USD-FH: Jamming-resistant Wireless Communication using Frequency Hopping with Uncoordinated Seed Disclosure

An Liu, Peng Ning, Huaiyu Dai, Yao Liu
North Carolina State University, Raleigh, NC 27695
{aliu3, pning, huaiyu_dai, yliu20}@ncsu.edu

***Abstract*—Spread spectrum techniques (e.g., Frequency Hopping (FH), Direct Sequence Spread Spectrum (DSSS)) have been widely used for anti-jamming wireless communications. Such techniques require that communicating devices agree on a shared secret before communication. However, it is non-trivial for two devices that do not share any secret to establish one in presence of a jammer. Recently, several schemes relying on Uncoordinated Frequency Hopping (UFH) were proposed to allow two devices to establish a secret key using Diffie-Hellman (DH) key establishment protocol in presence of jammers. Unfortunately, all these schemes are limited in efficiency.**

In this paper, we propose a novel scheme named *USD-FH*, which uses Uncoordinated Seed Disclosure in Frequency Hopping to establish a shared secret in presence of jammers. The basic idea is to transmit each DH key establishment message using a one-time pseudo-random hopping pattern and disclose the corresponding seed in an uncoordinated manner before the actual message. Due to the large number of channels available for wireless communication, the jammers cannot control all channels at the same time. When the receiver and the sender use the same channel during seed disclosure, the receiver can get the seed. If the jammer does not listen on the same channel (and thus it does not know the hopping pattern), the receiver can receive the actual message without being jammed. We validate USD-FH through both theoretical analysis and simulation. Our results show that USD-FH is much more efficient and robust than previous solutions.

***Keywords*-Anti-jamming Wireless Communication; Frequency Hopping; Spread Spectrum**

I. Introduction

Spread spectrum techniques such as Frequency Hopping (FH) and Direct Sequence Spread Spectrum (DSSS) are widely used to mitigate jamming attacks [1]. These techniques use transmission patterns known to both sender and receiver but unknown to jammers to achieve anti-jamming capability. However, they all rely on a *secret* shared by sender and receiver to control the transmission pattern. Without such a shared secret, it is impossible to establish anti-jamming communication between sender and receiver.

When there are jammers, it is critical to allow a sender and a receiver that do not share any secret to establish one so that they can use either FH or DSSS for jamming-resistant communication. To address this problem, a scheme named Uncoordinated Frequency Hopping (UFH) [2] and several variations [3], [4] have been proposed to establish a shared secret between a sender and a receiver using Diffie-Hellman (DH) key establishment protocol before the FH communication starts.

Unfortunately, all these schemes share a common limitation: They have to split each DH message into multiple packets due to the constraint of packet size, which is determined by the hop duration and the bit rate. Due to the need to reassemble these packets into meaningful DH messages, each of these packets has to include additional fields and thus cannot be shorter than a certain length. This feature makes all these schemes vulnerable to responsive jamming attacks, which sense and jam channels with meaningful signals reactively. As a result, it takes a long time (and sometimes it is impossible) for these schemes to finish a DH key establishment protocol in presence of jammers.

In this paper, we propose a novel, efficient, and robust scheme named USD-FH, which uses Uncoordinated Seed Disclosure in Frequency Hopping to establish a shared secret in presence of jammers. The basic idea of USD-FH is to transmit each DH key establishment message using a one-time pseudo-random hopping pattern and disclose the seed of the pseudo-random hopping pattern in an uncoordinated manner before the actual message transmission. A key difference between USD-FH and previous solutions is that in USD-FH a DH key establishment message is transmitted in its entirety, while all previous solutions have to split a DH message into multiple pieces.

The rationale behind USD-FH can be explained as follows. Due to the large number of channels available for wireless communication, it is very difficult and extremely costly for a jammer to control all channels at the same time. Since neither the receiver nor the jammer knows the hopping pattern for the seed disclosure, there is always a chance that the receiver gets the seed of the pseudo-random hopping pattern while the jammer does not. As a result, the jammer cannot jam the message transmitted using the pseudo-random hopping pattern, while the receiver can receive the DH message correctly. To ensure that the receiver can get the DH message, each message is transmitted multiple times, each using a different pseudo-random hopping pattern.

We compare USD-FH with UFH and its variations through both theoretical analysis and simulation. Our results

This work is supported by the National Science Foundation under grants CNS-1016260, CAREER-0447761, and CNS-0721424. The contents of this paper do not necessarily reflect the position or the policies of the U.S. Government.

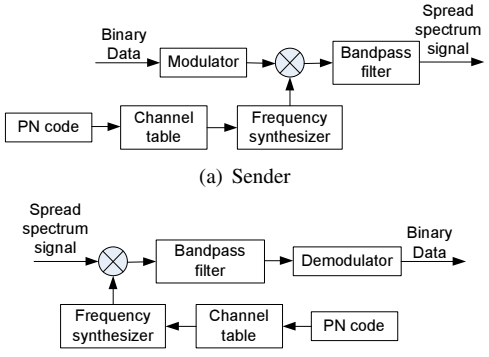
indicate USD-FH is much more efficient than UFH and its variations in presence of jammers. In particular, USD-FH can still complete DH key establishment protocol even when previous approaches all fail under powerful jamming attacks.

Our contribution in this paper is three-fold. First, we develop the USD-FH scheme, which allows a much more efficient and robust DH key establishment than previous solutions in presence of jammers. Second, we perform a systematic analysis of jamming probabilities under various jamming attacks for USD-FH. Finally, we provide a comparison of USD-FH with previous solutions through both theoretical analysis and simulation and show that USD-FH is much more efficient and robust than previous techniques.

The rest of this paper is organized as follows. Section II gives some background information on FH. Section III describes our assumptions and threat model. Section IV presents the proposed USD-FH scheme. Section V provides theoretical analysis of the performance of USD-FH and its jamming probability under various jamming attacks. Section VI compares USD-FH with previous solutions through both theoretical analysis and simulation. Section VII describes related work, and Section VIII concludes this paper.

II. Background

In FH, data transmission hops among a set of carrier frequencies according to a frequency hopping pattern (i.e., a sequence of channels). The frequency hopping pattern is typically pseudo-random, controlled by a secret key shared by the sender and the receiver. It is commonly referred to as *Pseudo-Noise (PN) code* in wireless communication. In FH, the receiver uses the PN code to hop among the correct sequence of channels to reconstruct the data bits.



(a) Sender

(b) Receiver

Figure 1. FH communication

Figure 1 shows the typical steps in FH communication. The sender generates carrier frequencies (channels) according to the PN code and the carrier frequency (channel) table. The sender then multiplies the modulated binary data with the carrier frequency, and passes through the bandpass filter before transmitting the signal.

