2013 IEEE Symposium on Security and Privacy

Ally Friendly Jamming: How to Jam Your Enemy and Maintain Your Own Wireless Connectivity at the Same Time

Wenbo Shen, Peng Ning
Department of Computer Science
North Carolina State University
Raleigh, NC 27695
{wshen3, pning}@ncsu.edu

Xiaofan He, Huaiyu Dai
Department of Electrical and Computer Engineering
North Carolina State University
Raleigh, NC 27695
{xhe6, hdai}@ncsu.edu

***Abstract*—This paper presents a novel mechanism, called *Ally Friendly Jamming*, which aims at providing an intelligent jamming capability that can disable unauthorized (enemy) wireless communication but at the same time still allow authorized wireless devices to communicate, even if all these devices operate at the same frequency. The basic idea is to jam the wireless channel continuously but properly control the jamming signals with secret keys, so that the jamming signals are unpredictable interference to unauthorized devices, but are recoverable by authorized ones equipped with the secret keys. To achieve the ally friendly jamming capability, we develop new techniques to generate ally jamming signals, to identify and synchronize with multiple ally jammers. This paper also reports the analysis, implementation, and experimental evaluation of ally friendly jamming on a software defined radio platform. Both the analytical and experimental results indicate that the proposed techniques can effectively disable enemy wireless communication and at the same time maintain wireless communication between authorized devices.**

***Keywords*-Wireless; friendly jamming; interference cancellation**

I. Introduction

Wireless communication technology has been widely deployed and increasingly adopted due to the ease of installation and reduced operational cost. The applications that benefit from wireless communication range from traditional military operations to more recent civilian applications such as Wi-Fi and mobile phones. There have also been ongoing efforts aimed at adopting wireless communication in emerging and mission-critical applications (e.g., healthcare [12], [18] and critical infrastructure protection [6], [10]).

In mission-critical applications such as battlefield operations, anti-terrorism activities, and critical infrastructure protection, it is highly desirable and sometimes necessary to gain advantages over the adversary in terms of wireless communication capability. In particular, *it is highly desirable to disable the adversary's (unauthorized) wireless communication while still maintaining our own (authorized) wireless communication.* For example, wireless communication has been a common way to trigger Improvised Explosive Devices (IED) (a.k.a. roadside bombs), which were responsible for approximately 63% coalition deaths in the second Iraq war from 2001 to 2007 and over 66% of the coalition casualties in Afghanistan between 2001 and 2012 [2]. The capability of disabling enemy wireless communication and at the same time maintaining coalition's wireless connectivity would greatly reduce the casualties due to radio-controlled IED. It is conceivable that such a capability will also enhance the security of other non-military mission-critical applications such as critical infrastructure protection and health-care applications.

This paper aims at providing such a capability. Specifically, we develop a novel mechanism, called *Ally Friendly Jamming*, to provide an intelligent jamming capability that can disable unauthorized (enemy) wireless communication but at the same time still allow authorized wireless devices to communicate, even if both the authorized and unauthorized devices operate at the same frequency.

The basic idea behind ally friendly jamming is to jam the wireless channel continuously but properly control the jamming signals using secret keys, so that the jamming signals are unpredictable interference to unauthorized devices, but are recoverable by authorized devices equipped with the secret keys. As a result, when authorized devices need to communicate, they can employ proper signal processing techniques to remove the jamming signals and recover the messages transmitted by other authorized devices. In other words, authorized devices can regenerate jamming signals using the secret keys and subtract them from the received, mixed signals to get jamming-free transmissions.

Though conceptually simple, ally friendly jamming turns out to be non-trivial to achieve. We have to resolve three technical challenges to ensure effective jamming and at the same time enable authorized devices to actually receive messages under ally friendly jamming, even though such devices know the secret keys.

First, to achieve ally friendly jamming, the ally jamming signals need to be irresolvable interference to unauthorized devices. Simply transmitting modulated pseudo random numbers as jamming messages can be easily defeated due to the strong patterns introduced by the digital communication

1081-6011/13 $26.00 © 2013 IEEE
DOI 10.1109/SP.2013.22

process (e.g., modulation) [15]. Thus, the jamming signals injected by ally jammers must resemble real random noises. In the proposed ally friendly jamming scheme, we introduce the concept of *epoch* and use the shared keys with epoch indices as the input of a pseudo random number generator to directly control physical layer symbols, so that these signals are random noises to unauthorized devices and easy for authorized devices to synchronize with.

Second, an authorized receiver has to synchronize with the ally jammers, so that it can estimate the ally jamming signals, remove them from received signals, and recover potential transmissions from authorized transmitters. Though synchronization is a well-studied problem in digital communication, synchronization in ally friendly jamming faces a new challenge. As the channel and hardware effects (e.g., frequency offset) on the received ally jamming signals are unknown, the authorized receiver cannot synchronize with ally jammers even though it can generate the same transmitted ally jamming signals. The frequency offset can be compensated for by using the phase-locked loop which depends on the strong phase patterns existing in the transmitted signals. As ally jamming signals mimic random noises, no strong patterns can be relied on, existing synchronization approaches (e.g., [11], [16], [28], [34]) cannot be applied directly in ally friendly jamming. In this paper, we propose to use the pilot frequency aided correlation to synchronize authorized receivers with multiple ally jammers.

Third, when multiple ally jammers exist in the network, an authorized receiver needs to first identify these ally jammers properly and then regenerate the transmitted ally jamming signals in order to recover the authorized transmission. A particular challenge lies in how to identify these ally jammers rapidly while their ally jamming signals are pseudo-random signals and the channel and hardware effects on the received ally jamming signals are unknown. To solve this problem, we propose to use the pilot frequency and the fast Fourier transform (FFT) to identify ally jammers and further compensate for the hardware difference effects on the received signals.

A similar technique called IMD (Implantable Medical Device) shield [12] was proposed recently which exploited jamming to provide access control to an IMD. The IMD shield is a small radio device that employs two antennas for jamming and receiving, respectively. The receive antenna is physically connected to a transmit (jam)-and-receive chain, so that when sending a jamming signal, the jam chain can inject an "antidote" signal to the receive antenna to cancel the jamming signal. Due to the physical connection between the jamming and the receiving antennas, IMD shield does not have to deal with the synchronization challenge addressed in this paper. Moreover, the multiple-jammer case was not considered in IMD shield. This means if multiple IMD shields operate at the same time in the same area, their jamming signals will interference with each other, and all accesses will be denied. Therefore, by providing solutions to the above problems, our work further advances the current state of the art in security enhancement through friendly jamming.

We have implemented a prototype for ally friendly jamming using the Universal Software Radio Peripheral (USRP) platform [25] and GNURadio [1]. Our experimental results show that under ally friendly jamming, authorized devices have close-to-0 packet loss rate, and at the same time unauthorized devices suffer from 100% packet loss rate.

The contributions of this paper are summarized as follows: We explore a new concept called ally friendly jamming that can disable unauthorized wireless communication and at the same time allow authorized devices to maintain wireless connectivity. We develop new techniques to generate ally jamming signals, to identify and synchronize with multiple ally jammers. We have also implemented a prototype for ally friendly jamming and performed analysis and extensive experimental evaluation to validate the techniques.

The remainder of this paper is organized as follows. Section II provides some background knowledge on wireless communication. Section III clarifies our assumptions and threat model. Section IV presents the proposed ally friendly jamming scheme in detail. Section V describes the analysis and limitations. Section VI presents the implementation and experimental evaluation of ally friendly jamming. Section VII discusses related work. Finally, Section VIII concludes the paper and points out some future research directions.

II. Preliminaries

Wireless digital communication systems generally employ radio frequency (RF) signals to transmit information. Transmitters need to convert digital messages represented in bits to RF signals, while receivers convert received RF signals back to digital messages. Figure 1 shows a simplified structure for a wireless digital communication system with one transmitter and one receiver. On the transmitter side, upon receiving bits from upper layers, the transmitter first modulates them to discrete baseband signals (a.k.a. *physical layer symbols*, or simply *symbols*), then converts them to analog signals using a digital to analog converter (DAC), and finally up-converts them to RF signals. The RF signals go through the wireless channel and reach the receiver. Upon receiving the RF signals, the receiver performs the inverse processing. It down-converts and samples the received signals to discrete baseband signals, and then demodulates them to bits.

Physical layer symbols are represented by complex numbers. For example, when BPSK is used for modulation, the transmitter modulates bit "1" to $x = 1 + 0j$ and bit "0" to $x' = -1 + 0j$ (j is the imaginary unit, satisfying $j^2 = -1$). A symbol $x_i = a + bj$ is often represented in its

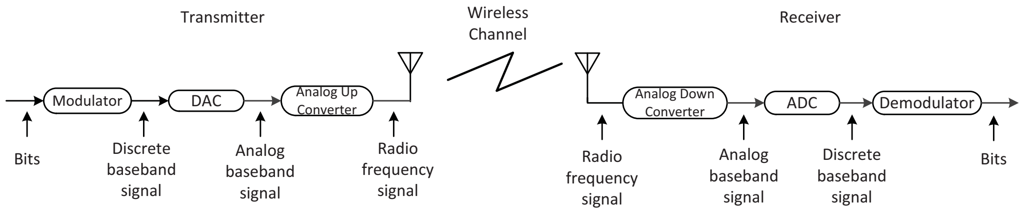


Figure 1. Simplified structure for a wireless digital communication system.

polar form $x_i = Me^{j\theta}$, where $M = |x_i| = \sqrt{a^2 + b^2}$ and $\theta = \tan^{-1}(b/a)$ [26].

