randomized Differential DSSS: Jamming-Resistant Wireless Broadcast Communication
- Broadcast Control Channel Jamming: Resilience and Identification of Traitors
- Jamming and sensing of encrypted wireless ad hoc networks
- Jamming-resistant Key Establishment using Uncoordinated Frequency Hopping

Share this paper:

View more about this paper here: https://typeset.io/papers/mitigating-control-channel-jamming-attacks-in-multi-channel-42h1vbtzr5

Mitigating Control-Channel Jamming Attacks in Multi-channel Ad Hoc Networks

Loukas Lazos, Sisi Liu, and Marwan Krunz
Dept. of Electrical and Computer Engineering, University of Arizona, Tucson, AZ, USA
{llazos, sisimm, krunz}@ece.arizona.edu

ABSTRACT

We address the problem of control-channel jamming attacks in multi-channel ad hoc networks. Deviating from the traditional view that sees jamming attacks as a physical-layer vulnerability, we consider a sophisticated adversary who exploits knowledge of the protocol mechanics along with cryptographic quantities extracted from compromised nodes to maximize the impact of his attack on higher-layer functions. We propose new security metrics that quantify the ability of the adversary to deny access to the control channel, and the overall delay incurred in re-establishing the control channel. We also propose a randomized distributed scheme that allows nodes to establish a new control channel using frequency hopping. Our method differs from classic frequency hopping in that no two nodes share the same hopping sequence, thus mitigating the impact of node compromise. Furthermore, a compromised node is uniquely identified through its hop sequence, leading to its isolation from any future information regarding the frequency location of the control channel.

Categories and Subject Descriptors

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

General Terms

Security

Keywords

Jamming, Denial of Service, Control channel, Ad hoc networks, Multi channel

1. INTRODUCTION

Multi-channel wireless networks utilize several orthogonal frequency bands to eliminate interference between parallel transmissions and hence, improve the network capacity [1]. Due to their increased performance compared to single-channel networks, they are being integrated into various network architectures, such as mobile ad hoc, vehicular, sensor, wireless local area, mesh, and cognitive radio networks. However, the increased capacity of multi-channel networks can be translated into actual throughput only if critical network functions such as channel allocation and routing are efficiently coordinated. These functions are collaboratively coordinated by exchanging messages on a broadcast channel known as the *control channel*. In this work, we define the control channel as *a frequency band used to broadcast messages for coordinating network functions.*

From a security point of view, convergence on a preassigned control channel constitutes a single point of failure. An adversary can severely degrade the network performance by launching a Denial of Service (DoS) attack on the control channel, thus negating any gain due to the availability of multiple data channels. One of the simplest DoS attacks is the jamming of the communication medium. In this attack, an adversary interferes with the set of frequency bands used for communication by transmitting a continuous jamming signal [2], or several short jamming pulses [3]. Traditionally, jamming attacks have been analyzed and addressed as a degradation in performance at the physical layer. However, a sophisticated adversary can intelligently utilize jamming to attack higher-layer functionalities and deny network availability at a very small energy cost [3, 4].

In fact, it was shown that jamming the control channel in GSM networks reduces the required power for performing a DoS attack by several orders of magnitude in [4, 5]. Control-channel jamming is particularly devastating for wireless ad hoc networks due to their cooperative nature. In such networks, the majority of network functions, including neighbor discovery and authentication, clustering, multiple access control, and routing, are actualized through the cooperation of all hosts in the network. Hence, control messages exchange among nodes within the same vicinity is frequent.

1.1 Problem Motivation

The impact of control-channel jamming in ad hoc networks can propagate way beyond the physical jamming range of an adversary, defined as the area within which packets are corrupted due to jamming. A sophisticated adversary can combine control-channel jamming with his knowledge of protocol specifics to impact different network layers. As an example, consider the implementation of a reactive routing protocol [13, 14] in multi-channel networks. Assume that nodes use a predefined control channel to broadcast route request (RREQ) messages for the purpose of route discovery.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
WiSec'09, March 16–18, 2009, Zurich, Switzerland.
Copyright 2009 ACM 978-1-60558-460-7/09/03 ...$5.00.

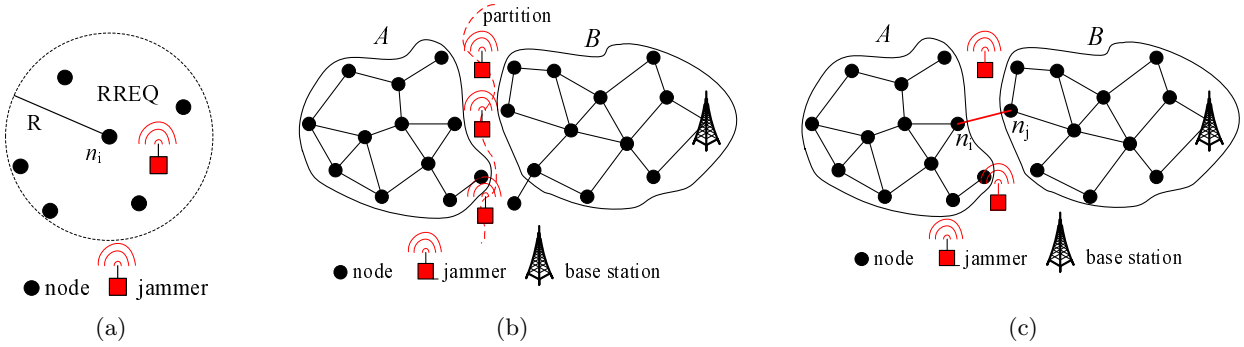

Figure 1: (a) The adversary jams route requests (RREQ) broadcasted by node n_i on the control channel, (b) the adversary partitions the network into two components A and B, by deploying multiple jamming devices, so nodes in A cannot alert their base station regarding the jamming attack, (c) the adversary forces all traffic from A to B to pass through the link n_i, n_j.

Consider a jammer that attacks only RREQ messages, in order to partition the network or to route traffic through specific paths. In Figure 1(a), we show how the adversary can jam RREQ messages so that node n_i is unable to discover routes to any destination. In Figure 1(b), the adversary deploys jamming devices along a cut in the network which partition the network into two components A and B. In Figure 1(c), the jamming attack is intended to divert traffic to a particular link, thus forcing all traffic from A to B to flow through that link. This attack is similar to the sinkhole attack [12], in which a node attracts surrounding traffic by advertising the shortest route to a particular set of destinations. Once traffic is diverted, the adversary can control the flow of traffic from A to B by compromising a single node (either n_i or n_j). *Note that a jamming attack that targets the routing function cannot be prevented by existing secure routing protocols, as such protocols consider jamming outside the scope of their adversarial model* [11]. Similar intelligent attacks against the control channel can be launched on critical network functionalities in other protocol layers. For example, the adversary may choose to jam the request-to-send (RTS) and clear-to-send (CTS) messages at the MAC layer so that the medium access delay is significantly increased. The critical vulnerability in all described attacks is the fact that although the network is multi-channel, nodes use a preassigned common channel to exchange control messages.

1.2 Our Contributions

In this paper, we address the problem of control-channel jamming in multi-channel wireless ad hoc networks. We consider a sophisticated adversary who exploits knowledge of protocol mechanics along with cryptographic quantities extracted from compromised nodes to maximize the impact of his attack in higher layers. New security metrics are defined that quantify the adversary's ability to localize and deny legitimate nodes access to the control channel. We develop a randomized distributed channel establishment scheme that allows nodes to establish a new control channel using frequency hopping. Under our scheme, network nodes are able to temporarily construct a control channel until the jammer is removed from the network. Our scheme differs from classical frequency hopping in that the communicating nodes are not synchronized on the same hopping sequence, but each node follows a unique hopping sequence. This leads to unique identification of the set of compromised nodes by nearby nodes. Assuming perfect random sequence generators, we analytically evaluate the expected delay until a control channel is re-established and the expected fraction of time that the control channel is available. We verify our analytic results via extensive simulations.