The receiver first synchronizes with the sender and then decodes the received signals. It performs synchronization in two steps: FH acquisition (coarse synchronization) and FH tracking (fine synchronization) [5]. After synchronizing with the sender, the receiver generates carrier frequencies (channels) according to the PN code shared with the sender. By passing the signal through the bandpass filters at the right frequencies and demodulation, the receiver can get the binary data transmitted in the message.

III. Assumptions and Threat Model

Assumptions: We consider the following problem: Two nodes that do not share any secret want to establish one through a DH key establishment protocol or its Elliptic Curve Diffie-Hellman (ECDH) [6] variation in presence of jammers. Since the DH key establishment protocol is a mature protocol, we only focus on message transmission instead of the key establishment protocol itself.

We use the parameters used in the evaluation of UFH [2] to help the discussion and analysis of our scheme. (Apparently, this choice offers convenience when we compare the proposed USD-FH scheme with UFH and its variations.) We assume that the set of channels (C) on which transceivers can work is public. Each transceiver can simultaneously send and receive on c_n and c_m channels, respectively. Typically, a normal transceiver can only simultaneously send and receive on one channel (i.e., $c_n = c_m = 1$). An attacker can concurrently sense c_s channels. The required time to switch the frequency of a transmission (receiving) channel is t_s (t_j). The time to transmit a packet p_i on one channel is t_p. The minimum jamming period to jam the packet so that the packet cannot be decoded is $e \cdot t_p$, where $0 < e < 1$ is determined by the error correction code used in the packet.

We assume that the maximum bit rate (r_b) of a transceiver is fixed. For example, the bit rate of Bluetooth radio is up to 1 Mbps. In this paper, we use a pseudo-random number generator to generate the frequency hopping pattern, which controls how the transceiver hops among channels. We call the time duration that a transceiver stays in a channel as *hop duration* (i.e., t_p). In each hop, the transceiver can send or receive a packet. Thus, the packet size is determined by the hop duration when the bit rate is fixed, i.e., $t_p \cdot r_b$. The smaller the hop duration is, the smaller the packet size is.

Threats: Since we focus on message transmissions in presence of jammers, we only consider jamming attacks in this paper. We assume that the jammer's transmission power is limited. In other words, the jammer cannot jam all channels simultaneously. However, the jammer can jam a limited number (c_j) of channels at the same time, where $c_j < |C|$. The jammer may have powerful hardware (e.g., a large number of parallel bandpass filters) to quickly detect signals on c_s channels before jamming those channels. We also assume that the jammer can receive packets on multiple channels ($1 \leq c_{mj} < |C|$) at any point in time.

To help the reader, we summarize the notation discussed above in Table I.

Table I
NOTATION

C	the set of public channels
c_n	transceiver's # of channels for send operation
c_m	transceiver's # of channels for receive operation
c_{mj}	jammer's # of channels for receive operation
c_j	jammer's # of channels for jam operation
c_s	jammer's # of channels for sense operation
t_s	jammer's switch time of input channel
t_j	jammer's switch time of output channel
t_p	hop duration
$e \cdot t_p$	minimum jamming period for one packet
r_b	bit rate
x	# of hops for seed disclosure
l	# of packets in the DH message

IV. The Proposed USD-FH Scheme

The basic idea of USD-FH is to transmit each message multiple times *independently*, where each transmission uses a one-time pseudo-random hopping pattern in a *coordinated* fashion and discloses the seed of the pseudo-random hopping pattern in an *uncoordinated* manner before the actual message transmission. When expecting a message transmission from the sender, a receiver first hops at a much lower speed than the sender among the channels to receive the random seed. Once the receiver gets the seed, she can regenerate the pseudo-random hopping pattern and predict when the actual transmission will start. As a result, the receiver can use the regenerated hopping pattern to receive the message during its transmission.

The proposed USD-FH scheme is based on the following observation: Due to the large number of channels available for wireless communication, the jammer cannot control all channels at the same time. Even though the jammer can follow the same procedure as receivers to learn the random hopping pattern for the actual message transmission, since neither the receiver nor the jammer knows the hopping pattern for the seed disclosure, there is always a chance that the receiver gets the seed of the pseudo-random hopping pattern while the jammer does not. In this case, the jammer cannot jam the message transmitted using the pseudo-random hopping pattern, while the receiver can receive the transmitted message correctly.

Thus, USD-FH transmits each message multiple times, and each transmission uses a different pseudo-random hopping pattern. The jammer may have a good chance to jam each individual transmission. However, each receiver only needs to receive one message transmission correctly. The probability for the jammer to jam all transmissions decreases when the number of transmissions for each message increases. With a carefully configured parameter (e.g., determined with a conservative assumption of the jammer), all receivers will have a high probability to receive the message even in the presence of jammers.

A key difference between USD-FH and the previous solutions such as UFH [2] and BMA [3] is that in USD-FH though the random seed is transmitted in an uncoordinated way, the actual message is transmitted in a coordinated manner over very short hop durations. In contrast, all previous solutions transmit the actual message in an uncoordinated way, and thus have to handle the overhead to assist message assembly, such as the hash chain in UFH [2] and the cryptographic accumulator in BMA [3]. A direct consequence is that each hop duration cannot be too short. This makes all these schemes vulnerable to reactive jamming attacks, in which the jammer senses and jams the channels with communication activities reactively. As a result, it takes a long time (and sometimes impossible) for these schemes to finish transmitting a meaningful message (e.g., a DH key exchange message) in the presence of jammers. As we will show in our evaluation, our approach can achieve much better performance than these previous solutions.

In the following, we describe the sender and the receiver behaviors in USD-FH in detail.

A. Sender

For each message M, the sender repeats the transmission of M for a number of times. (The exact number of repeated transmissions can be determined through analysis in Section V.) Each transmission of M uses a pseudo-random hopping pattern generated by a different random seed, which is disclosed multiple times through another pseudo-random hopping pattern before the actual transmission of M. It is easy to see that different transmissions of M are independent of each other.

Now let us focus on how each transmission of M is performed in USD-FH. The sender divides message M into l segments: p_1, p_2, ..., p_l so that each segment can be transmitted in one hop duration (t_p). In other words, the sender prepares message M to be transmitted through frequency hopping, where each message segment is transmitted in one channel per hop. The sender then generates two random seeds s_1 and s_2, and uses them as inputs to a pseudo-random number generator ($PRNG$) to further generate two frequency hopping patterns fhs_1 and fhs_2, respectively. For the sake of presentation, we denote $fhs_1 = fs_1, fs_2, ..., fs_x$ and $fhs_2 = fm_1, fm_2, ..., fm_l$, where fs_i and fm_j ($1 \leq i \leq x$ and $1 \leq j \leq l$) are indexes to the channels in the public channel set C. These channels will be used in frequency hopping, where each channel is used in one hop. We use fhs_1 to determine the sequence of channels to disclose the seed s_2 and fhs_2 to determine the sequence of channels to actually transmit message M. Seed s_1 is only known to the sender and never disclosed.