The wireless channel introduces attenuation, phase shift, and additional noise during transmission. After the signal x_i is transmitted through the channel, it is transformed into the received signal

$$y_i = he^{j\gamma}x_i + n_i,$$

where h is the *channel attenuation*, γ is the *phase shift*, and n_i is the *noise*.

In practice, the signal reception at the receiver is also affected by two additional factors: *frequency offset* and *sampling offset*. Frequency offset Δf generally exists between the transmitter and the receiver, since there is no practical way to guarantee that two radios operate at exactly the same frequency. Δf causes variations on the phases of received signals [13]. Thus, if we take Δf into consideration, the received signal becomes

$$y_i = he^{j\gamma}e^{j2\pi\Delta f t_i}x_i + n_i, \quad (1)$$

where t_i is the time at which the receiver gets the sample y_i.

Moreover, the receiver uses sampling and quantization to recover the original baseband signals. Due to the lack of perfect synchronization in wireless communications, the receiver usually cannot sample perfectly to get the exact physical layer symbols sent by the transmitter. When the sampling offset is considered, the received signal becomes

$$y_i = he^{j\gamma}e^{j2\pi\Delta f t_i}x_{i+\mu} + n_i, \quad (2)$$

where μ is the sampling offset due to mis-sampling.

In summary, the wireless channel and the hardware differences introduce various distortion to the signal transmission. To correctly recover the transmitted messages, the receiver need to either estimate these parameters to certain accuracy or tolerate their influences.

III. Assumption and Threat Model

Assumptions: We assume that there are multiple ally jammers and multiple authorized wireless devices, all of which share a secret key set that is unknown to unauthorized devices. We assume a high signal-to-noise radio (SNR) for both transmission signals and ally jamming signals at the receiver. We also assume that the clocks at ally jammers and authorized devices are loosely synchronized, and the frequency offsets between ally jammers and authorized devices are within a given range. We assume that ally jammers can block the operational frequencies of all devices, including both authorized and unauthorized devices. In other words, unauthorized devices cannot find a wireless communication channel that is not being jammed by the ally jammers. We also assume that the adversary cannot defeat ally friendly jamming by physically removing ally jammers. Finally, we assume that each device (authorized or unauthorized) is equipped with a single omnidirectional antenna and there is no adversarial jammer. How to accomplish ally friendly jamming with MIMO (multiple-input and multiple-output) devices and how to maintain wireless communication under both ally and adversarial jamming will be addressed in our future work.

Threat Model: We consider unauthorized devices as potential adversaries. The objective of unauthorized devices is to defeat the proposed scheme so that they can communicate under ally friendly jamming. They may analyze the ally friendly jamming signals and attempt to use the result of analysis to remove the jamming signals with signal processing techniques (e.g., [8], [9]). They may also employ anti-jamming communication techniques such as Direct Sequence Spread Spectrum (DSSS), Frequency Hopping Spread Spectrum (FHSS), and their variations (e.g., [20], [32], [39]).

IV. Ally Friendly Jamming

In ally friendly jamming, upon detecting a transmission, the authorized device can employ proper signal processing techniques to remove the jamming signals from the received, mixed signals. In contrast, the unauthorized device does not have the secret keys, and cannot remove the interference introduced by ally jamming signals.

Figure 2 further illustrates ally friendly jamming, where one ally jammer is presented for simplicity. Assuming the ally jammer, the authorized and unauthorized devices are all in the same area. As mentioned earlier, the ally jammer and authorized devices, including A_1, A_2, and AJ in Figure 2, share a secret key k. The ally jammer AJ uses a Pseudo-Random Number Generator (PRNG) with k as the seed to continuously emit jamming signals X_J.

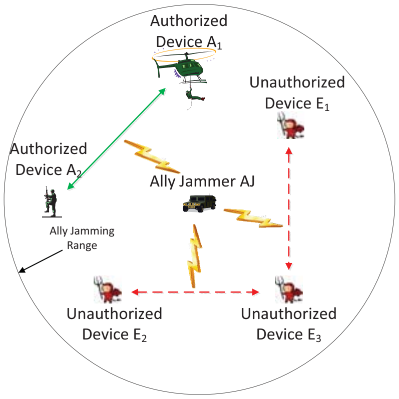


Figure 2. Illustration of ally friendly jamming.

When the unauthorized device E_1 transmits signals X_{E_1} to another unauthorized device E_3, the signals received by E_3 will be the mixture of both X_{E_1} and some portion of X_J. With enough jamming power, the jamming signals from AJ can effectively distort the signals X_{E_1} at E_3. As a result, the wireless communication between unauthorized devices E_1 and E_3 is disabled.

When A_1 transmits signals X_{A_1} to A_2, the jamming signals X_J will also distort the received signals at A_2. However, since A_2 shares the same secret key k with AJ, it can regenerate the same jamming signals X_J using k. If it can find out which portion of X_J is mixed with X_{A_1}, it can subtract this portion of X_J to get a clean copy of X_{A_1}. To remove X_J from the mixed signals, authorized devices need to synchronize with the ally jamming signals, estimate their values in the mixed signals, and remove them from the received, mixed signals to recover meaningful transmissions.

In the following sections, we will present how the ally jammer generates ally jamming signals and how the authorized device synchronizes with ally jammers and recovers the transmissions.

A. Generation of Ally Jamming Signals

Every ally jammer uses a shared, unique secret key to generate its ally jamming signals. Ally jammers and authorized devices share a set of secret keys. Either group key agreement (e.g., [17], [22], [43]) or group key distribution protocols (e.g., [7], [23], [30]) can be used to generate the secret key set. Assuming there are n ally jammers in the network, identified as $AJ_1, AJ_2, \ldots, AJ_n$ and n keys $k_1, k_2, \ldots, k_n$ in the key set, the key k_g will be assigned to the ally jammer AJ_g.

To ensure effective jamming against unauthorized devices, the jamming signals injected by ally jammers should resemble random noises. To achieve this goal, we use a PRNG to directly control the physical layer symbols so that these signals appear to be random noises to unauthorized devices. Since a physical layer symbol is represented as a complex number, we can use a PRNG to generate random floating point numbers with certain precision as the real and the imaginary parts of each symbol.

Moreover, the injected jamming signals should allow the authorized devices, which have access to the secret keys, to synchronize with ally jammers, even they join the network in the middle of a jamming session and the jamming has been going on for a long period of time.

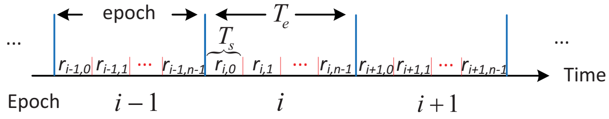


Figure 3. Generation of jamming signals.

To accomplish these goals, we make the following design, illustrated in Figure 3. We divide the time into equal-sized epochs, each of which consists of n physical layer symbols. Assuming that the duration of each physical layer symbol is T_s. Then the duration of each epoch is $T_e = n \cdot T_s$. For simplicity, we consider the time period $[i \cdot T_e, (i+1) \cdot T_e)$ as the i-th epoch, where i is the epoch index. For convenience, we also index and label the physical layer symbols within each epoch. For example, in Figure 3, the symbols in the i-th epoch are indexed from 0 to $n-1$ and labeled as $r_{i,0}$ through $r_{i,n-1}$. With this design, for any given time t, we can easily compute the corresponding epoch index as $i = \lfloor \frac{t}{T_e} \rfloor$, and the symbol index within the epoch as $m = \lfloor \frac{t - i \cdot T_e}{T_s} \rfloor$. The corresponding physical layer symbol is thus $r_{i,m}$.

To allow easy synchronization with the jamming signals on authorized devices, we propose to use both the secret key and the epoch index to control the PRNG for jamming signal generation. Specifically, to generate the jamming symbols in epoch i, the ally jammer, say AJ_g, first uses the key k_g and the epoch index i as the seed to the PRNG to get a sequence of pseudo random floating numbers, i.e., $\langle a_0, a_1, \ldots, a_{2n-1} \rangle = PRNG(k_g, i)$, and then forms each jamming symbol $r_{i,m}$ as $r_{i,m} = a_{2m} + a_{2m+1} \cdot j$, where $m = 0, 1, ..., n-1$. As a result, the jamming signals are pseudo-random samples, which are independent of the noise and shifted versions of themselves. Therefore, when an authorized device comes to the network, it can refine its synchronization with the ally jammer, and eventually remove the jamming signals.

Note that the quality of the jamming signals is affected by two parameters: the duration of each jamming symbol T_s, and the precision of the pseudo random numbers used for the real and the imaginary parts of jamming symbols. To maximize the uncertainty of the jamming signals, the smallest value for T_s and the maximum precision allowed for the jamming symbols can be used. Both parameters are eventually limited by the hardware used for emitting jamming signals. Finally, to ensure the randomness, the jamming symbols should be transmitted without modulation and encoding.

B. Synchronizing with Ally Jamming Signals

1) Synchronizing by Correlation: An authorized device has to synchronize with ally jammers, so that it can estimate and remove the ally jamming signals to maintain its communication. The goal of synchronization is to align the received ally jamming symbols with the locally generated ally jamming signals, even though these received signals have been distorted by the unknown wireless channel parameters(i.e., when the parameters γ, Δf, and μ in Equation (2) are unknown).