The remainder of this paper is organized as follows: In Section 2 we present related work. In Section 3 we present network and adversarial models, state our problem, and define new security metrics for the jamming attack. Section 4, describes our proposed control channel architecture. In Section 5, we present our randomized distributed scheme for re-establishing the control channel. The analytical evaluation of the performance of our scheme is presented in Section 7. In Section 8, we summarize our contributions.

2. RELATED WORK

When treated only as a physical-layer attack, jamming is often mitigated by employing spread spectrum techniques [2, 15]. If the pseudo-random noise (PN) sequence used to spread the information is kept secret, the adversary has to expend a disproportionate amount of energy compared to the sender to successfully jam a signal. The typical processing gain of anti-jamming techniques utilizing spread spectrum communications is in the range of 20 to 30 dB [2, 15].

Unfortunately, spread spectrum communications can provide anti-jamming protection only to the extent that the PN sequence remains secret. Once this sequence is revealed, a jamming adversary can deny communications by using very little energy. This can be particularly devastating for the control channel, which is by design a broadcast channel. Hence, a large number of nodes are synchronized to the same sequence. The compromise of any node reveals the control-channel hopping sequence to the adversary.

The problem of control channel jamming in the presence of node compromise was previously addressed in the context of GSM networks [4, 5]. In such networks, multiple control channels are implemented over specified frequency bands and time slots, so that any subscriber can listen to them. Chan et. al. proposed the replication of control information over multiple control channels according to a binary encoding based key (BBK) assignment [4]. Assuming that the adversary can jam only one channel in a given time slot, the authors derived the necessary conditions to guarantee control channel access to all users within a period of sev-

eral slots. They also showed that the BBK assignment leads to the identification of a threshold number of compromised nodes. Tague et. al. proposed the probabilistic assignment of cryptographic keys to network users so that nodes can discover the location of the control channels with certain probability. Their method allows for graceful degradation in the control channel secrecy as a function of the number of compromised nodes, as opposed to the threshold approach in [4]. Both methods in [4, 5] consider a server-client model where base stations are assumed to be secure.

The use of jamming to impact higher-layer functionalities was studied in [3, 16, 17, 19–23, 25]. Xu et. al. addressed the problem of detecting physical-layer and MAC-layer DoS attacks [17]. They proposed frequency hopping to avoid jamming, but assumed that all cryptographic quantities are secure (no node compromise) [17]. They also proposed the use of spatial retreats to avoid communication within the jammed area. Formal measures for detecting jamming attacks in wireless networks were introduced in [18]. Xu et. al. also proposed a timing-based low bit rate covert channel in the presence of jamming [24]. This channel maps the inter-arrival times of corrupted packets into bits.

Liu et. al. proposed an architecture called SPREAD to mitigate the impact of smart jammers that target multiple layers [20]. SPREAD alternates between a multitude of protocols at each layer, thus increasing the uncertainty of the adversary with respect to the protocol mechanics. Cagalj et. al. proposed wormhole-based anti-jamming techniques for sensor networks [21]. Using a wormhole link, sensors within a jammed region establish communications outside the jammed area to notify the network operator of the presence of a jammer. Wood et. al. studied proactive techniques for detecting and mapping jammed areas in sensor networks [25]. McCune et. al. proposed methods for detecting DoS attacks against sensor network broadcasts [23]. Finally, Li. et. al. provided a game theoretic formulation for optimal jamming and anti-jamming strategies at the MAC layer in wireless sensor networks [19].

Strasser et. al. proposed an uncoordinated frequency hopping (UFH) scheme for establishing shared secret keys between devices that do not share any prior secrets in the presence of a jammer [26]. The common principle between UFH and our scheme is that nodes do not hop using the same hopping sequence. However, in our scheme, hopping sequences are designed to implement the control channel, while in UFH hopping sequence are purely random. Slater et. al. improved the communication efficiency of the UFH scheme using Merkle trees, distillation codes, and erasure coding [27].

3. MODEL ASSUMPTIONS AND PROBLEM STATEMENT

3.1 Network Model

We consider a multi-channel ad hoc network that operates over K orthogonal frequency channels. We will use the terms frequency, frequency channel, or simply channel interchangeably to denote a frequency band over which communication takes place. Each node is equipped with a single half-duplex transceiver. Hence, a node can only listen to or transmit over one channel at a time. We further consider a time-slotted system for communication. Network nodes are assumed to be capable of slowly hopping between different frequencies. For simplicity, we assume that one frequency hop can occur at every time slot. Moreover, several messages may be exchanged during one slot, i.e., the time that a host resides on one frequency can be larger than the duration of one message.

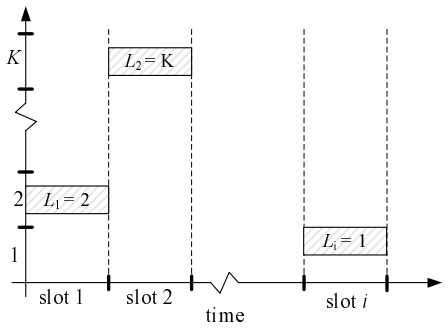

Figure 2: The location L_i of the control channel on slot i is defined by the frequency f_i. The control channel is dynamically allocated in orthogonal frequencies over different time slots.

The network uses a broadcast control channel. A message sent over this channel can be heard by any node within the communication range of the transmitting node. The control channel "location" L_i is defined as the frequency $f_i \in \{1, ..., K\}$ used for control-channel access at slot i. In Figure 2, we show the dynamic location of the control channel over time and frequency.

To facilitate coordination, the network is assumed to be clustered with each cluster having a clusterhead (CH). Several methods are available for node clustering [9] and CH selection [28] in wireless ad hoc networks. Clustering and CH selection issues have also been considered in a hostile environment [29–31]. We assume that prior trust has been established between the nodes of the network using one of several available methods [8, 32, 33]. The integrity, confidentiality, and freshness of communications within a cluster is preserved using appropriate cryptographic methods. These methods either employ symmetric cryptography, or use resource-efficient asymmetric cryptography. For example, the CH can share a cluster key with all nodes in a cluster and a pairwise key with each node individually [8]. Alternatively, elliptic curve cryptography-based encryption and digital signatures can be used to preserve the confidentiality and integrity of the exchanged messages [34].

3.2 Adversarial Model

The goal of the adversary is to deny the availability of the control channel for the maximum possible period of time. In order to do so, the adversary is capable of jamming a single frequency in any given slot over a communication range R_{max}, with the ability of switching frequencies at every slot. All messages received by any node within the range of the jamming device at the jammed frequency are assumed to be "irrecoverably" corrupted, i.e., corrupted messages cannot be recovered using error correction techniques [35]. Network nodes are assumed capable of detecting the jamming attack if they are within the range R_{max} of the jammer and are tuned to the frequency blocked by the jammer. Several methods are available for jamming detection [18], and

any of them can be used for our purposes. We further assume that the adversary can physically compromise network devices and recover *all content of their memory,* including cryptographic quantities and channel hopping sequences.

Let N be the number of nodes using frequency f_i as the control channel and that are located within the communication range R_{max} of a jammer. Suppose the jammer blocks frequency f_i. We address the problems of: (a) re-establishing the control channel in the absence of any coordination channel and in the presence of node compromise, and (b) identifying the compromised node(s).

The problems addressed are particularly challenging because the jamming attack is no longer treated as an outside attack on the physical layer. Instead, the adversary is assumed to be capable of obtaining necessary cryptographic quantities via node compromise. Furthermore, the adversary can exploit any information recovered from compromised nodes to predict the future locations of the control channel, thus denying access to it.

3.3 Anti-Jamming Metrics

Several metrics were previously proposed to evaluate the effectiveness of a jammer in impacting the throughput of the network. Xu et. al. introduced the metrics of packet send ratio (PSR) and packet delivery ratio (PDR). The PSR is defined as the ratio of successfully sent packets, over the number of packets intended to be sent at the MAC layer. The PDR is defined as the ratio of successfully delivered packets over the number of sent packets [18]. However, these metrics are not representative of the effectiveness of a sophisticated jammer that only targets the control channel. For example, even if the PDR is low, the jammer may be successful in redirecting traffic over a link of his own choosing, as we showed in Figure 1(c). To capture the effectiveness of the adversary in denying the control channel, we define the following security metrics.

