No Time to Demodulate - Fast Physical Layer Verification of Friendly Jamming

Wenbo Shen*, Yao Liu[†], Xiaofan He*, Huaiyu Dai* and Peng Ning*
* North Carolina State University, {wshen3, xhe6, hdai, pning}@ncsu.edu
[†] University of South Florida, yliu@cse.usf.edu

***Abstract*—Jamming attacks are well-known threats to wireless communications, but on the other hand they provide insights for researchers to design novel approaches to protect wireless communications. In recent years, friendly jamming is used by a number of research works to achieve the wireless medium access control. However, in these works, the friendly jammer relies on bit-level information to distinguish the allies' wireless transmissions from the enemies', which requires the received signals to be processed through demodulation steps and thus introduces a non-trivial reaction time delay for the friendly jammer. This reaction delay is undesirable as the transmissions need to be jammed while they are still on the air.**

To address this problem, we propose *fast friendly jamming*, which eliminates the need for demodulation and enables the friendly jammer to verify the received signals directly on the physical layer. We have implemented a prototype of the proposed techniques based on GNURadio and USRP, and performed real-world experiments to validate the proposed techniques. The experiment results show that the proposed techniques reduce the normal reaction delay of the friendly jammer by $81.9\% - 85.7\%$, and achieve the accurate distinction between allies' and enemies' transmissions.

I. Introduction

Jamming attacks are well-known threats to wireless communications. A jammer uses a radio frequency device to transmit wireless signals. Due to the shared nature of wireless medium, signals of the jammer and the sender collide at the receiver, and the signal reception process is disrupted. On the other hand, jamming attacks provide insights for researchers to design novel approaches to protect wireless communications. Recently, friendly jamming (i.e., intentional signal interference from collaborating devices, denoted as friendly jammers) has been utilized as an effective technique to protect information confidentiality [3] as well as achieve wireless medium access control [6], [10], such as blocking unauthorized transmissions (i.e., unauthorized radio commands) for RFID systems [8], [9] and implantable medical devices [3], [15].

In this paper, we focus on using friendly jamming for wireless medium access control. To achieve wireless medium access control, the friendly jammer needs to block unauthorized wireless transmissions and avoid jamming the authorized ones mistakenly. In other words, the friendly jammer needs to identify the on-going wireless transmission first, and keeps silent if it is authorized or launches jamming attacks otherwise.

To achieve effective jamming, the friendly jammer needs to identify and jam an unauthorized wireless transmission while the transmission is still on the air. Thus the reaction time is crucial to the jamming performance. Previous friendly jamming studies (e.g., [3], [8], [9], [13]–[15]) proposed to distinguish wireless transmissions by using bit-level information, such as matching certain patterns in the message bits. However, in order to obtain the message bits, the friendly jammer needs to perform signal demodulation, which normally involves cascading steps, such as frequency offset compensation, symbol synchronization, and constellation decoding. These steps impose a non-trivial time delay for the friendly jammer, and thus the friendly jammer may fail to identify the unauthorized transmissions in a timely manner. Therefore, to reduce the reaction time, we propose *fast friendly jamming*, which eliminates the need of demodulation and verifies the signals directly on the physical layer.

The basic idea of fast friendly jamming is that the authorized transmitter generates a special preamble (that we named as *auth-preamble*, short for authentication preamble) using a shared secret key and prepends the auth-preamble before the packet transmission (i.e., before the normal preamble). On the other side, the friendly jammer uses the same key to synchronize and verify the auth-preamble of an on-going transmission. If the verification succeeds, the friendly jammer will keep silent; otherwise, the current on-going transmission will be treated as an unauthorized one and the friendly jammer will launch jamming.

Though conceptually simple, two technical challenges need to be solved before achieving fast friendly jamming. First, to eliminate the demodulation steps and allow the direct verification of the auth-preamble on the physical layer, the auth-preamble signals cannot be modulated bits[1]. Moreover, the auth-preamble signals must introduce enough randomness and should be ever changing to prevent the adversary from predicting, mimicking and replaying them. To address these problems, we propose to use the shared secret key and the time to generate signal symbols in the auth-preamble directly.