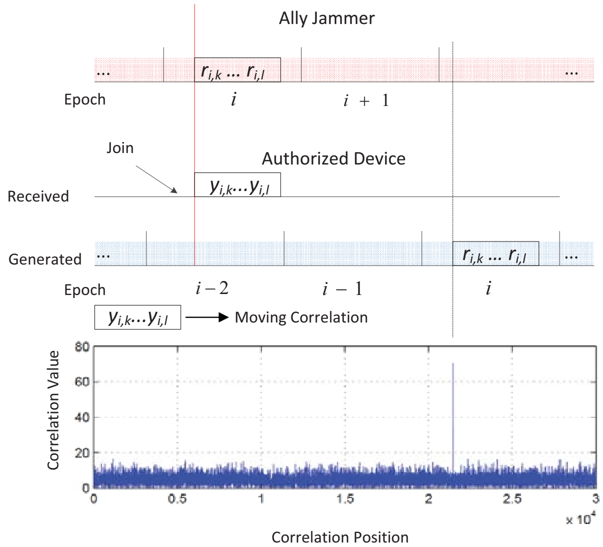


Figure 4. Synchronization with ally jamming signals.

Let us use Figure 4 to explain the synchronization process in ally friendly jamming. In this and the following two sections, we will focus on one ally jammer for simplicity, and defer the discussion of multiple ally jammers to the Section IV-E. Assuming when an authorized device joins the network, the ally jammer, say AJ_g, is in the i-th epoch on its local clock and the ally jamming signals being transmitted are $r_{i,k}, \dots, r_{i,l}$. The corresponding jamming signals received by the authorized device are $y_{i,k}, \dots, y_{i,l}$. Assuming the frequency offset between AJ_g and the authorized device is Δf_g, based on Equation (1), we have

$$y_{i,m} = he^{j\gamma} e^{j2\pi\Delta f_g t_{i,m}} r_{i,m} + n_{i,m}, m \in [k, l].$$

At the same time, the authorized device is in the $(i+\delta)$-th epoch on its own clock ($\delta = -2$ in Figure 4). Assuming the authorized device knows that the ally jammer is AJ_g (we will address how to distinguish ally jammers in Section IV-C), it can use the secret key k_g and its epoch indices to regenerate the ally jamming symbols locally. It is assumed that the ally jammer and authorized devices are loosely synchronized, with maximum clock difference of ΔT. Thus, the current local epochs of this authorized device and the ally jammer will not be more than $w = \lceil \frac{\Delta T}{T_e} \rceil$ epochs away from each other, and the authorized device only needs to consider possible symbol alignments within this time window. In our example, since the authorized device is in the $(i-2)$-th epoch, it should regenerate the following sequence of jamming symbols from the ally jammer: $r_{d,0}, r_{d,1}, \dots, r_{d,n-1}$, where $d \in [i-2-w,\ i-2+w]$.

To obtain the synchronization with the ally jammer, the authorized device can use correlation to find the location of the received samples $y_{i,k}, \dots, y_{i,l}$ in the locally generated symbols. Correlation is a popular technique for detecting known signal patterns on the receiver side. Assuming the correlation length is L. The authorized device can firstly align $y_{i,k}, \dots, y_{i,k+L-1}$ with the first L signals in $r_{d,0}, r_{d,1}, \dots, r_{d,n-1}$, compute the correlation, shift the alignment by one sample and re-compute the correlation, until a spike at the correlator output is identified. The jamming signals are pseudo-random samples, which are independent of the noise and shifted versions of themselves. Therefore, the correlation is near zero except when the correct alignment is found.

However, the above statement is only partially correct as the frequency offset can disrupt the correlation. For example, assuming the correlation output is Γ:

$$\begin{aligned}\Gamma &= \sum_{n=0}^{L-1} y_{i,k+n} \cdot r^*_{i',k'+n} \\ &= \sum_{n=0}^{L-1} [he^{j\gamma} e^{j2\pi\Delta f_g t_{i,k+n}} r_{i,k+n} + n_{i,k+n}] \cdot r^*_{i',k'+n},\end{aligned}$$

where $r_{i',k'+n}$ is a signal in the locally generated jamming signal sequence $r_{d,0}, r_{d,1}, \dots, r_{d,n-1}$ and $r^*_{i',k'+n}$ is its complex conjugation. As $r^*_{i',k'+n}$ is independent of noise, $n_{i,k+n}$ will be canceled out. If the correct alignment is found, say $i' = i$ and $k' = k$, then we have

$$\Gamma \approx he^{j\gamma} \sum_{n=0}^{L-1} |r_{i,k+n}|^2 e^{j2\pi\Delta f_g t_{i,k+n}}.$$

The frequency offset part $e^{j2\pi\Delta f_g t_{i,k+n}}$ introduces dynamic phases to the individual components in the above sum, which may lead to signal cancellation. Therefore, the authorized device must compensate for frequency offset before the correlation can be used for synchronization. After compensating for the frequency offset (we will discuss frequency offset compensation in IV-C), the correlation output becomes:

$$\begin{aligned}\Gamma &\approx he^{j\gamma} \sum_{n=0}^{L-1} |r_{i,k+n}|^2 e^{j2\pi\Delta f_g t_{i,k+n}} \cdot e^{-j2\pi\Delta f_g t_{i,k+n}} \\ &\approx he^{j\gamma} \sum_{n=0}^{L-1} |r_{i,k+n}|^2.\end{aligned}$$

The correlation spikes when the received signals are aligned correctly with the generated signals, as shown in Figure 4.

Therefore, by detecting the correlation spike, the authorized device is able to synchronize with the ally jammer.

Recall that there is also a sampling offset between the received ally jamming signals and the self-generated signals. For example, assuming for any transmitted jamming signal $r_{i,m}$, the received signal by authorized device with sampling offset μ is $r_{i,m+\mu}$. After generating $r_{i,m}$ with the shared key, the authorized device interpolates it at a rate of N. As a result, $r_{i,m}$ will be expanded to $r_{i,m+p/N}, p = 0, \ldots, N-1$. When N is large enough (in our experiments, $N = 16$ gives a good enough resolution), there will be a value p_0 such that $p_0/N \approx \mu$, as shown in Figure 5. The authorized device can use p_0/N to approximate the sampling offset μ.

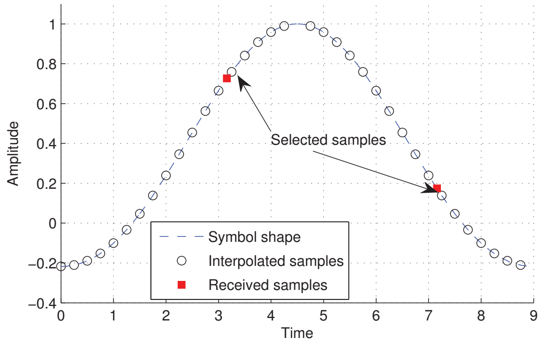


Figure 5. Received samples interpolation. Interpolation rate $N = 16$. The selected interpolated samples are close to the received samples.

To decide the value of p_0, the authorized device uses a selection of the interpolated samples rather than the samples before interpolation, to correlate with the received signals. The authorized device can try all values of $p = 0, 1, \ldots, N-1$, the one achieving the maximum correlation spike value is regarded as p_0, which can be used to approximate the sampling offset for the following samples.

C. The Introduction of Pilot Frequencies

In order to compensate for the frequency offset as well as identify ally jammers rapidly, we introduce the concept *pilot frequency* into ally friendly jamming. A pilot frequency is a 1 Hz wide frequency uniquely associated with each ally jammer, injected along with the pseudo-random jamming signals into the channel. On the receiver side, the authorized device can use this pilot frequency to identify the associated ally jammer and compute the frequency offset between them.

Before applying pilot frequency, we need to assign a proper pilot frequency to each ally jammer. Assuming the maximum frequency offset between ally jammers and authorized devices is f_{max}, the frequency offset $\Delta f \in [0, f_{max})$. We assign $(2g-1)f_{max}$ as the ally jammer AJ_g's pilot frequency and designate $[(2g-2)f_{max},\ 2gf_{max})$ as the associated shift range, as shown in Figure 6.