DEFINITION 1. **Evasion Entropy E_i**–*Let X_i be a random variable that denotes the location of the control channel at slot i. We define the evasion entropy as:*

$$E_i = H(X_i|X_{i-1}, X_{i-2}, \dots X_0)$$

where $H(X|Y)$ is the conditional entropy of random variable X given random variable Y :

$$H(X|Y) \stackrel{def}{=} \sum_x \sum_y \Pr[x] \Pr[x|y] \log_2 \Pr[x|y]$$

where by $\Pr[x]$ we abbreviate the probability $\Pr[X = x]$ and by $\Pr[x|y]$ we abbreviate the conditional probability $\Pr[X = x|Y = y]$.

The evasion entropy characterizes the capability of the adversary to successfully determine the future location of the control channel given all previously observed locations and any knowledge that it may have acquired due to node compromise. In the worst case, the adversary can deterministically predict the location of the control channel ($E_i = 0$), while in the best case, all previous information is independent of the future control channel location ($E_i = H(X_i)$).

DEFINITION 2. **Evasion Delay D**–*The evasion delay is defined as the time between the successful jamming of the control channel and the re-establishment of a new one.*

DEFINITION 3. **Evasion Ratio ER**–*The evasion ratio is defined as the fraction of time that the control channel is available for communication, in the presence of the jammer.*

ER indicates the overall success of the adversary in blocking the control channel over a period T of interest. Note that such blocking occurs not only due to jamming, but also due to the delay in re-establishing the control channel.

4. CONTROL CHANNEL ARCHITECTURE

Control messages in wireless ad hoc networks are either intended to the one hop neighborhood (e.g., RTS/CTS messages), or are intended to several neighborhoods (e.g., RREQ messages), in which case they are relayed in a multi-hop fashion. In both cases, there is no particular benefit to use the same channel for control in all one-hop neighborhoods of the network. In fact, allocating different control channel to adjacent neighborhoods increases the control channel throughput due to the reduction in interference between such neighborhoods. Moreover, allocating the control channel on a single frequency has the following significant disadvantages in case of a jamming attack: (a) a single long-range transmission can jam the control channel in multiple neighborhoods, (b) the control channel re-establishment process has to be coordinated network wide, thus incurring significant resource overhead and delay, and (c) even if spread spectrum communications are used, the compromise of a single node reveals the commonly used PN sequence.

The impact of long-range jamming attacks can be significantly reduced by varying the spatial and temporal frequency allocation of the control channel. Such a design would also reduce the delay and communication overhead of the control channel re-establishment process, since it requires only local coordination. To mitigate the impact of jamming, we adopt a *dynamic* control channel allocation strategy, whereby each cluster establishes and maintains its own control channel. In this design, it is sufficient to ensure that different clusters in the network can receive broadcast control messages from members of their own cluster, and that nodes at the border of multiple clusters are aware of the multiple control channels used.

In Figure 3(a), we show the implementation of the control channel using a single frequency. All nodes within the range of the jammer are denied access to the control channel. In Figure 3(b), we show the use of multiple frequencies to implement the control channel. CHs are responsible for the establishment and maintenance of the control channel within their clusters. The impact of the jammer is now confined to clusters within R_{max} that use the jammed frequency. We now describe the process of maintaining the control channel within a cluster, in the presence of a jammer.

5. CONTROL CHANNEL MAINTENANCE

Consider a single cluster with each node being within one hop from the CH. Suppose the current control channel is jammed by an adversary. The main idea behind our scheme is to have each node of the cluster hop between channels in a pseudo-random fashion, following a unique hopping sequence not known to other nodes. This way if the jammer captures the hopping sequence of a compromised node, this node can be uniquely identified. Once the compromised

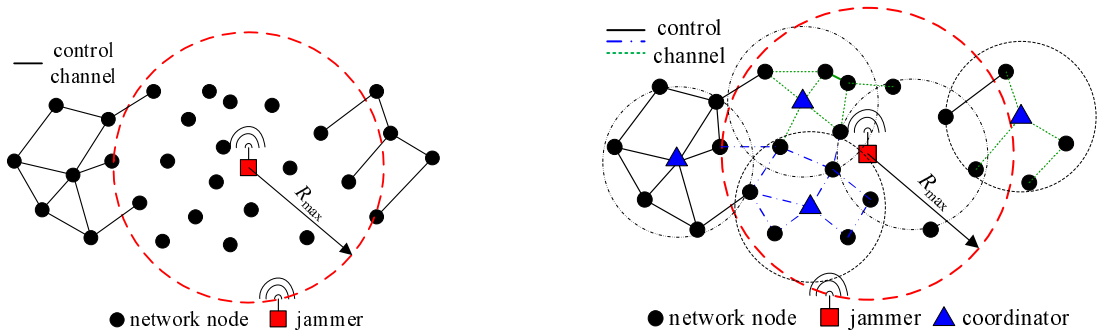

Figure 3: (a) The adversary blocks all control messages within R_{max} by jamming the control channel located on a single frequency band, (b) The control channel is allocated in different frequencies within each cluster. The impact of the jammer is now confined to those clusters within R_{max} that use the jammed frequency.

node has been identified, the CH updates the hopping sequences of all nodes in the cluster except the compromised one. Hence, the effectiveness of a jammer that exploits knowledge from compromised nodes becomes equivalent to the effectiveness of a jammer that randomly hops between channels. Note that our method is not a permanent solution for the control channel allocation, nor can it permanently be used for data communications due to its high communication overhead and delay. Rather, our scheme temporarily restores a control channel until the jammer and any compromised nodes are removed from the network.

The hopping sequences assigned to various cluster nodes are specially designed so these nodes overlap in frequency only a fraction of time, thus implementing the broadcast control channel. However, the slots where the control channel is located are not revealed to the cluster nodes. Given the uncertainty in the control channel location, control information is replicated in multiple slots until the control channel is accessed. Our scheme consists of three phases: (a) generation of hopping sequences, (b) assignment of hopping sequences, and (c) update of hopping sequences.

5.1 Generation of Hopping Sequences

By design, the hopping sequences assigned to nodes overlap only in a pre-defined number of slots, thus implementing the broadcast control channel in the cluster. In order to protect the secrecy of the control channel, the hopping sequences must satisfy the following properties: (a) *high invasion entropy* E_i; knowledge of previous hops does not reveal any information about future ones, (b) *independence*; knowledge of one sequence does not reveal any information regarding another, (c) *high minimum Hamming distance*; when interpreted as codewords, the sequences should have a high Hamming distance so that a compromised node can be uniquely identified. To construct a hopping sequence of length $L+M$, where M is the number of slots implementing the control channel, the following steps are executed:

1. Generate n random hopping sequences $s_j,\ 1 \le j \le n$, of length L, where n denotes the number of nodes in the cluster other than the CH. For each sequence $s_j = \{s_j(1), \ldots, s_j(L)\}$ where $s_j(i) \in \{1, \ldots, K\}$, we have $\Pr[s_j(i) = k] = \frac{1}{K}$.

2. Generate a random hopping sequence $c = \{c(1), \ldots, c(M)\}$, of length M, where $c \in \{1, \ldots, K\}$, and $\Pr[c(i) = k] = \frac{1}{K}$. Sequence c represents the control channel.

3. Generate a random position vector $v = \{v(1), \ldots, v(M)\}$, of length M, where $v \in \{1, \ldots, L+M\}$, and $\Pr[v(i) = k] = \frac{1}{L+M}$, with $v(i) \neq v(j)\ \forall i \neq j$.

4. Insert element $c(i)$ after element $s_j(v(i))$ on all sequences $s_j,\ 1 \le j \le n$ to generate new sequences $m_j,\ 1 \le j \le n$.