Figure 2 illustrates this process. The sender uses fhs_1 to select x channels and discloses the seed s_2 in these x hops repeatedly. Due to the short length of a seed, we assume each hop is long enough to transmit a hop index and the entire seed s_2. The hop index is used to indicate the position of the current hop among all x hops to help the receiver synchronize with the sender; it is important for the receiver to determine the start of the actual transmission of message M. After x hops of seed disclosure (of s_2),

the sender uses fhs_2 to select a sequence of l channels to transmit l segments of message M.

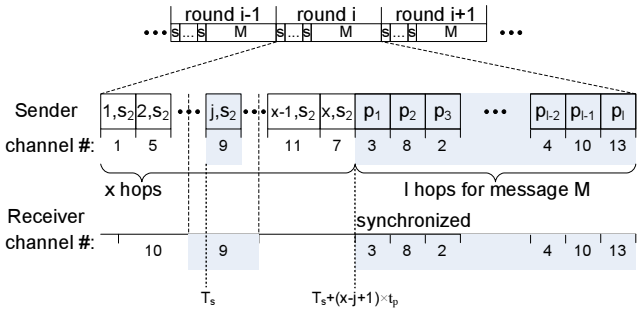


Figure 2. Message transmission using USD-FH

Since each seed is only used *once* to make the hopping pattern unpredictable only *before* the message transmission, seeds s_1 and s_2 do not have to be long. Having 32 bits in seeds s_1 and s_2 is sufficient to prevent the attacker from guessing the sequence of channels. To tolerate occasional losses of segment transmissions caused by the environment or the jammer's interference, the sender may optionally use Error Correcting Code (ECC).

B. Receiver

The receiver first attempts to receive the disclosed seed s_2. Specifically, the receiver randomly hops through all channels in the public channel set C at a much lower speed than the sender, as Figure 2 shows. Eventually, the receiver will have an overlapping channel with the sender (e.g., channel 9) from one of the first x hops and receive the disclosed seed s_2, which is used to generate the pseudo-random hopping pattern fhs_2 for message M. After getting the seed s_2 and its index number j at time T_s, the receiver calculates both the starting time of the message transmission for M and the pseudo-random hopping pattern fhs_2 to synchronize with the sender.

Since the sender discloses the seed s_2 and its index together, the receiver can easily estimate when the actual message transmission will start. As Figure 2 shows, when the receiver receives the disclosed seed s_2 in the j-th hop at time T_s, the starting time of the transmission of message M is $T_s + (x - j + 1) \times t_p$. Using the seed s_2 as the input of $PRNG$, the receiver can further generate the sender's pseudo-random hopping pattern fhs_2. As a result, the receiver knows precisely when to start receiving segments of message M on which channels throughout the frequency hopping process.

As mentioned earlier, the jammer may follow the same procedure as receivers to learn the pseudo-random hopping pattern and then jam the actual message transmission. However, as long as the jammer cannot listen on and jam all channels, the probability of a receiver to finally get the message will increase and approach 1 eventually as the number of transmissions of this message increases. More importantly, despite the multiple rounds of transmissions, USD-FH still achieves better performance than previous solutions. This is demonstrated through our analysis and experimental results in the following Sections.

V. Analysis

In this section, we analyze the performance of USD-FH in normal situations as well as its ability to deal with various jamming attacks. To facilitate the analysis, we classify possible jamming attacks against USD-FH into several categories, including low-level jamming attacks, high-level responsive jamming attacks, and hybrid jamming attacks, and perform analysis accordingly.

A. A Classification of Jamming Attacks

Low-level jamming attacks: We call jamming attacks that simply inject noise signal (either continuously or adaptively) to disrupt wireless communication *low-level jamming attacks*. Such jamming attacks are not specific to USD-FH, but applicable to any wireless communication scheme. There are two kinds of low-level jamming attacks, *non-responsive* jamming and *responsive* jamming attacks, as pointed out in [7]. A non-responsive jammer continuously jams the transmission without knowledge about the transmission at all. A responsive jammer detects the transmission and then launches jamming adaptively. For both non-responsive and responsive jamming attacks, the jammer can apply three strategies in each attack: *static*, *sweep*, and *random* strategies. In the static strategy, the jammer remains on the same jammed channel for a long time. In the sweep strategy, the jammer jams each channel for a period of time and hops through channels such that all channels will be jammed once after $\left\lceil \frac{|C|}{c_j} \right\rceil$ jamming cycles. In the random strategy, the jammer jams each channel for a period of time and changes the target channel randomly.

High-level responsive jamming attack: In addition to low-level jamming attacks, the jammer may launch *high-level responsive jamming attack*, in which the jammer attempts to follow the receiver's procedure to get the seed of the pseudo-random hopping pattern and then jam the message transmission.

Low-level jamming attacks and the high-level responsive jamming attack target at USD-FH from different angles. Low-level jamming attacks try to jam each packet transmission so that the packet transmitted in each hop cannot be recovered. The high-level responsive jamming attack attempts to catch the seed s_2 so that the whole message transmitted using the pseudo-random hopping pattern fhs_2 can be jammed.

Hybrid jamming attack: The jammer may certainly take advantage of both low-level jamming attacks and high-level responsive jamming attack. In other words, the jammer can launch *hybrid jamming attack*, in which she launches these low-level jamming attacks and high-level responsive jamming attack simultaneously.

In the following, we analyze the performance of USD-FH when there are no jamming attacks, in presence of low-level jamming attacks, high-level responsive jamming attacks, and hybrid jamming attacks. We mainly examine two performance metrics, the expected communication time required to transmit the message and the jamming probability.

Our analysis uses the notation discussed in Section III. Please refer to Table I for quick reference.

B. No Jamming Attacks

We first analyze the expected communication time required to transmit a message M (divided into l packets) when there is no jamming attacks. Following the analysis in [2], we can obtain the probability that the receiver can receive the seed s_2 and its hop index of one round message transmission as $p_s = 1-\left(\prod_{i=0}^{c_m-1}\left(1-\min\left\{\frac{c_n}{|C|-i},1\right\}\right)\right)^x$. Thus the probability that the message is successfully received in the i-th round transmission is $(1-p_s)^{i-1}p_s$. The expected number of packets transmitted is $N=\sum_{i=1}^{\infty}(1-p_s)^{i-1}\cdot p_s\cdot i\cdot(x+l)=\frac{x+l}{p_s}$. Considering the hop duration t_p, we can get the expected time to transmit a message as $T_{no}=\frac{x+l}{p_s}\cdot t_p$.