For each ally jammer, along with the generated pseudo-random signals, it also generates the signals of its pilot frequency. Assume an epoch has n pseudo-random signals, the ally jammmer will generate n pilot frequency signals, and apply them to all epochs. For example, for the ally jammer AJ_g with the pilot frequency $(2g-1)f_{max}$, the pilot frequency signal it will generate for the m-th pseudo-random signals in all epochs is

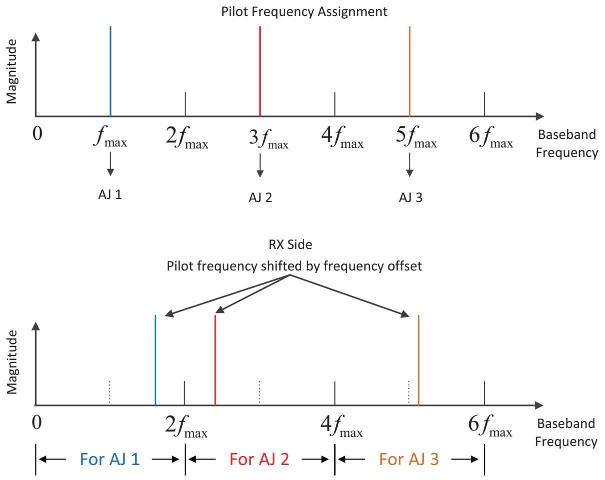


Figure 6. Pilot frequency assignment.

$$pf_m = e^{j2\pi(2g-1)f_{max}mT_s}.$$

pf_m will be added up onto the m-th generated pseudo-random signals in all the epochs. Hence the m-th jamming signals in epoch i, say $s_{i,m}$, is given by

$$s_{i,m} = r_{i,m} + pf_m.$$

On the receiver side, for transmitted signal $s_{i,m}$, assuming the frequency offset is Δf_g, the authorized device will receive

$$\begin{aligned} y_{i,m} &= he^{j\gamma}e^{j2\pi\Delta f_g t_{i,m}}s_{i,m} + n_{i,m} \\ &= he^{j\gamma}e^{j2\pi\Delta f_g t_{i,m}}(r_{i,m} + pf_m) + n_{i,m}. \end{aligned}$$

As $r_{i,m}$ are pseudo-random samples, their energy is spread over a wide range of spectrum. On the other hand, the pilot frequency signals pf_m concentrate all their energy on a narrow band (1Hz wide), which will achieve a much larger magnitude, as shown in Figure 7. Therefore, on the receiver side, if the authorized device analyzes the spectrum of the received signals, it will find a spike within the designated shift range of the pilot frequency. Since the designated pilot frequency shift ranges of different ally jammers do not overlap, as shown in Figure 6, the pilot frequencies can be used for ally jammer identification.

Assuming the ally jammer AJ_g is identified, the authorized device knows its pilot frequency $(2g-1)f_{max}$. And as $\Delta f_g + (2g-1)f_{max}$ has also been detected, the authorized device can infer their frequency offset Δf_g, which can be used further to compensate for their frequency offset.

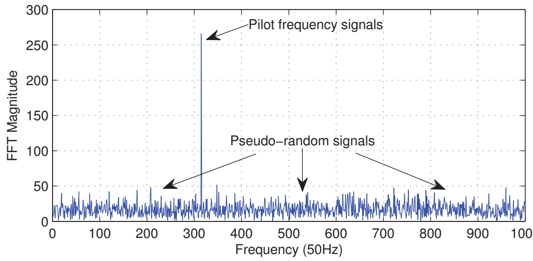


Figure 7. Received signal spectrum. Only show a portion of the whole spectrum.

D. Detecting and Recovering Transmissions

After synchronizing with the ally jamming signals, the authorized device needs to detect and recover potential transmissions from other authorized devices. Before a transmission is recovered, the authorized device cannot distinguish it is authorized or unauthorized. Therefore, the authorized device will try to detect and recover all transmissions in the same way. For simplicity, in this section and the following section, we assume all transmissions are authorized transmissions. And we also assume that there is only one authorized transmission at one time, the media access control mechanism in ally friendly jamming will be presented later.

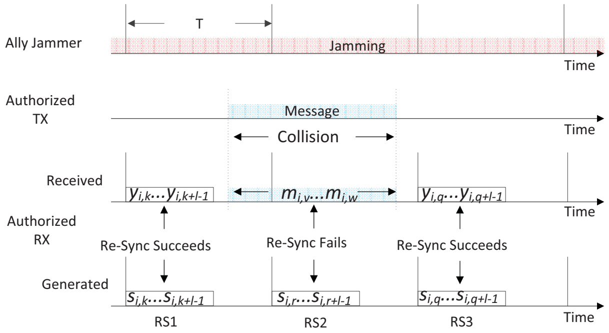


Figure 8. Transmission detection and recovery under ally friendly jamming. The authorized RX and the ally jammer are both in i-th epoch. s is the regenerated ally jamming signal, y is the received ally jamming signal, m is the received collided signal. T is the re-synchronization interval.

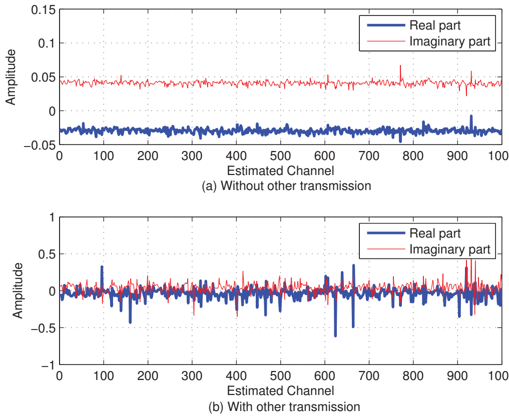
1) Re-synchronization & Transmission Detection: When the authorized device joins the network, it needs to synchronize with the ally jamming signals, this process is denoted as the *initial synchronization*. After initial synchronization, we have each authorized device re-synchronize with the ally jamming signals periodically. Figure 8 illustrates the re-synchronization process. Assuming that an authorized device re-synchronizes with the ally jamming signals every T time units. At the beginning of each re-synchronization period (e.g., $RS1$ in Figure 8), the authorized device compensates for the frequency offset, and correlates the received symbols with the regenerated ones to get the right alignment. Then it will estimate the channel by forming a quotient between each pair of received and transmitted (regenerated) jamming symbols. For example, as the frequency offset has already been compensated for and the noise is negligible, estimated channel coefficient for the samples in $RS1$ is

$$\begin{aligned} c_{i,u} &= \frac{y_{i,u}}{s_{i,u}} = \frac{he^{j\gamma}s_{i,u}}{s_{i,u}} \\ &= he^{j\gamma}, u \in [k, \ldots, k+l-1]. \end{aligned}$$

If there are no transmissions other than the ally jamming signals in $RS1$, $c_{i,u}$ tends to be stable, as shown in Figure 9 (a). However, when there is an authorized transmission (e.g., $RS2$ in Figure 8), we have

$$c_{i,u} = \frac{he^{j\gamma}s_{i,u} + x_{i,u}}{s_{i,u}}, u \in [r, \ldots, r+l-1],$$

where $x_{i,u}$ is the received signal from the authorized transmission. The stableness of $c_{i,u}$ is corrupted by $x_{i,u}$, as shown in Figure 9 (b). Thus by imposing a threshold on the standard deviation of the estimated channel coefficient, we can detect the existence of an authorized transmission under ally jamming.



(a) Without other transmission

(b) With other transmission

Figure 9. Estimated channel.

To ensure that authorized device does not miss authorized transmissions, we set the re-synchronization interval T as a value smaller than the minimal packet transmission duration.

2) Recovery of Authorized Transmissions: To remove the ally jamming signals, the authorized device firstly needs to estimate the corresponding components from the ally jammer in the received, mixed signals, then subtract them out to recover the detected transmissions.

Let us use the scenario shown in Figure 8 as an example, where the authorized device re-synchronizes successfully in $RS1$, but fails in $RS2$ due to the collision. Since re-synchronization in $RS1$ is successful, the authorized device can obtain the received ally jamming symbols in this interval (i.e., $y_{i,k}, \ldots, y_{i,k+l-1}$ in Figure 8), which contain no strong interference (other strong signals, e.g., authorized transmission signals). As the frequency offset is already compensated for and the SNR is high, the least-square (LS) estimator can be employed to obtain a sufficiently accurate estimation of both h and γ.

The re-synchronization failure in $RS2$ is caused by the collision of an authorized transmission with the ally jamming signals. Assuming the received signal components from the authorized transmission are $x_{i,v}, \ldots, x_{i,w}$, the corresponding received ally jamming signal components in collision are $y_{i,v}, \ldots, y_{i,w}$, then the received collided symbols $m_{i,v}, \ldots, m_{i,w}$, are given by

$$m_{i,u} \quad = \quad y_{i,u} + x_{i,u} + n_{i,u}, u \in [v, \ldots, w].$$

Assuming the estimated channel parameters are h' and γ', the authorized device can get an estimation of $y_{i,v}, \ldots, y_{i,w}$, say $y'_{i,v}, \ldots, y'_{i,w}$, as

$$y'_{i,u} = h'e^{j\gamma'} \cdot s_{i,u}, u \in [v, \ldots, w],$$

where $s_{i,u}$ is the generated ally jamming symbol. Then the authorized transmission can be recovered by subtracting the estimated received ally jamming signals $y'_{i,v}, \ldots, y'_{i,w}$ from the received collided signals $m_{i,v}, \ldots, m_{i,w}$. Thus, assuming the recovered authorized signal is $x'_{i,u}$, we have

$$\begin{aligned} x'_{i,u} &= m_{i,u} - y'_{i,u} \\ &= y_{i,u} + x_{i,u} + n_{i,u} - y'_{i,u} \\ &= he^{j\gamma} \cdot s_{i,u} + x_{i,u} + n_{i,u} - h'e^{j\gamma'} \cdot s_{i,u} \\ &= (he^{j\gamma} - h'e^{j\gamma'}) \cdot s_{i,u} + x_{i,u} + n_{i,u}, u \in [v, \ldots, w]. \end{aligned}$$

As h' and γ' are accurate enough, $(he^{j\gamma} - h'e^{j\gamma'}) \cdot s_{i,u}$ is close to 0. Recall that the SNR of $x_{i,u}$ is larger enough, then the recovered signal $x'_{i,u}$ has sufficient SNR to be demodulated correctly, which further indicates the authorized transmission can be recovered readily.