In Figure 4, we show an example of the generation of the hopping sequences for three nodes. In step 1, three sequences s_1, s_2, and s_3 of length $L = 12$ are generated, with $s_j(i) \in f_1, \ldots, f_8$. In step 2, a control-channel hopping sequence c of length $M = 5$ is generated with $c(i) \in f_1, \ldots . f_8$. In step 3, vector v indicates the five slots where the control channel is interleaved. In step 4, the sequences m_1, m_2, and m_3 are generated. They overlap in five slots which implement the control channel. Note that since the m_j's are generated by the random interleaving of the sequence c with the sequences $s_j,\ 1 \le j \le n$, *they are also random and independent of any other randomly generated sequence.* Note that the sequences $m_j,\ 1 \le j \le n$ are not independent of each other, since c is interleaved in specific slots on all sequences. In the following section, we describe the method of assigning the hopping sequence to each node in the cluster.

5.2 Hopping Sequence Assignment

The hopping sequences are generated by the CH and assigned to each node via secure pairwise communication, according to the cryptographic methods employed in the network. For example, if each node n_j shares a pairwise symmetric key K_{CH,n_j} with CH [8], the CH can protect the confidentiality of m_j by encrypting it with K_{CH,n_j}. Alternatively, if public key cryptography is employed, the CH can encrypt the sequence m_j with the public key of node n_j, allowing only node n_j that holds the corresponding private key to perform decryption.

Furthermore, the authenticity of the source must be ensured to prevent an adversary from assigning arbitrary sequences to the cluster nodes. In the case of symmetric key cryptography, knowledge of the symmetric key verifies the identity of the CH, assuming that the CH itself is not compromised. In the case of asymmetric cryptography, the CH must digitally sign each message containing a sequence m_i. The digital signature also verifies that the message has not

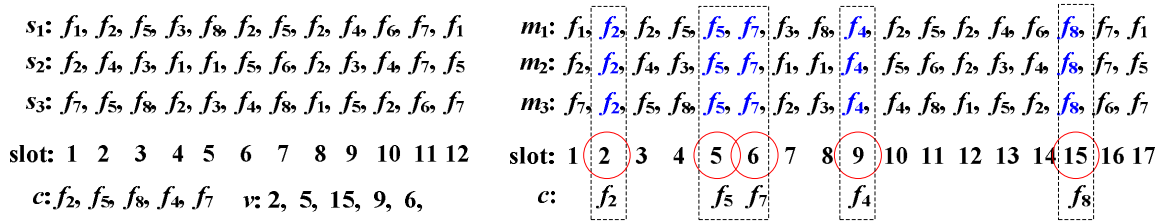

Figure 4: Hopping sequence generation for $L = 12$ and $K = 8$. The control channel sequence c is interleaved with the random sequences s_1, s_2, and s_3, at the locations indicated by the M-long vector v, leading to sequences of length $L + M = 17$. The control channel is implemented in a total of five slots.

been tampered with while in transit, thus guaranteeing message integrity. In the symmetric key case, message integrity can be ensured with the use of a Message Authentication Code [37]. Finally, the freshness of each message can be ensured with the use of a time stamp to avoid replays of old messages that can lead to de-synchronization of cluster nodes from the common control channel.

5.3 Hopping Sequence Update

The hopping sequences assigned to various nodes need to be updated either periodically or on demand. Such an update is needed because: (a) the hopping sequences have a finite period, (b) a node compromise may have taken place, and (c) the CH is rotated. Although in Section 5.1 we have assumed that the generated sequences are perfectly random, in practice PN sequences are used. It is well known that a PN sequence of length L and linear span S can be reconstructed using $2S$ consecutive known samples [38]. Hence, the hopping sequences need to be updated before the jamming adversary is able to reconstruct them. Note that the adversary can only monitor one channel at any one slot and cannot predict the channel location in the next hop. Hence, it is very hard to collect consecutive samples of a PN hopping sequence, unless a node is compromised.

Besides the need to periodically update the hopping sequences due to the finite linear span, the CH updates these sequences in case a node compromise has been detected. The CH engages in secure pairwise communication with each of the uncompromised nodes and assigns a new hopping sequence to each one. To update an uncompromised node n_j, the CH executes the following steps:

1. Synchronize with the hopping sequence of the uncompromised node n_j.
2. Assign a new sequence m'_j to node n_j, using frequency $m_j(i)$, where i is the current slot. If channel $m_j(i)$ is jammed, repeat until the assignment is confirmed.

Note that the adversary does not know the hopping sequences of uncompromised nodes and hence, it cannot prevent the assignment of new sequences. The case of CH compromise which reveals all hopping sequences to the adversary (and puts the CH under its control), is addressed through CH rotation, as detailed in Section 6.3. Once a CH rotation has occurred, the new CH updates the hopping sequences of all cluster nodes.

6. IDENTIFICATION OF COMPROMISED NODES

In this section, we describe how we can identify compromised nodes, based on their assigned hopping sequences.

6.1 Compromise of a Single Node

Once the control channel is denied, nodes start hopping according to their pre-assigned hopping sequences. However, if the jammer compromises one of the nodes n_j, it can obtain the corresponding hopping sequence m_j. By jamming frequencies according to m_j, the adversary can deny the control channel within the cluster, even if it does not know the slot locations of the control channel. However, following the unique sequence m_j reveals the identity of the compromised node n_j. The identification is based on the following proposition:

PROPOSITION 1. *The expected Hamming distance $E[d(m_j, m_\ell)]$ between two random sequences m_j and m_ℓ, as a function of their length X is*

$$E[d(m_j, m_\ell)] = \frac{K-1}{K} X. \quad (1)$$

PROOF. The proof is a direct consequence of the randomness and independence of the sequences m_j and m_ℓ. Based on the sequence generation process outlined in Section 5.1, $\Pr[m_j(i) = k] = \frac{1}{K}$, $\forall i$. Since the two sequences m_j, m_ℓ are assumed to be independent and random, they differ at slot i with probability,

$$\Pr[m_j(i) \neq m_\ell(i)] = \frac{K-1}{K}. \quad (2)$$

Equation (2) denotes the probability of success in a Bernoulli trial, where the success is defined as the event of rolling two K-faceted dice and obtaining a non-matching dice outcome. Let the Hamming distance between the two sequences be denoted by the random variable S. The event of having a Hamming distance equal to d is equivalent to having d successes in X trials, i.e., the Hamming distance is binomially distributed. Hence, the expected value of S is $E[S] = E[d(m_j, m_\ell)] = \frac{K-1}{K} X$. □

Proposition 1 states that the expected rate of increase of the Hamming distance between two random sequences is $\frac{K-1}{K}$. If the adversary has not compromised any node and is hopping according to a random sequence m_{jam}, the Hamming distance between m_{jam} and any of the assigned sequences m_j, $1 \leq j \leq n$ increases at the rate of $\frac{K-1}{K}$. On the other hand, if m_{jam} is a subset of the sequence m_j of a compromised node n_j, the Hamming distance between m_{jam} and m_j is expected to be significantly lower (although the adversary may be aware of m_j, he may choose to follow only a subset of it to avoid being identified). The CH can exploit this observation to identify the compromised node.

One of the challenges in identifying the compromised node, is the half-duplex limitation of the receiver. The CH can only listen to one channel at any slot. Since the CH is

Algorithm 1 Single Compromised Node Identification Scheme

Initialize :
$d(m_j, m_{jam}) = 0,\ \forall j$;
$j = 1$; $i = 0$; $CN \leftarrow \emptyset$
while J ==FALSE **do**
  for $x = 1,\ x \leq X,\ x++$ **do**
    if $m_j(i)$ NOT JAMMED **then**
      $d(m_j, m_{jam}) = d(m_j, m_{jam}) + 1$
    end if
    if $d(m_j, m_{jam}) < \frac{K-1}{K}x - \delta_x$ && $x > th$ **then**
      J =TRUE
      $CN \rightarrow n$
      break
    else
      $i++$
    end if
  end for
  if J ==TRUE **then**
    break
  else
    $j++$
  end if
end while
return CN

not aware of the hopping sequence m_{jam}, it cannot know in advance which frequencies to monitor. However, the CH is aware of all hopping sequences m_j of the nodes within its cluster. Hence, it can periodically listen to any one of them and compute their Hamming distance relative to the jammer's sequence. To do so, the CH need only know if channel $m_j(i)$ was jammed at each monitored slot i. If $m_{jam}(i) = m_j(i)$ the Hamming distance $d(m_j, m_{jam})$ remains the same. Otherwise, the Hamming distance increases by one. We now describe the steps for the identification of the compromised node. The pseudo-code of the identification scheme is shown in Algorithm 1.