C. Low-level Jamming Attacks

Since low-level jamming attacks target the packet transmission instead of the whole message transmission, we first analyze the packet jamming probability. Then we analyze the jamming probability of the whole message transmission and the expected time of transmitting a message.

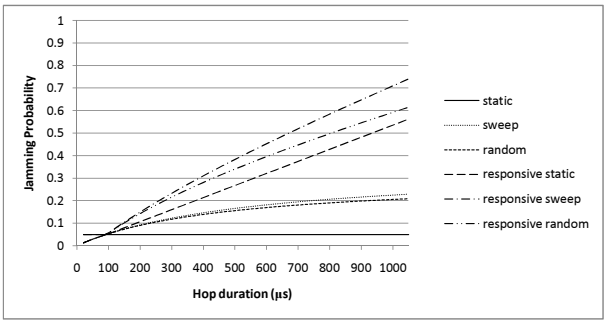
1) Packet Jamming Probability: The packet jamming probability can be easily derived based on the jamming performance equations in [2]. In the following, we examine the packet jamming probability in various situations.

Scenarios: We use four scenarios to examine and illustrate the capability of USD-FH against low-level jamming attacks, as shown in Figures 3 and 4. In these four scenarios, we set the total number of channels as $|C|=200$. We also use the Reed-Solomon error-correcting code that encodes 15-bit blocks into 21-bit blocks (to correct up to 3 bits error).

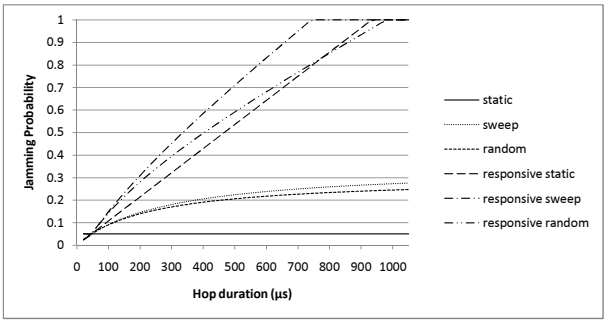
As Figure 3 shows, scenarios 1–3 are used to examine the relationship between the packet jamming probability and the hop duration. In scenarios 1–3, the jammer can simultaneously sense/jam 10, 10, and 20 channels, respectively. The time required for the jammer to switch channel for sensing/jamming operation is 80 μs, 40 μs, and 80 μs in Scenarios 1, 2, and 3, respectively.

Scenario 4 is used to examine the relationship between the packet jamming probability and the number of simultaneously jammed channels. We set $c_s=10$ and $t_j=t_s=40\,\mu$s for scenario 4. Figure 4 shows how the packet jamming probability is increased in scenario 4 when the number of simultaneously jammed channels (c_j) is increased for different hop durations (630 μs and 63 μs).

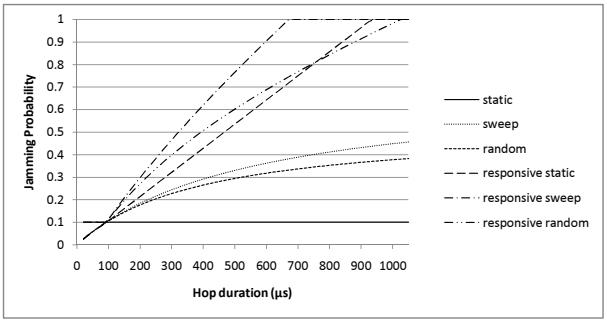
Short hop duration is preferred: From Figure 3 (scenarios 1–3), we can see that having short hop duration can



(a) Scenario 1 ($c_j=c_s=10$, $t_j=t_s=80\,\mu$s)



(b) Scenario 2 ($c_j=c_s=10$, $t_j=t_s=40\,\mu$s)



(c) Scenario 3 ($c_j=c_s=20$, $t_j=t_s=80\,\mu$s)

Figure 3. Jamming probability of low-level jamming attacks for different hop durations ($|C|=200$)

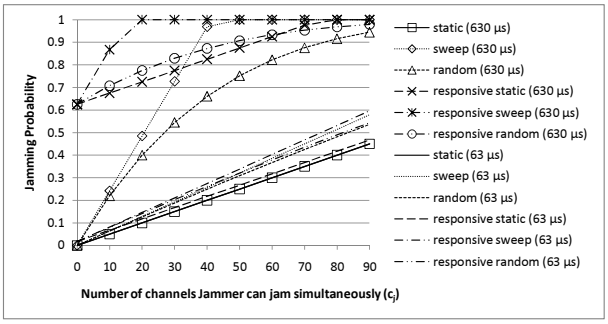


Figure 4. Scenario 4: Jamming probability of low-level jamming attacks for jammers who can jam c_j channels simultaneously ($|C|=200$, $c_s=10$, $t_j=t_s=40\,\mu$s, hop duration is 630 μs and 63 μs)

dramatically reduce the packet jamming probability of low-level responsive jamming attacks. If the hop duration is short enough, the packet jamming probability under the low-level responsive jamming attack is even lower than that under the low-level non-responsive jamming attack because the low-level responsive jammer does not have enough time to switch to multiple channels due to relatively large channel switch time (t_j). The jamming probability of one packet transmission is determined by hop duration (t_p) once other parameters ($|C|$, t_j, t_s, c_s, c_j) are fixed. The longer the hop duration is, the higher the jamming probability will be.

We also calculate the packet jamming probability for jammers with different jamming capabilities in Figure 4. From the figure, we can see that the jamming probability of the short hop duration (63 μs) increases much more slowly than that of the long hop duration (630 μs). Moreover, the jammer can get more advantage from the responsive jamming attack over the non-responsive jamming attack if the hop duration is longer. For short hop duration, the responsive jamming attack does not have much advantage over the non-responsive one. When the jammer can jam 20 channels simultaneously, the scheme with long hop duration (630 μs) will fail (i.e., always be jammed), while the scheme with the short hop duration (63 μs) still works.

USD-FH's support for short hop duration: Based on the above discussion, it is obvious that we need to reduce the hop duration as much as possible for frequency hopping based schemes. However, as discussed in the introduction, previous solutions such as UFH and its variations require additional fields for message assembly purpose, and thus it is impossible for them to shrink the hop duration to a desired size (e.g., 93 μs in Figure 3(a)). For example, UFH [2] and BMA [3] need 110 and 180 bits, respectively, in each packet for message assembly purpose as recommended by [3]. Given 1 Mbps bit rate (e.g., Bluetooth), hop durations of UFH and BMA must be larger than 110 μs and 180 μs, respectively.