Note that as the authorized device does not know the boundary of the authorized transmission, it will recover all the signals between two succeed re-synchronizations (i.e., all signals between $RS1$ and $RS3$ in Figure 8). Moreover, the authorized device can also use the received signals in the later successful re-synchronization interval to estimate the channel coefficients and recover transmission in previous intervals. For example, in the scenario shown in Figure 8, the authorized device can use $y_{i,q}, \ldots, y_{i,q+l-1}$ in $RS3$ to estimate the channel, and recover the transmission in $m_{i,v}, \ldots, m_{i,w}$.

E. Dealing with Multiple Ally Jammers

When an authorized device joins the system, it is likely that more than one ally jammers exist in the network, the authorized device needs to be able to remove the ally jamming signals from multiple ally jammers.

1) Synchronization with Multiple Ally Jammers: The authorized device can compute the spectrum of the received signals through FFT and identify all ally jammers by detecting all the spikes on the spectrum. It can further compensate for their frequency offsets and synchronize with each ally jammer through correlation.

Let us use an example to illustrate the process. Assuming that there are n active ally jammers, from AJ_1 to AJ_n, and the received signals at the authorized device are Y, which contain the jamming signals from all ally jammers. For one ally jammer, say AJ_g, if the authorized device does FFT on the received signals, it will find a spike within $[(2g-2) \cdot f_{max},\ 2g \cdot f_{max})$, which indicates that AJ_g is jamming the channel. And then the authorized device can compute their frequency offset Δf_g and find out AJ_g's key k_g which can be used to generate the jamming signal sequences used by AJ_g, say $s_g(1), s_g(2), \ldots, s_g(n)$.

Since the received signals Y contain the ally jamming signals from multiple ally jammers, we cannot compensate for AJ_g's frequency offset on Y directly without disrupting other ally jammers' frequency offsets. To address this problem, the authorized device applies Δf_g on $s_g(1), s_g(2), \ldots, s_g(n)$ to mimic the same frequency offset effect. Then it can correlate the frequency offset compensated $s_g(1), s_g(2), \ldots, s_g(n)$ with Y to synchronize with the ally jammer AJ_g. Thus by finding out all the pilot frequency spikes on spectrum and repeating this process n times, the authorized device is able to synchronize with all ally jammers.

2) Authorized Transmission Detection & Recovery: The detection of the authorized transmission under multiple ally jammers is similar to the detection under single ally jammer: when there is no authorized transmission, the estimated channels between these multiple ally jammers and the authorized device tend to be stable in short period (e.g., several milliseconds).

In the previous n active ally jammers example, the authorized device can get sample $y(k)$ which contains ally jamming signals from all n ally jammers. As the frequency offsets have already been compensated for, we have

$$y(k) = \sum_{g=1}^{n} c_g \cdot s_g(k) + n_0(k), k \in [1, n],$$

where $c_g = h_g e^{j\gamma_g}$ is the channel coefficient between the ally jammer AJ_g and the authorized device, $s_g(k)$ is the jamming signal sent by the ally jammer AJ_g and $n_0(k)$ is the white noise in received sample $y(k)$. Assuming $\mathbf{y} = [y(1), y(2), \ldots, y(n)]^T$, $\mathbf{s_g} = [s_g(1), s_g(2), \ldots, s_g(n)]^T$ and $\mathbf{n_0} = [n_0(1), n_0(2), \ldots, n_0(n)]^T$, we have

$$\mathbf{y} = [\mathbf{s_1}\ \mathbf{s_2}\ \ldots\ \mathbf{s_n}] \cdot \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix} + \mathbf{n_0}.$$

The distribution of the noise $\mathbf{n_0}$ is known, and we know all the transmitted ally jamming signals $[\mathbf{s_1}\ \mathbf{s_2}\ \ldots\ \mathbf{s_n}]$. Thus the LS estimator can be used to solve the above equation and get the estimated channel coefficients

$$[c_1\ c_2\ \dots\ c_n]^T = (S^H S)^{-1} S^H \mathbf{y},$$

where $S = [\mathbf{s_1}\ \mathbf{s_2}\ \dots\ \mathbf{s_n}]$, $()^H$ denotes the conjugate transpose and $()^{-1}$ is the matrix inverse operation. The authorized device can use different received signals to compute multiple versions of $[c_1\ c_2\ \dots\ c_n]$ and further compute the standard deviation of each channel coefficient. If the mean value of all these standard deviations is larger than a threshold, then an authorized transmission is detected, the authorized device should start to remove the ally jamming signals.

By detecting the authorized transmission, the authorized device knows whether the received signals contain authorized transmission signals or not. Therefore, it can use the transmission-free samples to estimate the channel coefficients $[c_1\ c_2\ \dots\ c_n]$, then apply these channel coefficients to estimate the received ally jamming signals in the received collided signals and finally subtract them out to recover the detected transmission.

F. Dealing with Multiple Authorized Transmitters

In practice, it is possible that multiple authorized transmitters exist in the network. Since ally jamming signals will always occupy the channel, the traditional media access control (MAC) protocol (e.g., CSMA/CA) for wireless networking cannot be applied. It turns out that the transmission detection techniques can be used to solve this problem.

Before sending any packets, the authorized transmitter listens to the channel and computes the channel coefficients between itself and the multiple ally jammers by using the techniques described in Section IV-E. Suppose that there are n ally jammers and the computed channel coefficients are $[c_1\ c_2\ \dots\ c_n]$. If $[c_1\ c_2\ \dots\ c_n]$ are stable for sometime (e.g., DIFS), then there is no other ongoing transmissions and the authorized transmitter will start to transmit, otherwise, it will back-off for some random time, listen to the channel and compute $[c_1\ c_2\ \dots\ c_n]$ again.

V. Analysis

In this section, we provide an analysis of the proposed ally friendly jamming technique, including ally jamming power control and the limitation discussion.

Let us first clarify the notations. We denote the power of received ally jamming signals, the power of a received transmission (from either an authorized or unauthorized transmitter), and the power of received noise as J, R, and N_0, respectively. The jamming to signal power ratio at the receiver side is $JSR = \frac{J}{R}$, the Signal to Noise Ratio is $SNR = \frac{R}{N_0}$. For simplicity, we assume authorized and unauthorized receivers observe the same received ally jamming powers and the same received transmission powers.

A. Maintaining Authorized Communication

We would like to understand how well the authorized communication can be maintained through analyzing the Bit Error Rate (BER) at authorized devices. According to [11], the BER of a wireless device is mainly dependent on its Signal to Interference and Noise Ratio (SINR) and the modulation method.

Let x be the portion of the ally jamming signal power that can be removed using our techniques. Consider the situation where the authorized devices use BPSK for modulation. Based on the result in [11], we can derive the BER as

$$P_e^a = Q\left\{\sqrt{\frac{2}{\frac{1}{SNR} + JSR(1-x)}}\right\},$$

where $Q(\cdot)$ is the Q-function (i.e., $Q(x)$ is the probability that a standard normal random variable will obtain a value larger than x). Figure 10 (a) gives the BER values w.r.t. x and JSR, where $\frac{1}{SNR}$ can be ignored as SNR is high enough. The results for other modulation methods can be derived similarly.

In our experiments, the percentage of removed jamming power x is between 99.2% and 99.6% (See Figure 14). It is generally agreed that wireless communication can be well maintained when the BER is less than 10^{-3} [13]. This implies that we can maintain authorized wireless communication even if the JSR is as high as $17dB$.

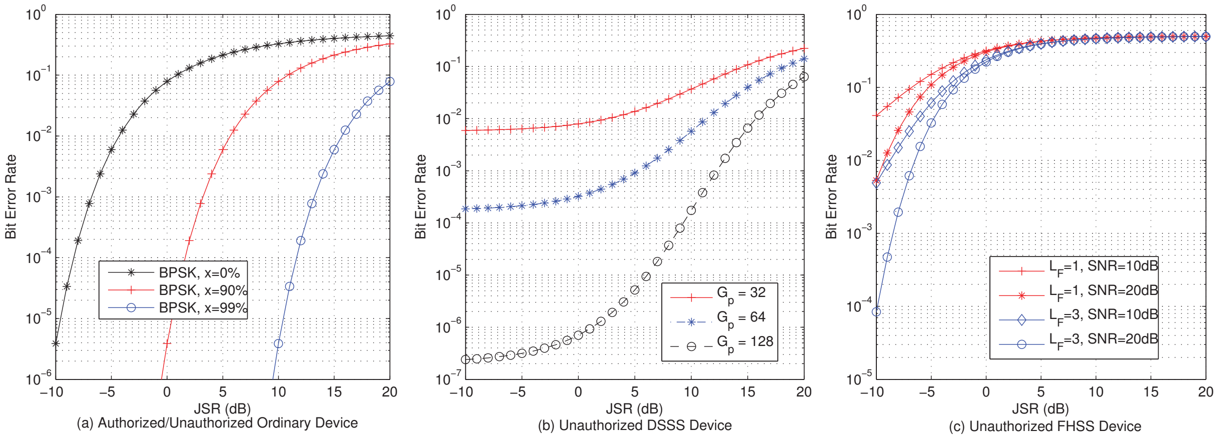
B. Disabling Unauthorized Communication

We consider three kinds of unauthorized devices: ordinary ones that do not use any anti-jamming techniques, those with DSSS-based anti-jamming capability, and those with FHSS-based anti-jamming capability.