1. Synchronize to the hopping sequence m_j of a randomly selected node n_j and initialize $d(m_j, m_{jam}) = 0$.

2. For each slot i, if $m_j(i)$ is not jammed, $d(m_j, m_{jam}) = d(m_j, m_{jam}) + 1$.

3. If $d(m_j, m_{jam}) < E[d(m_j, m_{jam})] - \delta_X$, after some sufficient number of slots $X > th$, node n_j is compromised.

4. Randomly pick another node n_ℓ and repeat steps 1-3 for a duration of X slots.

In Algorithm 1, if the computed Hamming distance falls below the expected Hamming distance, by a margin of δ_x then the compromised node has been successfully identified. The parameter δ_x is computed based on the variance of the Hamming distance given by $\frac{K-1}{K^2}X$, where X is the number of slots monitored. Note that this process is repeated for all nodes in a *random* round robin manner. When a node is monitored for a second time, the CH combines the results from the first monitoring round, to evaluate the status of the node.

6.2 Compromise of Multiple Nodes

When multiple nodes are compromised, the jammer can significantly reduce the number of slots jammed for denying the control channel, by jamming only the channel locations common to all compromised hopping sequences. Without loss of generality, assume that nodes $\{n_1, \ldots, n_q\}$ $q < n$ are compromised. The adversary can compute a new hopping sequence m_{jam} consisting of all the channel locations common to hopping sequences $m_1, \ldots, m_q$. The expected length of m_{jam} is given by Proposition 2.

PROPOSITION 2. *The expected length $E[X]$ of a sequence m_{jam}, consisting of the channel locations common to the compromised hopping sequences $m_1, \ldots, m_q$ is*

$$E[X] = M + \left(\frac{1}{K}\right)^q L. \tag{3}$$

PROOF. The q compromised sequences have at least M locations in common, corresponding to the M locations of the control channel sequence c, interleaved with each sequence $s_1, \ldots, s_q$. Furthermore, for the random sequences $s_1, \ldots, s_q$, the expected number of matching channel locations can be computed as follows. For a slot i, we can define the event that all q hopping sequences match as the outcome of a Bernoulli trial with probability of success equal to

$$\Pr[s_1(i) = s_2(i) = \ldots = s_q(i)] = \left(\frac{1}{K}\right)^q. \tag{4}$$

The number of successes in multiple independent slots is a repetition of independent Bernoulli trials which yields a random variable S that is binomially distributed. Hence, the expected number of successes, i.e., the expected number of matches in q randomly generated sequences is $\left(\frac{1}{K}\right)^q L$ yielding an overall expected length for m_{jam} as in (3). □

Proposition 2, shows that when q nodes are compromised, the adversary can reduce its overhead for denying the control channel by $\left(1 - \left(\frac{1}{K}\right)^q\right) L$ slots, on average. Note that the adversary cannot differentiate between the M locations that realize the control channel and the $\left(\frac{1}{K}\right)^q L$ locations that match due to the s_j's. Hence, the adversary must jam all channel locations common to the compromised sequences to deny access to the control channel.

To identify the compromised nodes, the CH correlates the known sequences m_j $1 \leq j \leq n$, to the control channel locations that are jammed. Let the CH follow a monitoring hopping sequence m_{CH} which is a concatenation of subsequences from the m_j's:

$$m_{CH} = m_1(1:t)||m_2(t+1:2t)||\ldots||m_n((n-1)t+1:nt),$$

where t is the time period that each node n_j is monitored. The CH keeps a matrix

$$A = \{a_{jk} | a_{jk} \in \{0,1\}\}_{J \times K}, \tag{5}$$

where each row j corresponds to node n_j and each column k corresponds to the k^{th} slot detected as jammed. Note that only slots of the s_j's are taken into account in the construction of matrix A. All locations of the control channel are ignored since, if jammed, they do not contribute to the identification of the compromised nodes. For the matrix A, $a_{jk} = 1$ if the k^{th} jammed slot is in m_j, and $a_{jk} = 0$ otherwise.

Algorithm 2 Multiple Compromised Nodes Identification Scheme

Initialize :
$A = 0,\ W = 0,\ j = 1,\ j$ =FALSE, $CN \leftarrow \emptyset$
v, n; //v vector of control channel slot positions
$m_{CH} = m_1(1:t)||m_2(t+1:2t)||\ldots||m_n((n-1)t+1:nt)$
while J ==FALSE **do**
  if $m_{CH}(i)$ JAMMED && $i \notin v$ **then**
    $a_{jk} = 1,\ W(j) = W(j) + 1\ \forall i,\ \exists\ m_{CH}(i) = m_j(i),\ i++$
  end if
  if $i > th_1$ // sufficient sampling **then**
    sort(W) // sort weights in descending order
    find $j,\ \exists\ W(A(j)) - E[W(A(j)) > \delta_q$
  end if
  if $W(1)$ AND $W(2)$ AND...AND $W(q) > \left(\frac{1}{k}^q qt\right) - \delta_q$ **then**
    $J = TRUE$, $CN = [n_1 \ldots n_q]$
  end if
end while
return CN

Considering each row $A(j)$ as a codeword, the CH computes the weight $W(A(j))$ of each codeword and ranks the weights in descending order. The weight $W(C)$ of a binary codeword C is defined as the number of ones in the codeword. The compromised nodes are expected to have a significantly larger weight than all other nodes and their weight will be of the same order. In fact, if q nodes are compromised, the expected weight for each codeword $A(j)$ is

$$E[W(A(j))] = \left(\frac{1}{K}\right)^q qt, \tag{6}$$

where we have computed $E[W(A(j))]$ over the qt slots that compromised nodes were monitored. The CH identifies the set of nodes with high weights and compares those weights with the expected weight as expressed in (6). If the weight of a codeword is larger than the expected weight by some margin δ_q, i.e., $W(A(j)) \geq \left(\frac{1}{K}\right)^q qt - \delta_q$, the corresponding node n_j is identified as compromised. The parameter δ_q is a tolerance margin related to the variance of $W(A(j))$. The pseudo-code for the identification of multiple compromised nodes is shown in Algorithm 2.

6.3 Compromise of the Clusterhead

If the adversary compromises CH, it obtains all sequences $s_j,\ 1 \leq j \leq n$, the corresponding sequences m_j as well as c and v. Hence, the adversary is aware of the hopping schedules of all nodes within the cluster. Using the knowledge of c and v, the adversary can deny control-channel access to all legitimate nodes by jamming only the control channel locations. To escape from this deadlock, the role of the CH is periodically rotated among the nodes within the cluster.

Alternatively, a CH rotation can be triggered if the control channel is denied for a prolonged period of time, indicating the compromise of the CH. In this case, cluster nodes start to randomly hop between channels, using self-generated random sequences. The new CH assigns a new hopping sequence to each node via secure pairwise communication. However, the CH is not aware of the random hopping sequences of the cluster nodes. Hence, the CH attempts to communicate with each node by also randomly hopping among the available channels. The steps of the hopping sequence assignment are as follows:

1. CH randomly hops between frequencies.
2. At each slot, the CH attempts to assign a new hopping sequence m'_j to a randomly selected node n_j, until n_j verifies reception of m'_j.
3. Step 2 is repeated until all cluster nodes are assigned new hopping sequences, except the previous CH.

In the performance analysis section, we analytically characterize the overall delay in case of a CH compromise, and also quantify criteria for identifying the compromise of the CH.

7. PERFORMANCE ANALYSIS

In this section, we analyze the performance of our proposed scheme with respect to the anti-jamming metrics introduced in Section 3.3.

7.1 Resistance Under No Node Compromise

Let us first examine the case where the jammer has not yet compromised any node, but tries to guess the location of the control channel. When no node has been compromised, knowledge of previous control channel locations does not reveal any information about future ones, due to the random hopping sequence generation. In this case, the evasion entropy for any slot i is equal to

$$E_i = H(X_i|X_{i-1}, X_{i-2}, \ldots X_0) = H(X_i) = \log_2 K \text{ bits.}$$

Note that so far we have not made any assumption about the jamming strategy of the adversary when no node has been compromised. Because $E_i = H(X_i)$, the adversary has no gain in favoring one channel over another. The jamming strategy that maximizes the probability of denying access to the control channel is expressed by the following proposition.