Unlike UFH [2] and BMA [3], USD-FH does not need any additional field in each packet to assist the message assembly since all l packets following the seed disclosure belong to the same message. Thus the minimum hop duration of USD-FH depends on the size of the seed s_2 and the size of the hop index. Since the transmission time of message M is short (e.g., less than one second), we only need a short seed (e.g., 32 bits) to hold the attacker for a short period of time (e.g., several minutes), so that the jammer cannot guess the seed s_2 before the corresponding message transmission. Thus, the hop duration of USD-FH can be very short as long as the seed s_2 (e.g., 32 bits) and the hop index (e.g., 10 bits) can be transmitted in one packet. With such a short hop duration, USD-FH can effectively defeat low-level jamming attacks, especially low-level responsive jamming attacks.

2) Jamming Probability of Message Transmission: Given the packet jamming probability ($p_j \geq 0$) under low-level jamming attacks, we can compute the jamming probability of one round message transmission as follows. To jam one round message transmission, the jammer either jams all hops of the seed disclosure so that no receiver can get the seed s_2 and the hop index to synchronize with the sender, or jams enough packets of message M so that the message cannot be recovered.

The probability that the jammer can jam all x hops of the seed disclosure in one round is $p_{js} = p_j^x$. USD-FH can directly deploy ECC for message M to recover the message even the packet is lost. Assume the message is encoded using ECC to tolerate packet losses in up to ρl hops ($0 < \rho < 1$), the probability that the jammer can jam the message M is $p_{jm} = \sum_{i=0}^{\rho l - 1} \binom{l}{i} (1-p_j)^i p_j^{(l-i)}$. Thus the jamming probability of one round message transmission is $p_{jr\_low} = 1 - (1-p_{js})(1-p_{jm})$.

3) Expected Communication Time: The receiver can successfully receive the message when the seed is received and the message can be recovered. Thus the probability that the receiver can receive the message under low-level jamming attacks is $p_{low} = p_{s\_low} \cdot p_{m\_low}$, where p_{s_low} and p_{m_low} are probabilities to receive the seed and to recover the message, respectively. $p_{s\_low} = 1 - \left(\prod_{i=0}^{c_m - 1} \left(1 - \min\left\{\frac{c_n}{|C|-i}, 1\right\}(1-p_j)\right)\right)^x$ and $p_{m\_low} = \sum_{i=0}^{\rho l} \binom{l}{i} p_j^i (1-p_j)^{(l-i)}$, where p_j is the packet jamming probability.

The expected number of packets transmitted is $N_{low} = \sum_{i=1}^{\infty} (1-p_{low})^{i-1} \cdot p_{low} \cdot i \cdot (x+l) = \frac{x+l}{p_{low}}$. Thus the expected time to transmit a message is

$$T_{low} = \frac{x+l}{p_{low}} \cdot t_p. \quad (1)$$

Since $p_{s\_low} \leq p_s$ and $p_{m\_low} \leq 1$, $p_{low} \leq p_s$, and thus $T_{low} \geq T_{no}$. This means that low-level jamming attacks can slow down message transmission in USD-FH. However, as we will see in Section VI, USD-FH is much more resilient than the previous solutions in defending against low-level jamming attacks.

D. High-level Responsive Jamming Attack

Since the seed s_2 is disclosed before the transmission of message M, the jammer can also follow the receiver's procedure to get the seed s_2 and the hop index. Once the jammer gets the seed s_2 and the hop index for a round, she can synchronize with the sender (i.e., infer the starting time of M's transmission and the pseudo-random hopping pattern for M) and jam the transmission of M for that round.

Since we assume the jammer cannot receive on all channels, there is always a chance that in a round of message transmission, the receiver receives the seed s_2 and the hop index while the jammer does not. In such cases, the receiver can synchronize with the sender and receive message M without being jammed.

1) Jamming Probability of Message Transmission: When the jammer receives the seed s_2 and the hop index, she can synchronize with the sender and jam the transmission of

message M. Thus the jamming probability of the message transmission (i.e., the probability that the jammer receive the seed and the hop index) is

$$p_{jr\_high} = 1 - \left(\prod_{i=0}^{c_{mj}-1}\left(1 - \min\left\{\frac{c_n}{|C|-i}, 1\right\}\right)\right)^x,$$

where c_{mj} is the number of channels the jammer can receive simultaneously.

2) Expected Communication Time: The probability that the jammer does not receive the seed s_2 is $1-p_{jr\_high}$. The probability that the receiver receives the seed s_2 is p_s. So the probability that the receiver gets the seed while the jammer does not is $p_{high} = p_s(1-p_{jr\_high})$. Similar to calculating N_{low}, we can get the expected number of transmitted packets as $N_{high} = \frac{x+l}{p_{high}}$. Thus the expected communication time to transmit the message M is $T_{high} = \frac{x+l}{p_{high}} \cdot t_p$. Since $p_{high} \leq p_s$, $T_{high} \geq T_{no}$. This means that high-level responsive jamming attacks can slow down message transmission in USD-FH.

E. Hybrid Jamming Attack

To jam USD-FH more effectively, the jammer may launch hybrid jamming attacks, i.e., both low-level jamming attacks and high-level responsive jamming attack at the same time.

1) Jamming Probability of Message Transmission: In the hybrid jamming attack, one round of message transmission is jammed either by the low-level jamming attack, or the high-level responsive jamming attack or all of them. Thus the jamming probability of one round message transmission is $p_{jr\_hybrid} = 1 - (1-p_{jr\_low})(1-p_{jr\_high})$. Since $0 \leq p_{jr\_low} \leq 1$ and $0 \leq p_{jr\_high} \leq 1$, p_{jr_hybrid} is larger than or equal to both p_{jr_low} and p_{jr_high}. Thus the hybrid jamming attack is more effective than both low-level and high-level responsive jamming attacks.

2) Expected Communication Time: In the hybrid jamming attack, the receiver can successfully receive the message only when both the low-level jamming attacks and the high-level responsive jamming attack fail. Thus, the probability that the receiver gets the seed and successfully receives the message without being jammed is $p_{hybrid} = p_{low} \cdot p_{high}$. The expected number of packets required to transmit the message M successfully is then $N_{hybrid} = \frac{(x+l)}{p_{hybrid}}$, and the expected time to successfully transmit the message is

$$T_{hybrid} = \frac{(x+l)}{p_{hybrid}} \cdot t_p. \quad (2)$$

Since $p_{low} \leq 1$ and $p_{high} \leq 1$, we have $p_{hybrid} \leq p_{low}$ and $p_{hybrid} \leq p_{high}$. Thus T_{hybrid} is larger than or equal to both T_{low} and T_{high}. In other words, the hybrid jamming attack is the most effective jamming attack against the proposed USD-FH scheme.