1) Ordinary Unauthorized Devices: Unauthorized devices do not know the secret keys, and thus cannot regenerate the ally jamming symbols and remove them from the received signals. An ordinary unauthorized device may attempt to guess the jamming symbols to remove the jamming signals. Note that the random generation of the ally jamming symbols is essentially to randomly pick points from the constellation map. Even assuming a coarse-grained random generation with only 10 possibilities for the real and the imaginary parts of a random jamming symbol, there are 10^2 possible symbols in total. The probability of guessing y consecutive symbols right will be 10^{-2y}, which quickly approaches 0 when y increases. Thus, the probability of removing the ally jamming signals through random guessing is very close to 0.

Based on the results in [11], if BPSK is used for modulation, the BER for an unauthorized device is

$$P_e^o = Q\left\{\sqrt{\frac{2}{\frac{1}{SNR} + JSR}}\right\}.$$



(a) Authorized/Unauthorized Ordinary Device

(b) Unauthorized DSSS Device

(c) Unauthorized FHSS Device

Figure 10. Bit error rate analysis.

The BER for other modulation methods can be derived similarly. Again assuming that the SNR is high enough, $\frac{1}{SNR}$ can be ignored, we can get the BER as shown in Figure 10 (a), in which the line for $x = 0\%$ shows the expected BER for an unauthorized device when BPSK is used for modulation. It is easy to see that when the jamming signal is $10dB$ stronger than the power of a transmission, the BER of the unauthorized device is close to 50%, a value obtainable with random guesses, and their communication is disabled.

2) DSSS-based Unauthorized Devices: To jam DSSS-based unauthorized devices, the ally jammer needs to act as a broadband jammer [31] by increasing its symbol rate and injecting jamming signals with a bandwidth approximately the same as the DSSS signals from unauthorized devices. Assuming the spreading code length of unauthorized devices is G_p and BPSK is used for modulation, according to [31], we can estimate the BER of a DSSS-based unauthorized device under ally jamming as

$$P_e^d = Q\left\{\sqrt{\frac{2G_p}{\frac{1}{SNR} + JSR}}\right\}.$$

Figure 10 (b) shows the BER when SNR= $-10dB$. It indicates that to disrupt the reception at an unauthorized receiver, the jamming signal must overcome the processing gain of spreading in DSSS. The result is consistent with the situation when ally friendly jamming is not used.

3) FHSS-based Unauthorized Devices: To jam FHSS-based unauthorized devices, the ally jammer needs to use broadband jamming to make sure the jamming signals are strong enough on all hopping channels. Assuming a fast hopping system, the probability that the unauthorized device fails to receive the transmission in one hop is $P_{e_k} = \frac{1}{2}exp(-\frac{1}{2(\frac{1}{SNR}+JSR)})$. According to [31], the BER of the FHSS communication under ally jamming is

$$P_e^f = 1 - \sum_{k=\lfloor \frac{L_F}{2} \rfloor + 1}^{L_F} \binom{L_F}{k} [P_{e_k}]^{L_F - k}(1 - P_{e_k})^k,$$

where L_F is the number of hops per data bit.

Figure 10 (c) illustrates the jamming performance against FHSS-based unauthorized devices. It is clear that when the JSR increases, the BER of FHSS-based unauthorized devices reaches 50% quickly and the communication is disabled.

C. JSR Trade-off

Maintaining authorized communication and disabling unauthorized communication have different requirements for JSR. JSR needs to be large to obtain effective jamming against unauthorized communication, but at the same time, JSR cannot be too large to affect authorized communication. Assuming that the BER of authorized devices should be at most $P_e^{a,u}$, and the BER of unauthorized devices should be at least $P_e^{o,l}$ to disable their communication. Based on the earlier analysis, we can conclude that in order to maintain authorized communication and disable ordinary unauthorized devices, the JSR should be in the following range:

$$[(\frac{2}{(Q^{-1}(P_e^{o,l}))^2} - \frac{1}{SNR}), \frac{1}{1-x}(\frac{2}{(Q^{-1}(P_e^{a,u}))^2} - \frac{1}{SNR})].$$

For unauthorized devices using DSSS or FHSS, the jamming performance also depends on their processing gains besides JSR. When the processing gain is high enough, the ally jammer may not find a usable JSR to both allow authorized communication and disable unauthorized ones. However, authorized devices can also use anti-jamming techniques such as DSSS and FHSS. As a result, the JSR upper bound derived earlier can be significantly increased to allow effective jamming of unauthorized devices with anti-jamming capabilities.

D. Limitations

Ally friendly jamming provides us a desirable capability: disabling unauthorized wireless communication while still maintaining authorized wireless communication. This paper may be viewed as the first step toward this goal. Several problems remain open for future works.

Fast Identification of Ally Jammers: Ally friendly jamming uses pilot frequencies for fast identification of ally jammers, which may introduce potential vulnerabilities. The attacker can inject or replay pilot frequency signals to mislead the authorized receiver's synchronization process. Therefore, a more robust fast identification approach deserves further investigations.

Fast Synchronization: Shifting correlation based synchronization used by the authorized receiver is expensive in computation, and may have scalability issues, especially when the sample size and/or the number of ally jammers are large. Thus, a more computational efficient synchronization approach is desirable.

Ally Friendly Jamming with MIMO Devices: To make ally friendly jamming suitable for MIMO devices, we need to consider authorized/unauthorized MIMO devices (e.g., TX, RX) and MIMO ally jammers. One possible way of extending the current approach to the MIMO ally jammer case is: using a different key to generate jamming signals on each of the transmit paths of a MIMO ally jammer, and let the authorized receiver treat the MIMO ally jammer as multiple ally jammers. More studies are required for authorized/unauthorized MIMO devices cases.

Handing Adversarial Jamming: Authorized devices can use the anti-jamming techniques (e.g., DSSS and FHSS) to suppress the adversarial jamming signals after removing the ally jamming signals, which calls for efforts on the integration of ally friendly jamming and the anti-jamming techniques.

VI. Implementation and Evaluation

We have implemented an "off-line processing" based prototype based on GNURadio and USRP. In the following of this section, we will give the implementation details and the corresponding evaluation results.

A. Experiment Setup

The prototype system consists of two ally jammers AJ_1 and AJ_2, a transmitter, and a receiver. Each of them is implemented by a USRP N210 board connected to a laptop. Each USRP N210 uses a XCVR2450 daughter board operating in the 2.4GHz range as the RF front end. The receiver acts as an authorized device by using the techniques in ally friendly jamming to synchronize and remove the ally jamming signals, and as an unauthorized device by directly demodulating the received signals. Our prototype implementation uses both GNURadio and MATLAB for signal processing. The USRP N210 uses a 2.5 PPM [3] temperature-compensated crystal oscillator (TCXO) as its frequency reference [4], the frequency drift is within [-6KHz, $+6$KHz] (2.4GHz $\cdot 2.5$ PPM= 6KHz). Therefore, the maximum frequency offset $f_{max} = 12$KHz, and the pilot frequencies for AJ_1 and AJ_2 are 12KHz and 36KHz, respectively.

The experiments contain three steps as described below. First, we use a PRNG with two different keys to generate the random floating point numbers with precision of 0.1 and uniformly distributed within $[-1, 1]$, which are then used to form the ally jamming symbols for AJ_1 and AJ_2 respectively.

Second, we keep the transmitter silent, turn on the receiver and let two ally jammers emit the ally jamming symbols simultaneously with the same transmit power. Ally jammers are about 2 meters away from the receiver. The ally jammer's symbol rate is 5×10^5 sps (symbols per second). The receiver samples the channel at 10^6 sps and dumps the received samples in a file for the subsequent off-line processing. The samples collected in this step will be referred to as the *TX Off Samples*.

Third, we start the transmitter, which uses DBPSK modulation and sends packets with the length of $1,500$ bytes at a data rate of 500kb/s. The interval between packets is 15ms. Ally jammers and the transmitter are about 2 meters away from the receiver and they all use the default transmit power with the same transmit gain. Ally jammers are still jamming the channel and the receiver still records the received samples in a file. The collected samples are termed as the *TX On Samples*.

B. Evaluation Methodology

The experimental evaluation consists of two parts: *micro-evaluation* and *macro-evaluation*. In micro-evaluation, we evaluate the performance of critical techniques used in ally friendly jamming. In macro-evaluation, we compare the bit error rates and packet loss rates for authorized and unauthorized devices under ally friendly jamming, including the case where unauthorized devices use DSSS for anti-jamming communication.

C. Micro-Evaluation

1) Synchronization: The authorized receiver does spectrum analysis on the *TX Off Samples* using FFT. Figure 11 shows the result on frequency domain when 10000 samples is used for FFT, from which we can clearly see that there is a spike at 7.9KHz, and another one at 32.7KHz. As 7.9KHz is within $[0, 24\text{KHz})$ and 32.7KHz is within [24KHz, 48kHz), the authorized receiver knows that AJ_1 and AJ_2 are jamming the channel.