PROPOSITION 3. *When no node is compromised, the adversary maximizes the probability of denying access to the control channel by randomly hopping between channels, on every slot.*

PROOF. Since the hopping sequences are random, they form an i.i.d. process that follows the uniform distribution. Due to the independence of the channel location between slots, at any slot i there is an equal probability that a channel $k \in \{1, \ldots, K\}$ is selected. Hence, the adversary cannot increase its success in guessing the channel location more than the success of a random guess. In fact, based on the *asymptotic equipartition property*, the only sequences that are probable for a sequence length sufficiently large, are those sequences that belong to the typical set, i.e., those sequences that approach the uniform distribution in probability [39]. Hence, the adversary has the best chance in matching a random hopping sequence, by picking a sequence from the typical set, i.e., another random sequence. □

We now characterize the expected evasion delay $E[D]$, when the jammer follows a random hopping sequence.

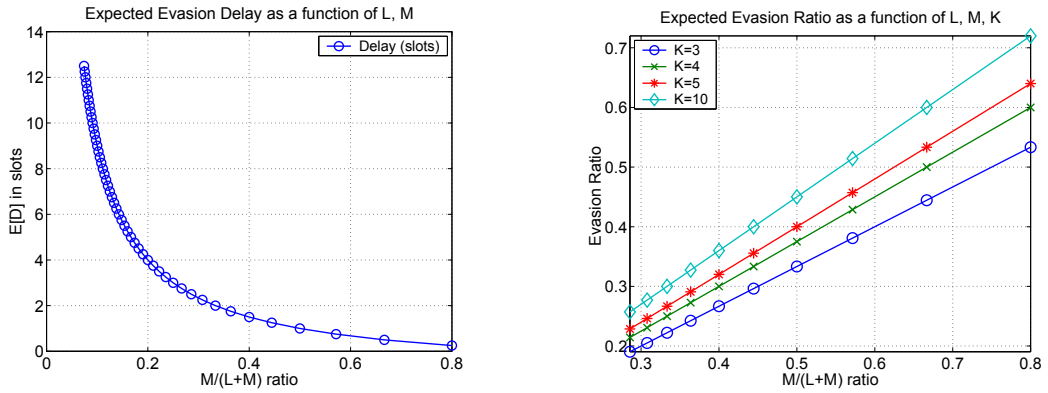

Figure 5: (a) The expected evasion delay as a function of the ratio $\frac{M}{L+M}$. As M increases, control channel slots occur more frequently, thus reducing the evasion delay, (b) ER as a function of $\frac{M}{L+M}$ for different values of K. The availability of a larger number of channels K decreases the ability of the jammer to guess the location of the control channel.

PROPOSITION 4. *The expected evasion delay $E[D]$ in re-establishing the control channel when no node has been compromised is:*

$$E[D] = \frac{1-p-(L+1)(1-p)^{L+1}+L(1-p)^{L+2}}{p}, \quad (7)$$

where $p = \frac{M}{L+M}$.

PROOF. Consider position vector $v = (v(1), \ldots, v(M))$ indicating the slot positions of the control channel. The delay until the control channel is re-established is equal to the lowest slot position in v. For each position $v(i)$, $\Pr[v(i) = k] = \frac{M}{L+M}$. The evasion delay is equal to one, if position one is not selected while position two is selected. Similarly, D is equal to two if positions one, and two are not selected but position three is selected. In the general case, the event $D = q$ occurs when the first q slots are not selected while slot $q+1$ is selected. This event occurs with probability:

$$\Pr[D = q] = \left(\frac{L}{L+M}\right)^q \left(\frac{M}{L+M}\right) = (1-p)^q p. \quad (8)$$

The evasion delay resembles the geometric probability with the distinct difference that it takes values only up to $D = L$, since in a sequence of finite length $(L+M)$ we are bound to have M slots assigned to the control channel. Computing the expectation of D yields:

$$\begin{aligned} E[D] &= \sum_{i=1}^{L} i \left(\frac{L}{L+M}\right)^i \left(\frac{M}{L+M}\right) \quad (9) \\ &= \left(\frac{M}{L+M}\right) \sum_{i=0}^{L} i \left(\frac{L}{L+M}\right)^i \\ &= \frac{1-p-(L+1)(1-p)^{L+1}+L(1-p^{L+2}}{p}. \end{aligned}$$

□

In a similar manner, we can calculate the expected delay for the re-occurrence of the control channel. In Figure 5(a), we show $E[D]$ as a function of p. We observe that as p increases, the control channel occurs more frequently and hence $E[D]$ decreases. We now compute the evasion ratio ER, for the case of a random jammer.

PROPOSITION 5. *The expected evasion ratio $E[ER]$ in the presence of a random jammer, when no node has been compromised is:*

$$E[ER] = \frac{M(K-1)}{K(L+M)}. \quad (10)$$

PROOF. Out of the $L+M$ frequency hops, only M of them implement the control channel. To evaluate ER, we are interested in determining the fraction of these M control channel locations that are not successfully jammed by a random jammer. Consider the random sequence $c = (c(1), \ldots, c(M))$ implementing the control channel, and let $c_{jam} = (m_{jam}(v(1)), \ldots, m_{jam}(v(M)))$ denote the hopping sub-sequence of the adversary for only the positions indicated by position vector v. Since c_{jam} is a sub-sequence of a random sequence, it is also random. The number of slots where the sequences c and c_{jam} do not match is equal to their Hamming distance $d(c, c_{jam})$. According to Proposition 1, the expected Hamming distance for two random sequences of length M is $\frac{K-1}{K}M$. Dividing this number by the total number of slots in each hopping sequence yields the expected value of the evasion ratio. □

Note that in the calculation of the evasion ratio, we have neglected the event of a common channel location to all sequences s_j, $1 \leq j \leq n$. This event occurs with probability $\left(\frac{1}{K}\right)^n$, leading to a total of $\left(\frac{1}{K}\right)^n L$ occurrences, which is negligible when K and n are sufficiently large. In Figure 5(b), we show the average value of the evasion ration $E[ER]$ as a function of the ratio $\frac{M}{L+M}$ and for varying values of the number of channels K. As the ratio $\frac{M}{L+M}$ increases, more control channel slots are available for the same value of L and hence, the adversary has a smaller probability of successfully jamming a particular slot. Furthermore, as K increases the probability of the adversary guessing the control channel decreases, leading to higher values of $E[ER]$.

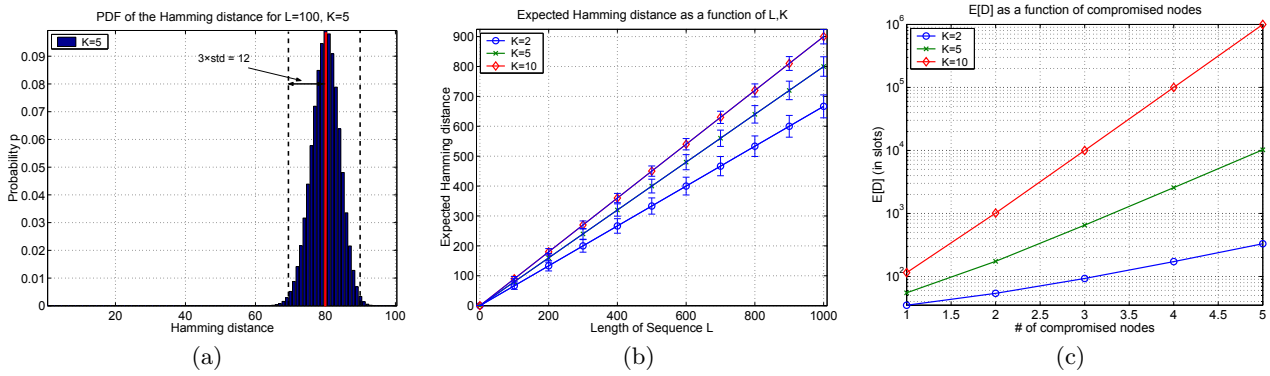

Figure 6: (a) The pmf of the Hamming distance between two random sequences of length 100, (b) the expected Hamming distance as a function of the sequence length for different K. Error margins denote 99% confidence intervals. The evasion delay as a function of the number of compromised nodes for varying K.