VI. Comparison

To show the efficiency and robustness of USD-FH, we compare USD-FH with UFH [2] and BMA [3] through both theoretical analysis and simulation. Note that BMA is the most efficient variation of UFH, and represents the best existing solution. To be conservative, in the comparison with previous solutions, we use the hybrid jamming attack, which is the most effective attack against USD-FH as discussed in the previous section.

A. Setup

Scenarios: We compare the communication time of transmitting an ECDH message using UFH, BMA, and USD-FH in five scenarios under hybrid jamming attacks. We first construct four scenarios for the hybrid jamming attack against USD-FH based on the four old scenarios discussed in Section V-C1. For each old scenario, we pick the highest jamming probability of low-level jamming attacks and combine it with the jamming probability of the high-level responsive jammer who can receive on c_{mj} channel(s) to construct the corresponding new scenario for the hybrid jamming attack. In addition, we construct the fifth scenario by adjusting the number of channels on which the jammer can receive packet simultaneously.

Since UFH and BMA are not vulnerable to the high-level responsive jamming attack, new scenarios with the hybrid jamming attack are the same as the old ones discussed in Section V-C1 for UFH and BMA.

Theoretical analysis: For UFH and BMA, we use the highest jamming probability for low-level jamming attacks in each old scenario (shown in Figure 3 and Figure 4) to analyze the expected communication time based on equations in [2], [3]. For USD-FH under the hybrid jamming attack, we analyze the expected communication time based on equation (1) in Section V-E.

Simulation: In addition to theoretical analysis, we also use simulation to compare USD-FH with UFH and BMA. We implement the simulation code for UFH, BMA, and USD-FH based on the Bluetooth Frequency Hopping demo in Simulink [8]. Our simulation modulates signal using Gaussian Frequency Shift Keying (GFSK) over a radio channel with the maximum capacity of 1 Mbps. We set the total number of available channels as 200 so that we can do frequency hopping over a 200 MHz frequency range. We change the hop duration according to the packet size. To simulate the low-level jamming attack in each scenario, we exploit a Bernoulli Binary Generator to control the packet loss using a switch module according to the jamming probability of low-level jamming attacks in each scenario. To simulate the high-level responsive jamming attack, we introduce an attacker who follows the receiver's procedure and notifies the receiver to skip current round of message transmission once the attacker gets the seed and hop index.

Hop duration: In our evaluation, we use the fixed bit rate of 1 Mbps. We choose hop durations for UFH and BMA appropriately so that all encoded packets can fit in their hopping slots. For USD-FH, we choose the hop duration as $63\,\mu$s, which is small enough to defeat the low-level responsive jamming attack and large enough to contain the

seed s_2 and the hop index. For UFH and BMA, we try hop durations from 273 μs to 1050 μs for scenario 1, hop durations from 273 μs to 630 μs for scenario 2 and scenario 3, and hop duration 630 μs for scenario 4 and scenario 5.

Message format: Following [2], [3], we set the size of each ECDH message M as 2,176 bits, which consists of identity (64 bits), sender's public key (512 bits), sender's certification (512 bits), time stamp (64 bits), key contribution (512 bits), and the signature of the message (512 bits). For BMA and USD-FH, the message M is encoded into multiple packets using optimal erasure code [9] so that M can be reconstructed even if up to half of them are lost. (UFH does not use any forward error correction.) As discussed earlier, each packet is then transmitted during one hop duration.

Packet format: For each packet, we further use Reed-Solomon error correcting code that encodes 15-bit blocks into 21-bit blocks to correct up to 3 bits error. Following [2], each packet in UFH has a message id (34 bits), a packet id (6 bits), the payload (from 85 bits to 640 bits) and the hash value for next packet (70 bits). Following [3], each packet in BMA has a message id (34 bits), a packet id (6 bits), the payload (from 15 bits to 570 bits) and the witness (140 bits). We use the short term security level of 70 bits for hash links and 140 bits for the witness as recommended by BMA [3]. All fields other than the payload are necessary for message assembly purpose in UFH and BMA.

In contrast, USD-FH does not need additional fields for message assembly because all packets after the seed disclosure in one round belong to the same message; the whole packet (45 bits) can be used for the payload.

Given the above message and packet formats, the total number of packets for UFH and BMA range from 26 to 4 and from 292 to 8, respectively, when the hop duration changes. The total number of packets for USD-FH is 98.

Optimal x for USD-FH: In USD-FH, the parameter x (i.e., the number of hops for seed disclosure) is critical. The larger x is, the higher the probability of the receiver getting the seed s_2 and the hop index. However, the increasing of x also increases the number of hops per round message transmission ($x + l$), which further increases the expected communication time as well. The increasing of x also increases the jamming probability of message transmission under the high-level responsive jamming attack. Thus, we need to know the optimal x value that can minimize the expected communication time of transmitting M. In our analysis and simulation, we use Mathematica [10] to compute the optimal x which can minimize the expected communication time T_{hybrid} for USD-FH in five scenarios under the hybrid jamming attack.

B. Simulation and Analytical Results

Figures 5 to 9 show both simulation and analytical results in the five scenarios.

Scenarios 1–3: From Figures 5–7, we can see that when the hop duration is short, both UFH and BMA have long communication time because the total number of packets in UFH and BMA are very large when the number of available effective bits of each packet for payload is very small (excluding additional fields for message assembly purpose). We also notice that when the hop duration is long, both UFH and BMA have long communication time since the low-level jamming probability is high, as shown in Figure 3.

As Figure 3 shows in Section V, compared with scenar-

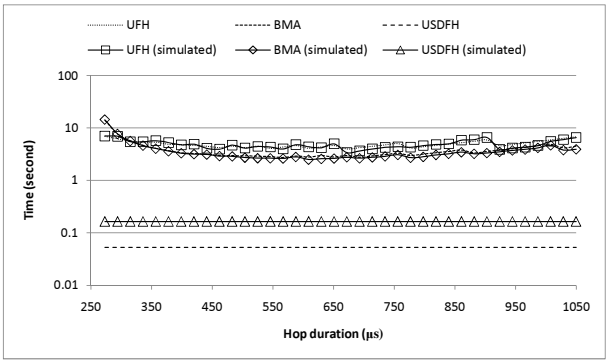


Figure 5. Communication time for scenario 1 under the hybrid jamming attack ($c_j = c_s = 10$, $t_j = t_s = 80\,\mu$s), hop duration for USD-FH is fixed as 63 μs, $x = 77$