After identifying ally jammers, the authorized receiver computes their frequency offsets, compensates for the frequency offsets on the locally generated symbols and correlates with the received jamming signals to synchronize with both AJ_1 and AJ_2. As shown in Figure 12, there is a

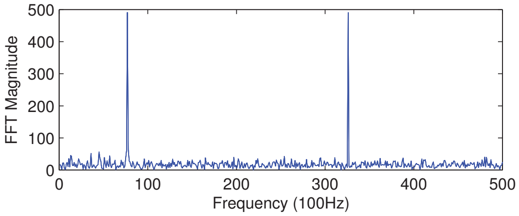
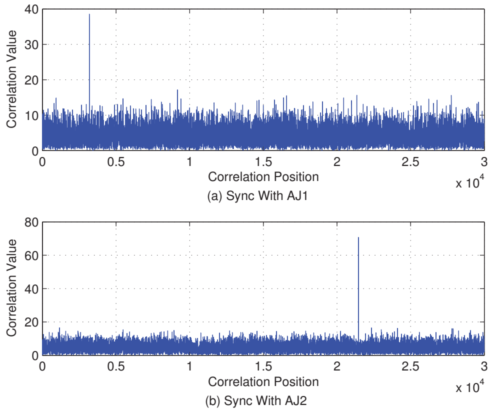


Figure 11. Identifying ally friend jammers.



(a) Sync With AJ1

(b) Sync With AJ2

Figure 12. Synchronizing with multiple ally jammers. The correlation length is 1000 samples.

correlation peak for AJ_1 at position 3190, which means that the timing offset between AJ_1 and the authorized receiver is $3190 \cdot T_p$, where T_p is the sampling interval. The authorized receiver can use this offset to synchronize with the ally jammer AJ_1. Similarly, there is another correlation peak for AJ_2 at position 22459. The authorized receiver can use the same process to synchronize with AJ_2.

We repeat this experiment 1,000 times with different samples. By using the correlation peak position as the indicator of timing offset, the success rate of synchronization is 100%. We also measure the time required for initial synchronization. It takes about 3 seconds for correlating 10^6 samples with a correlation length of 10^3 samples. After the initial synchronization, the re-synchronization takes less than 1 ms. Note that timing experiments are conducted on a laptop with an i7-2760QM CPU. The required time will be shorter on a dedicated radio chip. All of these experiments demonstrate that the authorized receiver can accurately synchronize with ally jammers.

2) Detecting Transmissions under Ally Jamming: In this experiment, we examine how well the authorized transmission can be detected under ally jamming by using the *TX On Samples*. Since the packet length is 1500 bytes and the rate is 500kb/s, the packet transmission time is 24 ms. We set the re-synchronization interval as 10 ms. We adjust the transmit and receive gains such that the JSR is $5dB$, $10dB$, and $15dB$, respectively, which are in the JSR trade-off range shown in Section V-C. Then we examine the true positive and false positive rates of transmission detection for different thresholds on the standard deviation of the estimated channel coefficients. Figure 13 shows the result of the experiment. It is easy to see that there is a range of threshold values that allow the transmissions to be detected almost 100% with close-to-0 false positive rate. In other words, the detection of transmissions under ally jamming can be performed very precisely.

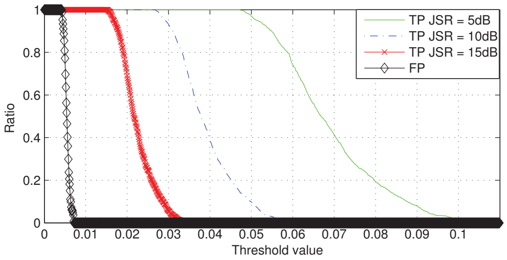


Figure 13. Transmission detection rate. FP is the false positive rate, TP is the true positive rate.

3) Removal of Ally Jamming Signals: We want to know how well the authorized device can estimate and remove ally jamming signals when only ally jamming signals are received. We use the *TX Off Samples* collected when one and two ally jammers are on, respectively. After synchronization, we use the first 1000 samples to estimate the channel(s), predict the ally jamming signals in the following received samples, and then subtract them out from the received samples to check how much ally jamming power remains.

In our experiment, the percentage of jamming power removed by the authorized receiver depends on how many ally jamming samples we need to the estimate. Intuitively, as channel changes over time, if we apply the same estimated channel coefficients to estimate too many samples, the quality of estimation will degrade, and less jamming power will be removed. Figure 14 shows that the authorized device can remove 99.2% to 99.6% ally jamming power when the length of the estimated samples increases from $1,000$ to $14,000$. In other words, the vast majority of the ally jamming signal power can be effectively removed.

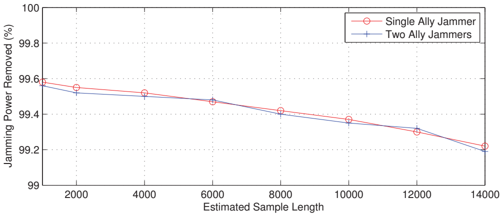
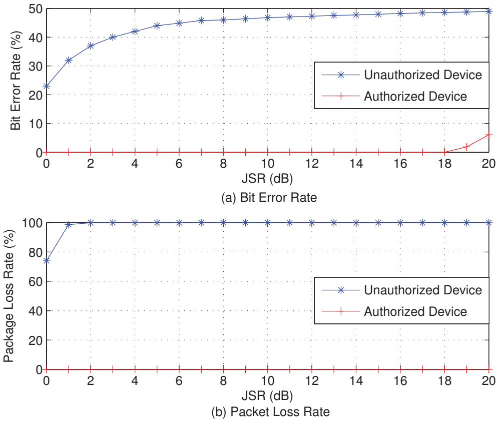


Figure 14. Removal of ally jamming signals.

D. Macro-Evaluation

The *TX On Samples* are used here. We adjust the transmitter's gain and ally jammers' gains to achieve different

JSRs. The authorized receiver first detects the transmissions, recovers the transmitted signals, and then streams them into the demodulation blocks. In contrast, the unauthorized receiver demodulates the received samples directly.



(a) Bit Error Rate

(b) Packet Loss Rate

Figure 15. Macro-evaluation.

Figure 15 (a) shows the BER for both authorized and unauthorized devices. It can be seen that as the JSR increases, the BER of the unauthorized receiver quickly increases to about 50%, a value achievable with random guesses. In contrast, with the ally jamming signals removal techniques, the authorized receiver can maintain close to 0 BER until the JSR exceeds 17dB. We use the GNURadio benchmark receiver to evaluate the overall packet loss rate. Figure 15 (b) shows the packet loss rates for both authorized and unauthorized receivers. Again, when the JSR increases, the packet loss rate at the unauthorized receiver quickly reaches 100%, while the packet loss rate at the authorized one remains close to 0 until the JSR reaches 16 dB. Unauthorized devices can certainly try to use Error Correction Code (ECC) to tolerate errors. However, with close to 50% BER, it is unlikely to reduce the packet loss rate much.

We also perform some preliminary evaluation of ally friendly jamming against unauthorized devices that are equipped with DSSS-based anti-jamming capability. In this experiment, we use IEEE 802.11b protocol running at 1 Mbps on unauthorized devices, which uses DSSS with an 11-bit barker code for spreading and despreading [29]. More specifically, we use two laptops with 802.11b wireless adapters operating at the DSSS mode as unauthorized devices. We use another laptop connected to a USRP N210 board as the ally jammer. All these three devices are about 2 meters away from each other. We set the USRP using 2.452GHz frequency and the 802.11b wireless adapters using the same frequency (i.e., channel 9). We adjust the ally jammer's gain to make sure it has the same transmit power with the 802.11b transmitter. We test the packet loss rate at the 802.11b receiver side when different jamming symbol rates are used. (Note that higher symbol rates will cover wider spectrum.) Figure 16 shows that when the symbol rate for the ally jammer is more than $600ksps$, the communication between these 802.11b DSSS devices is disabled.

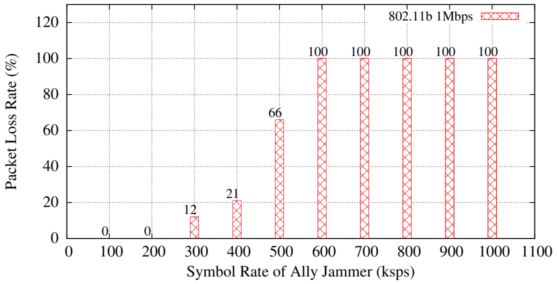


Figure 16. Jamming DSSS devices (ksps: kilo symbols per second).

Note that though 802.11b DSSS mode is designed for wireless communication under interference, it is not intended as a strong anti-jamming solution. More in-depth evaluation is necessary to understand the performance of ally friendly jamming against powerful anti-jamming communication schemes.

VII. Related Work

IMD Shield [12] is the most closely related work to ours. As discussed in the introduction, IMD Shield cannot achieve ally friendly jamming. We do not repeat it here.

Our work in this paper is in general related to research on interference cancellation and suppression. Zigzag recursively applies interference cancellation to get the interference free signals from colliding ones [13]. Another Interference Alignment and Cancellation (IAC) technique was proposed to enable collaborative Access Points (APs) in MIMO LANs to decode more packets by controlling transmitted signals with proper vectors [14]. 802.11n$^+$ was proposed to use "antidote" signals to nullify the transmitted signals from other nodes in order to enable multiple access to wireless channels [19]. An implementation of successive interference cancellation (SIC) for ZigBee on software radios was presented in [15] which can decode concurrently transmitted packets. Moreover, SAM [41] provides a chain-decoding technique to decode concurrent frames. All these techniques assume regular modulated signals are transmitted and perform interference cancellation accordingly. Unfortunately, when the ally jamming signals mimic random noises, none of them can be used due to the challenges in synchronization and channel estimation. Our proposed techniques have addressed these issues and advanced interference cancellation techniques to the next level.