7.2 Resistance Under Node Compromise

All hopping sequences of the compromised nodes are revealed to the adversary. By following any of the compromised hopping sequences or the intersection of them, the adversary can deny access to the control channel to any legitimate node. Hence, under node compromise, the evasion entropy and the evasion ratio are equal to zero. The quantity of interest in this case is the expected evasion delay which is related to the delay until the set of compromised nodes is identified, and new sequences sequences are assigned to the set of uncompromised nodes in the cluster.

The CH identifies compromised nodes by matching the adversary's hopping sequence m_{jam} with the sequences of the compromised nodes. According to Algorithm 1, when a single node is compromised, the Hamming distance $d(m_j, m_{jam})$ from a compromised hopping sequence m_j is significantly smaller than $E(d(m_k, m_{jam})$, where n_k is an uncompromised node. In Figure 6(a), we show the pmf of the Hamming distance of a random sequence from any other random sequence. In Figure 6(b), we show the expected Hamming distance as a function of the length of the sequence and the set of available channels K, with the error margins denoting 99% confidence intervals. We observe that the event of node compromise can be easily identified since the expected Hamming distance between two random hopping sequences fall within very confined margins.

In the case of q compromised nodes, the weight of each compromised node is expected to be $\left(\frac{1}{K}\right)^q qt$, where t is the time that each of the q compromised nodes is monitored. Let the number of *jammed* slots required for the identification of the compromised nodes be th. The CH needs to monitor channels according to m_{CH} for an average time of $qt = K^q th$ slots, in order to observe th jammed ones. Upon identification of the compromised nodes, the CH must assign new hopping sequences to the remaining $(n-q)$ uncompromised nodes, yielding an additional delay of $(n-q)$ slots. Once sequences are assigned a delay equal to the first occurrence of the control channel under a random jammer, is incurred. The total expected evasion delay is:

$$E[D] = K^q th + (n-q) + \frac{1-p-(L+1)(1-p)^{L+1}+L(1-p)^{L+2}}{p}.$$

In Figure 6(c), we show the expected evasion delay as a function of the number of compromised nodes q for different values of K. As K and q increase $K^q th$ grows exponentially large, thus becoming the dominant factor in the evasion delay.To avoid such large values of delay, the CH can construct the hopping sequences using only a smaller subset of the available channels.

For evaluating the ability of the CH to identify the set of compromised nodes, we have simulated Algorithm 2, and compared the weights of the compromised nodes with those of the uncompromised ones. In Figure 7(a), we show the weight $W(A(j))$ of a single compromised node compared to the maximum weight of all uncompromised nodes, as a function of K, for a sequence length of $L = 5,000$. We observe that the compromised node has a consistently higher weight, compared to all uncompromised nodes.

In Figure 7(b), we show the percentage difference in weight between the compromised sequences and the maximum weight of all uncompromised ones, for different number of compromised nodes and for varying values of K. We observe that the weight of the compromised nodes is consistently higher than the maximum weight of all uncompromised nodes by at least 50%, allowing the identification of the set of nodes that are compromised.

7.3 Resistance Under CH Compromise

When the CH is compromised, the adversary knows the hopping schedules of all cluster nodes as well as the locations of the control channel. Hence, both the evasion entropy E_i, and the evasion ratio ER are equal to zero. The evasion delay D is equal to the sum of three components: (a) the time until the compromise of the CH is detected by cluster nodes, denoted by D_1, (b) the delay of assigning new hopping sequences to each cluster, denoted by D_2, and (c) the delay for re-establishing the control channel, denoted by D_3.

The cluster nodes consider the CH compromised when ER is below a threshold value for an extended period of time. The value of D_1 is the expected evasion delay in the case of node compromise, as calculated in the previous section, i.e., the required time for an uncompromised CH to re-establish the control channel when q nodes are compromised. D_1 can obtain large values for clusters with large number of nodes, if the number of compromised nodes is large. In such a case,

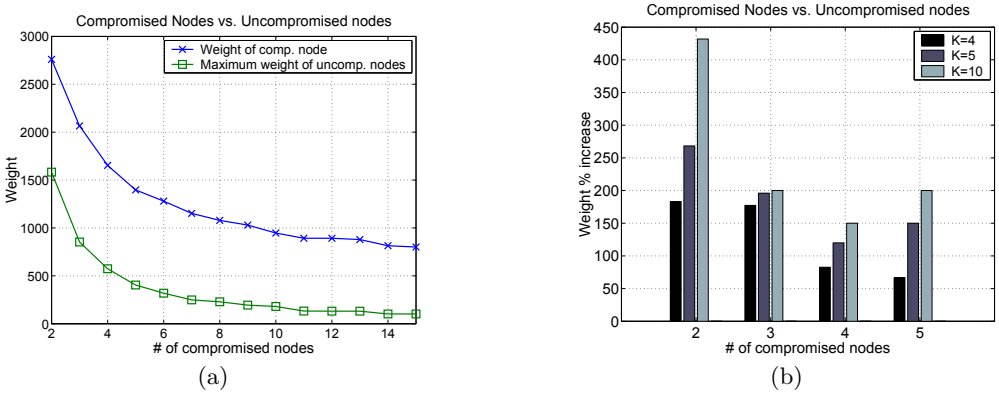

Figure 7: (a) The weight of compromised nodes compared to the maximum weight of uncompromised ones as a function of the number of compromised nodes, (b) the percentage weight increase when a node is compromised as a function of the number of compromised nodes, for varying K.

the delay D_1 hits an upper bound equal to the periodic CH rotation time. Once the CH is rotated, an additional delay D_2 occurs until the new CH is able to assign hopping sequences to all nodes. The expected delay $E[D_2]$ is given by the following proposition.

PROPOSITION 6. *The expected delay $E[D_2]$ until the new CH has assigned new hopping sequences to all cluster nodes is equal to:*

$$E[D_2] = \frac{K^2}{K-1}(n-1). \quad (11)$$

PROOF. After the CH rotation, each node is hopping according to a self-generated hopping sequence. Let m_j be the self-generated sequence of cluster node n_j, m_{CH} be the random hopping sequence of CH, and m_{jam} be the hopping sequence of the jammer. Let the CH attempt to communicate with node n_j, on consecutive slots. This process is a repetition of Bernoulli trials with probability of success equal to

$$\Pr[m_j = m_{CH}, m_j \neq m_{jam}] = \frac{1}{K}\frac{K-1}{K} = \frac{K-1}{K^2}, \quad (12)$$

The number of slots until the first success is geometrically distributed with an expected number of trials equal to $\frac{K^2}{K-1}$. The CH has to repeat the same process for all (n-1) cluster nodes (the compromised CH is excluded from the hopping sequence update process) and hence, the expected delay $E[D_2]$ until all cluster nodes have received a new hopping sequence is equal to $\frac{K^2}{K-1}(w-1)$. □

Once the new hopping sequences are assigned, the adversary's success in jamming the control channel becomes equivalent to that of a random jammer. An additional delay $E[D_3]$ is incurred until a slot implementing the control channel occurs in the new hopping sequences. This delay is equal to the evasion delay in the case of a random jammer. Delays $E[D_2], E[D_3]$ are negligible compared to the delay $E[D_1]$, since $E[D_1]$ is exponentially growing with the number of compromised nodes, while $E[D_2], E[D_3]$ are constant delays. Hence, the expected value of the evasion delay under CH compromise approximates the expected evasion delay as calculated in (11) for the maximum value of q. Again, $E[D_3]$ is upper-bounded be the period of the CH rotation, thus reducing the delay under large number of compromised nodes. Note that once a CH rotation is performed, the previous CH becomes a compromised node of the cluster and hence, can be identified by the new CH at a much shorter time.