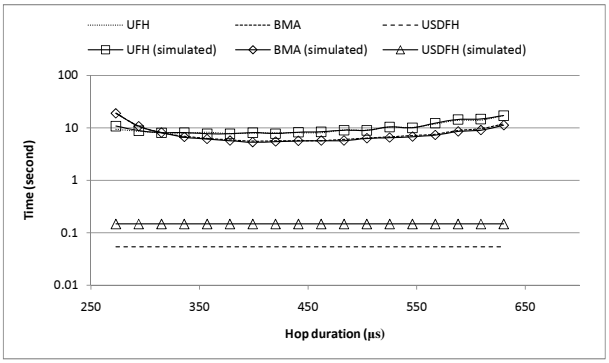


Figure 6. Communication time for scenario 2 under the hybrid jamming attack ($c_j = c_s = 10$, $t_j = t_s = 40\,\mu$s), hop duration for USD-FH is fixed as 63 μs, $x = 78$

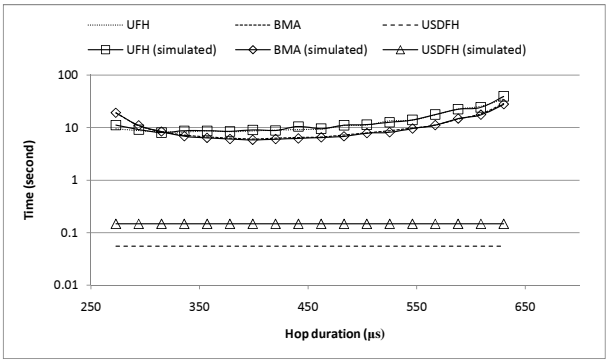


Figure 7. Communication time for scenario 3 under the hybrid jamming attack ($c_j = c_s = 20$, $t_j = t_s = 80\,\mu$s), hop duration for USD-FH is fixed as 63 μs, $x = 78$

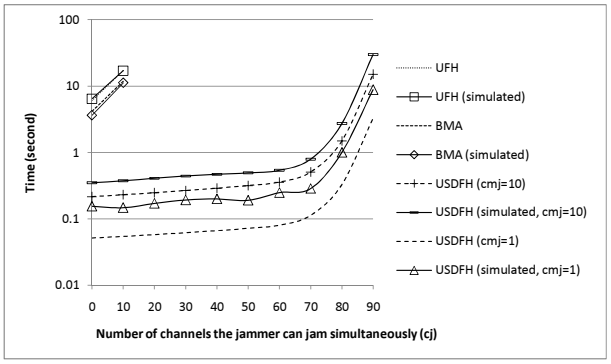


Figure 8. Communication time for scenario 4 under the hybrid jamming attack ($c_s = 10$, $t_j = t_s = 40\,\mu$s), hop durations for UFH and BMA are $630\,\mu$s, hop duration for USD-FH is fixed as $63\,\mu$s, $x = 77, 78, 79, 80, 81, 82, 84, 85, 86, 88$ for $c_j = 0$ to 90

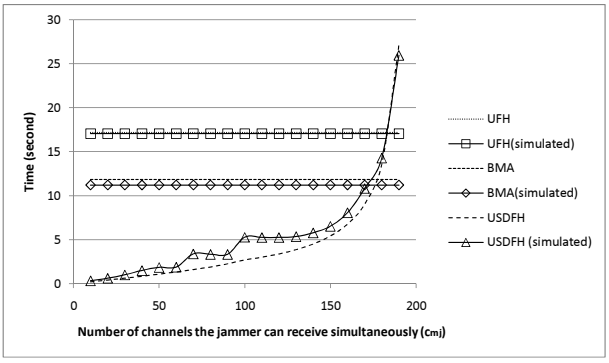


Figure 9. Communication time for scenario 5 under the hybrid jamming attack ($c_s = c_j = 10$, $t_j = t_s = 40\,\mu$s), hop durations for UFH and BMA are $630\,\mu$s, hop duration for USD-FH is fixed as $63\,\mu$s, $x = 16, 9, 6, 4, 3, 3, 2, 2, 2$ for $c_{mj} = 10$ to 90, $x = 1$ for $c_{mj} > 90$

ios 2 and 3, the low-level jamming probability increases slowly in scenario 1. Thus UFH and BMA have shorter communication time for wide hop duration range in scenario 1 than in scenarios 2 and 3. In the simulation results, UFH and BMA can finish the transmission in less than 10 seconds for all hop durations between $294\,\mu$s and $1{,}050\,\mu$s in scenario 1 (Figure 5). But for scenario 2 (Figure 6), UFH and BMA can only finish in less than 10 seconds for hop durations between $294\,\mu$s and $504\,\mu$s and between $315\,\mu$s and $609\,\mu$s, respectively. In scenario 3 (Figure 7), UFH and BMA can only finish in less than 10 seconds for hop durations between $294\,\mu$s and $420\,\mu$s and between $315\,\mu$s and $546\,\mu$s, respectively. From the first three scenarios, we can see that both UFH and BMA have an optimal range of hop duration which can keep the communication time low (e.g., under 10 seconds).

Figures 5–7 show that BMA is more efficient than UFH most of the times except when the hop duration is very short (e.g., $273\,\mu$s). BMA can be up to 1.7 times faster than UFH in scenarios 1–3.

USD-FH is more efficient than both UFH and BMA. Since the hop duration is only $63\,\mu$s for USD-FH, the low-level responsive jamming attack does not work well for USD-FH. As a result, the low-level jamming probability against USD-FH is very low (e.g., 0.05, 0.08, and 0.07 for scenarios 1, 2, and 3, respectively). Even when we consider the high-level responsive jamming attack, the receiver still has a high probability to receive message M. In the simulation results, the communication time of USD-FH is 0.18 second for scenarios 1, 2, and 3. Thus, USD-FH is at least 20, 43, and 45 times faster than UFH in scenarios 1, 2, and 3, respectively. Similarly, USD-FH is at least 14, 30, and 33 times faster than BMA in scenarios 1, 2, and 3, respectively.

Scenario 4: We can see that USD-FH is much more efficient and robust than both UFH and BMA in scenario 4 as well, as shown in Figure 8. When $c_j \geq 20$, the low-level jamming probability for UFH and BMA is 1 when the attacker applies the responsive sweep strategy as Figure 4 shows. That means both UFH and BMA are always jammed when $c_j \geq 20$. In contrast, USD-FH with the short hop duration ($63\,\mu$s) has much lower low-level jamming probability as Figure 4 shows. The low-level jamming probability is less than 0.6 even when the jammer can jam 90 channels (out of 200 channels) simultaneously. Thus USD-FH can still work when $c_j = 90$, while both UFH and BMA stop working when $c_j \geq 20$. Furthermore, USD-FH are much faster than UFH and BMA. When the hybrid jammer can receive on 1 channel only, USD-FH is 40 and 96 times faster than UFH, and 20 and 63 times faster than BMA for $c_j = 0$ and 10, respectively. When the hybrid jammer can receive on 10 channels simultaneously, USD-FH is 10 and 24 times faster than UFH, and 5 and 16 times faster than BMA. When $c_j = 90$, the communication time of USD-FH is still lower than 10 and 15 seconds for $c_{mj} = 1$ and 10, respectively.