Ally friendly jamming is also related to wireless jamming and anti-jamming research. For friendly jamming studies, Sankararaman et al. studied strategies of allocating friendly jammers to create wireless barriers which can prevent the eavesdropping [36]. There are also other literature (e.g., [27], [35], [44]) using friendly jamming to block the responses or unauthorized queries to protect particular wireless devices. For jamming and anti-jamming techniques, jamming attack models and several ways to detect jamming attacks have been studied in [45]. Game theoretical models have been developed for jamming and jamming defense [37], [46]. Spread spectrum techniques such as DSSS and FHSS have been traditionally used for anti-jamming wireless communication. In recent years, researchers have identified some weaknesses of such schemes due to shared keys and developed enhanced schemes, including Uncoordinated FHSS and its variations (e.g., [20], [38]–[40]), Uncoordinated DSSS and its variations (e.g., [21], [24], [32], [33]), and novel coding techniques (e.g., [5], [42]). Several filter designing jamming mitigation techniques have also been proposed [8], [9]. All these works are complementary to our results in this paper.

VIII. Conclusion

In this paper, we presented ally friendly jamming, a mechanism that jams unauthorized wireless communication and maintains legitimate communication at the same time. Ally friendly jamming is achieved by properly controlling the ally jamming signals using secret keys shared among authorized devices and the ally jammers. We have analyzed the properties of ally friendly jamming, implemented a prototype system, and performed a series of experimental evaluation. Our results demonstrated that the proposed techniques can effectively disable unauthorized wireless communication and at the same time allow wireless communication between authorized devices.

Our future work includes enhancing the robustness of the ally friendly jamming technique, investigating its capability against unauthorized anti-jamming devices and defending against adversarial jamming attacks.

Acknowledgment

The authors would like to thank our shepherd, Srđan Čapkun, and the anonymous reviewers for their insightful comments. We also thank Yao Liu, Chengzhi Li, Huazi Zhang, Yajin Zhou, Entong Shen, Ruowen Wang, Jason Gionta, Xianqing Yu, Quan Chen and Syed Hussain for helpful discussions. This work was supported in part by the US National Science Foundation (NSF) under Grants CNS-1016260 and ECCS-1002258. The opinions and conclusions in this paper are those of the authors and do not necessarily reflect the views of the NSF.

References

[1] GNU Radio - The GNU Software Radio. http://gnuradio.org/redmine/projects/gnuradio/wiki.

[2] Improvised explosive device - wikipedia. http://en.wikipedia.org/wiki/Improvised_explosive_device.

[3] PPM. http://en.wikipedia.org/wiki/Parts_per_million.

[4] USRP N210 Datasheet. https://www.ettus.com/content/files/07495_Ettus_N200-210_DS_Flyer_HR_1.pdf.

[5] L. C. Baird, W. L. Bahn, M. D. Collins, M. C. Carlisle, and S. C. Butler. Keyless jam resistance. In *IEEE Information Assurance and Security Workshop*, 2007.

[6] L. Buttyan, D. Gessner, A. Hessler, and P. Langendoerfer. Application of wireless sensor networks in critical infrastructure protection: challenges and design options. *IEEE Wireless Communications*, 17(5), 2010.

[7] H. Chan, A. Perrig, and D. Song. Random key predistribution schemes for sensor networks. In *IEEE Symposium on Security and Privacy*, 2003.

[8] B. DeBruhl and P. Tague. Digital filter design for jamming mitigation in 802.15.4 communication. In *ICCCN*, 2011.

[9] B. DeBruhl and P. Tague. Mitigation of periodic jamming in a spread spectrum system by adaptive filter selection. In *PECCS*, 2012.

[10] M. Erol-Kantarci and H.T. Mouftah. Wireless multimedia sensor and actor networks for the next generation power grid. *Ad Hoc Networks*, 2011.

[11] A. Goldsmith. *Wireless communications*. Cambridge University Press, 2005.

[12] S. Gollakota, H. Hassanieh, B. Ransford, D. Katabi, and K. Fu. They can hear your heartbeats: non-invasive security for implantable medical devices. In *SIGCOMM*, 2011.

[13] S. Gollakota and D. Katabi. Zigzag decoding: Combating hidden terminals in wireless networks. In *SIGCOMM*, 2008.

[14] S. Gollakota, S.D. Perli, and D. Katabi. Interference alignment and cancellation. In *SIGCOMM*, 2009.

[15] D. Halperin, T. Anderson, and D. Wetherall. Taking the sting out of carrier sense: interference cancellation for wireless LANs. In *MobiCom*, 2008.

[16] S.S. Haykin. *Digital communications*, volume 5. Wiley, 1988.

[17] Y. Kim, A. Perrig, and G. Tsudik. Simple and fault-tolerant key agreement for dynamic collaborative groups. In *CCS*, 2000.

[18] B. Latre, B. Braem, I. Moerman, C. Blondia, and P. Demeester. A survey on wireless body area networks. *Wireless Networks*, 17(1), 2011.

[19] K.C. Lin, S. Gollakota, and D. Katabi. Random access heterogeneous MIMO networks. In *SIGCOMM*, 2011.

[20] A. Liu, P. Ning, H. Dai, and Y. Liu. USD-FH: Jamming-resistant wireless communication using frequency hopping with uncoordinated seed disclosure. In *MASS*, 2010.

[21] A. Liu, P. Ning, H. Dai, Y. Liu, and C. Wang. Defending DSSS-based broadcast communication against insider jammers via delayed seed-disclosure. In *ACSAC*, 2010.

[22] D. Liu, P. Ning, and R. Li. Establishing pairwise keys in distributed sensor networks. *ACM Transactions on Information and System Security*, 2005.

[23] D. Liu, P. Ning, and K. Sun. Efficient self-healing group key distribution with revocation capability. In *CCS*, 2003.

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

[25] Ettus Research LLC. The USRP Product Family Products and Daughter Boards. http://www.ettus.com/products.

[26] R.G. Lyons. *Understanding digital signal processing*. Prentice Hall, 2011.

[27] I. Martinovic, P. Pichota, and J. B Schmitt. Jamming for good: a fresh approach to authentic communication in WSNs. In *WiSec*, 2009.

[28] H. Meyr, M. Moeneclaey, and S.A. Fechtel. *Digital communication receivers : synchronization, channel estimation, and signal processing*. John Wiley & Sons, 1998.

[29] K. Pahlavan and P. Krishnamurthy. *Principles of wireless networks*. Prentice Hall, 2001.

[30] A. Perrig, D. Song, and J.D. Tygar. ELK, a new protocol for efficient large-group key distribution. In *IEEE Symposium on Security and Privacy*, 2001.

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

[32] C. Pöpper, M. Strasser, and S. Čapkun. Jamming-resistant broadcast communication without shared keys. In *USENIX Security Symposium*, 2009.

[33] C. Pöpper, M. Strasser, and S. Čapkun. Anti-jamming broadcast communication using uncoordinated spread spectrum techniques. *JSAC*, 2010.

[34] J.G. Proakis and M. Salehi. *Digital communications*. McGraw-hill, 2008.

[35] M. Rieback, B. Crispo, and A. Tanenbaum. RFID guardian: A battery-powered mobile device for rfid privacy management. In *Information Security and Privacy*. Springer, 2005.

[36] S. Sankararaman, K. Abu-Affash, A. Efrat, S. D. Eriksson-Bique, V. Polishchuk, S. Ramasubramanian, and M. Segal. Optimization schemes for protective jamming. In *MobiHoc*, 2012.

[37] D. Slater, P. Tague, R. Poovendran, and M. Li. A game-theoretic framework for jamming attacks and mitigation in commercial aircraft wireless networks. In *AIAA Infotech@Aerospace Conference*, 2009.

[38] D. Slater, P. Tague, R. Poovendran, and B. Matt. A coding-theoretic approach for efficient message verification over insecure channels. In *WiSec*, 2009.

[39] M. Strasser, C. Pöper, S. Čapkun, and M. Čagalj. Jamming-resistant key establishment using uncoordinated frequency hopping. In *IEEE Symposium on Security and Privacy*, 2008.

[40] M. Strasser, C. Pöpper, and S. Čapkun. Efficient uncoordinated FHSS anti-jamming communication. In *MobiHoc*, 2009.

[41] K. Tan, H. Liu, J. Fang, W. Wang, J. S. Zhang, M. Chen, and G. M. Voelker. SAM: Enabling practical spatical multiple access in wireless LAN. In *MobiCom*, 2009.

[42] D. Willkomm, J. Gross, and A. Wolisz. Reliable link maintenance in cognitive radio systems. In *DySPAN*, 2005.

[43] C. K. Wong, M. G. Gouda, and S. S. Lam. Secure group communications using key graphs. In *SIGCOMM*, 1998.

[44] F. Xu, Z. Qin, C. C Tan, B. Wang, and Q. Li. IMDGuard: Securing implantable medical devices with the external wearable guardian. In *INFOCOM*, 2011.

[45] W. Xu, W. Trappe, Y. Zhang, and T. Wood. The feasibility of launching and detecting jamming attacks in wireless networks. In *MobiHoc*, 2005.

[46] Q. Zhu, H. Li, Z. Han, and T. Basar. A stochastic game model for jamming in multi-channel cognitive radio systems. In *ICC*, 2010.