8. CONCLUSION

We addressed the problem of control-channel jamming in multi-channel ad hoc networks, under node compromise. We proposed a randomized distributed channel establishment scheme that allows nodes to select a new control channel using frequency hopping. Our method differs from classical frequency hopping in that the communicating nodes are not synchronized to the same hopping sequence. Instead, each node follows a unique hopping sequence. We showed that our scheme can uniquely identify compromised nodes through their unique sequence and exclude them from the network. We evaluated the performance of our scheme based on the newly proposed metrics of evasion entropy, evasion delay, and evasion ratio. Our proposed scheme can by utilized as a temporary solution for the control channel re-establishment until the jammer and the compromised nodes are removed from the network.

Acknowledgments

This research was supported by the National Science Foundation (under grant CNS-0721935), BAE, Raytheon, and Connection One (an I/UCRC NSF/industry/university consortium). Any opinions, findings, conclusions, or recommendations expressed in this paper are those of the author(s) and do not necessarily reflect the views of NSF.

9. REFERENCES

[1] P. Kyasanur, and N. H. Vaidya, "Capacity of multi-channel Wireless Networks: Impact of Number of Channels and Interfaces," *in Proc. of MobiCom,* 2005.

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

[3] G. Noubir, and G. Lin, "Low Power DoS Attacks in Data Wireless LANs and Countermeasures," *in Proc. ACM MobiHoc,* 2003.

[4] A. Chan, X. Liu, G. Noubir, and B. Thapa, "Control Channel Jamming: Resilience and Identification of Traitors, *in Proc. of ISIT*, 2007.

[5] P. Tague, M. Li, and R. Poovendran, "Probabilistic Mitigation of Control Channel Jamming via Random Key Distribution," *in Proc. of PIMRC*, 2007.

[6] J. So, and N. H. Vaidya, "Multi-channel MAC for Ad Hoc Networks: Handling Multi-channel Hidden Terminals Using a Single Transceiver," *in Proc. of MobiHoc,* 2004.

[7] J. Mo, H. Wilson, and J. Walrand, "Comparison of Multi-channel MAC Protocols," *in Proc. of MSWiM,* 2005.

[8] S. Zhu, S. Setia, and S. Jajodia, "LEAP: Efficient Security Mechanisms for Large-scale Distributed Sensor Networks," *in Proc. of CCS*, pp. 62–72, 2003.

[9] S. Basagni, "Distributed Clustering for Ad Hoc Networks," *In Proc. of ISPAN,* 1999.

[10] S. D.M. Chatterjee, and D. Turgut, "WCA: A Weighted Clustering Algorithm for Mobile Ad Hoc Networks," *Journal of Cluster Computing (Special Issue on Mobile Ad hoc Networks),* 5(2), 2002.

[11] Y. C. Hu, D.B. Johnson, and A. Perrig, "SEAD: Secure Efficient Distance Vector Routing for Mobile Wireless Ad hoc Networks," *in Proc. of WMCSA,* 2002.

[12] C. Karlof, and D. Wagner, "Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures," *Elsevier's Ad Hoc Networks Journal, Special Issue on Sensor Network Applications and Protocols,* vol. 1 no. 2–3, pp. 293–315, 2003.

[13] C. E. Perkins, and E. M. Royer, "Ad hoc On-Demand Distance Vector Routing", *in Proc. of the 2nd IEEE Workshop on Mobile Computing Systems and Applications,* pp. 90–100, 1999.

[14] D.B. Johnson, D.A. Maltz and J. Broch, "The Dynamic Source Routing Protocol for Multihop Wireless Ad Hoc Networks," *Ad Hoc Networking, (Addison-Wesley, 2001),* ch. 5, pp. 139Ű-172, 2001.

[15] Bernard Sklar, Digital Communications, Fundamentals and Applications, Prentice-Hall, 2001.

[16] G. Lin, and G. Noubir, "On Link-layer Denial of Service in Data Wireless LANs," *Journal on Wireless Communications and Mobile Computing,* 2004.

[17] W. Xu, T. Wood, W. Trappe, and Y. Zhang, "Channel Surfing and Spatial Retreats: Defenses Against Wireless Denial of Service," *in Proc. of WiSe*, 2004.

[18] W. Xu, W. Trappe, Y. Zhang, and T. Wood, "The Feasibility of Launching and Detecting Jamming Attacks in Wireless Networks," *in Proc. of ACM MobiHoc,* pp. 46–57, 2005.

[19] M. Li, I. Koutsopoulos, and R. Poovendran, "Optimal Jamming Attacks and Network Defense Policies in Wireless Sensor Networks," *in Proc. of INFOCOM,* 2007.

[20] X. Liu, G. Noubir, R. Sundaram, and S. Tan, "Spread: Foiling Smart Jammers Using Multi-layer Agility," *INFOCOM Mini Symposium,* 2007.

[21] M. Cagalj, S. Capkun, and J.-P. Hubaux, "Wormhole-based Anti-jamming Techniques in Sensor Networks," *IEEE Trans. on Mobile Computing,* vol. 6, no. 1, pp. 100–114, Jan. 2007.

[22] Y. W. Law, L. van Hoesel, J. Doumen, P. Hartel, and P. Havinga, "Energy Efficient Link-layer Jamming Attacks Against Wireless Sensor Network MAC Protocols," *in Proc. of SASN,* 2005.

[23] J. M. McCune, E. Shi, A. Perrig, and M. K. Reiter, "Detection of Denial of Message Attacks on Sensor Network Broadcasts," *in Proc.of IEEE Symposium on Security and Privacy,* 2005.

[24] W. Xu, W. Trappe, Y. Zhang, "Anti-Jamming Timing Channels for Wireless Networks," *In Proc. of the 1st ACM Conference on Wireless Security (WiSec),* 2008.

[25] A. D. Wood, J. A. Stankovic, and S. H. Son, "JAM: A Jammed-Area Mapping Service for Sensor Networks," *in Proc. of RTSS,* 2003.

[26] M. Strasser, C. Pöpper, S. Capkun, and M. Cagalj, "Jamming-resistant Key Establishment using Uncoordinated Frequency Hopping," *in Proc. of IEEE Symposium on Security and Privacy, 2008.*

[27] D. Slater, P. Tague, R. Poovendran, and B. Matt, "A Coding-Theoretic Approach for Efficient Message Verification Over Unsecure Channels," to appear, WiSec, 2009.

[28] N. Malpani, J. L. Welch, and N. Vaidya, "Leader Election Algorithms for Mobile Ad Hoc Networks", *in Proc. of DIALM,* pp. 96–103, 2000.

[29] S. Vasudevan, B. DeCleene, N. Immerman, J. Kurose, D. Towsley, "Leader Election Algorithms for Wireless Ad Hoc Networks," *DARPA Information Survivability Conference and Exposition - Volume I,* 2003.

[30] D. Liu, "Resilient Cluster Formation for Sensor Networks," *in Proc. ICDCS*, 2007.

[31] K. Sun, P. Peng, P. Ning, and C. Wang, "Secure Distributed Cluster Formation in Wireless Sensor Networks," *in Proc. ACSAC,* 2006.

[32] N. Asokan, and P. Ginzboorg, "Key Agreement in Ad Hoc Networks," *Computer Communications,* vol. 23, no. 17, pp. 1627–1637, 2000.

[33] L. Eschenauer, and V.D. Gligor, "A Key-Management Scheme for Distributed Sensor Networks," *in Proc. of CCS,* pp. 41–47, 2002.

[34] W. Du, R. Wang, and P. Ning, "An Efficient Scheme for Authenticating Public Keys in Sensor Networks," *In Proc. of MobiHoc,* 2005.

[35] S. Lin, and D. Costello, Error Control Coding, 2nd edition, Pearson Prentice Hall, 2004.

[36] T. S. Rappaport, Wireless Communications, Principles and Practice, Prentice Hall, 1996.

[37] D. Stinson, Cryptogrpahy: Theory and Practice, 3rd edition, CRC Press, 2005.

[38] J. L. Massey, "Shift-register Synthesis and BCH Decoding," *in IEEE Transactions on Information Theory,* 1969.

[39] T. M. Cover, J. Thomas, Elements of Information Theory, 2nd Edition, Wiley, 1991.