Scenario 5: In this scenario, as Figure 9 shows, we set the number of channels c_{mj} on which the jammer can receive packet simultaneously from 10 to 190, so that we can see how USD-FH works under the powerful high-level responsive jamming attack. Since UFH and BMA are not vulnerable to high-level responsive jamming attack, the communication time for them does not change when c_{mj} changes. When $c_{mj} = 200$, the jammer can receive on all channels, thus USD-FH will fail. However, we notice that USD-FH is faster than both UFH and BMA when $c_{mj} < 180$. The jammer who can receive packets on 180 channels simultaneously is a very powerful jammer. The cost to get such powerful capability would be very high.

Analysis vs. simulation: To compare with simulation results, we also show analytical results of communication time of UFH, BMA, and USD-FH in Figures 5–9. For UFH and BMA, the difference between analytical results and simulation results is negligible. For USD-FH, the simulation result is always a little bit higher than the analytical result although they are close. A possible reason is that the hop duration used in USD-FH is much smaller than those in UFH and BMA, causing larger relative errors in the Bluetooth module in Matlab.

VII. Related Work

We have discussed most closely related work in the introduction, including UFH and its variations [3], [4]. We do not repeat them here.

There are other related works. A scheme named Uncoordinated DSSS (UDSSS) was proposed to provide the anti-jamming *broadcast* communication using DSSS [11]. Another scheme called Randomized Differential DSSS (RD-DSSS) was recently developed to provide more resistant against responsive (reactive) jamming attacks than UDSSS [12]. A coding approach was proposed to encode data to be transmitted into "marks" (e.g., short pulses at different times) that can be decoded without any prior knowledge of keys [13]. A code tree based technique that enables the system to identify insider jammers was proposed in [14], [15]. Xu et al. proposed to employ consistency checking for detecting jamming attacks [16]. The problems such as how to mitigate jamming on control channels [17], [18] and sensor networks [19]–[21] were also studied by previous researchers. Our technique is complementary to these techniques.

VIII. Conclusion

In this paper, we developed a novel, efficient, and robust scheme named USD-FH, which uses Uncoordinated Seed Disclosure in Frequency Hopping to establish a shared key in presence of jammers. USD-FH employs a one-time pseudo-random hopping pattern to transmit each DH key establishment message, and discloses the seed of the pseudo-random hopping pattern in an uncoordinated manner before the actual message transmission. We compared USD-FH with previous techniques developed for the same objective through both theoretical analysis and simulation. Our results show that USD-FH is much more efficient and robust than all previous approaches.

References

[1] A. Goldsmith, *Wireless Communications*. Cambridge University Press, August 2005.

[2] M. Strasser, C. Pöper, S. Čapkun, and M. Čagalj, "Jamming-resistant key establishment using uncoordinated frequency hopping," in *Proceedings of the 2008 IEEE Symposium on Security and Privacy*, 2008, pp. 64–78.

[3] M. Strasser, C. Pöpper, and S. Čapkun, "Efficient uncoordinated FHSS anti-jamming communication," in *Procceedings of MobiHoc '09*, May 2009.

[4] D. Slater, P. Tague, R. Poovendran, and B. Matt, "A coding-theoretic approach for efficient message verification over insecure channels," in *Procceedings of the 2nd ACM Conference on Wireless Networking Security (WiSec '09)*, March 2009, pp. 151–160.

[5] R. A. Scholtz, *Spread Spectrum Communications Handbook*. McGraw-Hill, 2001.

[6] Certicom Research, "Standards for efficient cryptography – SEC 1: Elliptic curve cryptography," http://www.secg.org/download/aid-780/sec1-v2.pdf, May 2009.

[7] R. Poisel, *Modern Communications Jamming Principles and Techniques*. Artech House Publishers, 2006.

[8] "Simulink," http://www.mathworks.com/products/simulink.

[9] L. Rizzo, "Effective erasure codes for reliable computer communication protocols," *ACM Computer Communication Review*, vol. 27, no. 2, pp. 24–36, April 1997.

[10] "Mathematica," http://www.wolfram.com/products/mathematica/index.html.

[11] C. Pöpper, M. Strasser, and S. Čapkun, "Jamming-resistant broadcast communication without shared keys," in *Proceedings of the USENIX Security Symposium*, 2009.

[12] Y. Liu, P. Ning, H. Dai, and A. Liu, "Randomized differential dsss: Jamming-resistant wireless broadcast communication," in *Proceedings of the 2010 IEEE INFOCOM*, 2010.

[13] L. Baird, W. Bahn, and M. Collins, "Jam-resistant communication without shared secrets through the use of concurrent codes," US Air Force Academy, Tech. Rep., 2007.

[14] J. Chiang and Y. Hu, "Extended abstract: Cross-layer jamming detection and mitigation in wireless broadcast networks," in *Proceedings of ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '07)*, 2007.

[15] ——, "Dynamic jamming mitigation for wireless boradcast networks," in *Proceedings of IEEE International Conference on Computer Communications (INFOCOM '08)*, 2008.

[16] W. Xu, W. Trappe, Y. Zhang, and T. Wood, "The feasibility of launching and detecting jamming attacks in wireless networks," in *Proceedings of ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '05)*, 2005.

[17] P. Tague, M. Li, and R. Poovendran, "Probabilistic mitigation of control channel jamming via random key distribution," in *Proceedings of IEEE 18th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC '07)*, 2007, pp. 1–5.

[18] L. Lazos, S. Liu, and M. Krunz, "Mitigating control-channel jamming attacks in multi-channel ad hoc networks," in *Proceedings of 2nd ACM Conference on Wireless Networking Security (WiSec '09)*, March 2009.

[19] W. Xu, W. Trappe, and Y. Zhang, "Channel surfing: Defending wireless sensor networks from jamming and interference," in *Proceedings of the 6th International Conference on Information Processing in Sensor Networks (IPSN '07)*, 2007.

[20] M. Li, I. Koutsopoulos, and R. Poovendran, "Optimal jamming attacks and network defense policies in wireless sensor networks," in *Proceedings of IEEE International Conference on Computer Communications (INFOCOM '07)*, 2007.

[21] W. Xu, T. Wood, W. Trappe, and Y. Zhang, "Channel surfing and spatial retreats: Defenses against wireless denial of service," in *Proceedings of the 3rd ACM Workshop on Wireless Security (WiSe '04)*, 2004.