Second, the auth-preamble signals introduce randomness to defend against the predict, mimic and replay attacks. However, the randomness also brings difficulties for the friendly jammer to verify the auth-preamble. Simple correlation won't work as the frequency offset will distort the correlation results [10]. Traditional synchronization approaches [2], [7] cannot be

[1] In this paper, modulation refers to the base-band modulation, in which bits are mapped to points on a constellation diagram. The modulated base-band signals still need to be up-converted to radio frequency band before sending out from an antenna.

978-1-5090-0073-9/15/$31.00 ©2015 IEEE

applied for the friendly jammer to synchronize with the auth-preamble signals, as the auth-preamble signals are changing continuously and the channel and hardware effects (i.e., channel attenuation, phase shift, and frequency offset) on the received signals are unknown. To address this problem, we propose a novel technique called *amplitude differential based correlation*, which can tolerate the unknown channel and hardware effects on the received signals, thereby allowing the friendly jammer to verify the received signals directly on the physical layer.

The contribution of this paper is two-fold. First, we propose fast friendly jamming as well as the related techniques, which enable the friendly jammer to verify the received signals directly on the physical layer. Second, we have implemented a prototype of the proposed techniques on GNURadio [1] and USRP [4], and performed real-world experiments to validate the proposed techniques. The experiment results show that fast friendly jamming reduces the reaction delay of the friendly jammer by $81.9\% - 85.7\%$, as compared to the traditional demodulation approach. Meanwhile, it enables the accurate distinction between allies' and enemies' transmissions with 100% true positive and 0% false negative rates.

The rest of this paper is organized as follows. Background information is given in Section II. Assumptions and the threat model are presented in Section III. The design of fast friendly jamming is detailed in Section IV, analyzed in Section V and evaluated experimentally in Section VI. Finally, the paper is concluded in Section VII.

II. Preliminaries

Wireless communication aims to transfer information via radio frequency (RF) signals. The wireless transmitter in general modulates message bits into discrete base-band signals (signal symbols) first, then uses the digital to analog converter to convert these discrete signals to analog signals, and up converts them to radio frequency signals [2], and finally sends them out from its antenna.

Discrete base-band signals can be represented as complex numbers and the modulation process is equivalent to the mapping from bits to complex number points on the constellation diagram. A complex number can be represented in its polar form, i.e., a complex number $a + bj = Me^{j\phi}$, where M is its amplitude and ϕ is its angle [5], [11].

RF signals travel through the wireless medium before being received by the receiver. The wireless channel will introduce the attenuation, phase shift, and additional noise to the original transmitted signals. As it is virtually impossible to operate two radios at exact the same frequency, the hardware of the transmitter and receiver will introduce a frequency offset [10]. Considering all these effects, for the transmitted signal $x(i)$, we have the received signal $y(i)$ as

$$y(i) = he^{j\gamma}e^{j2\pi\Delta f t_i}x(i) + n(i),^{2} \tag{1}$$

where h is the channel attenuation, γ is the phase shift, Δf is the frequency offset, t_i is the sampling time and $n(i)$ is the noise.

As the received signals are distorted by channel and hardware effects, the receiver needs to perform certain processes, such as frequency offset compensation and symbol synchronization, to demodulate these signals correctly. These processes not only complicate the receiver design, but also introduce certain delays to the reception process.

III. Assumptions and Threat Model

Assumptions: We assume that all ally devices, including friendly jammers, ally transmitters and ally receivers, are immobile. We also assume that friendly jammers and ally transmitters share a secret key which is unknown to unauthorized devices. As friendly jamming is normally applied for short range wireless communications, we assume that the wireless channels are single-tap and the propagation delay is negligible. We further assume that received signals have a sufficient signal to noise ratio (SNR) so that the friendly jammer can detect both the authorized and unauthorized wireless transmissions. We assume that clocks of all ally devices are loosely synchronized and the maximum clock drift is ΔT. To facilitate the discussion, we assume that there are no adversarial jammers to authorized wireless communications.

Threat Model: The unauthorized devices will try various approaches to maintain their wireless communications under the friendly jamming. They may replay the received legitimate auth-preambles right before their transmissions so that their transmissions can bypass the auth-preamble verification at the friendly jammer. They may hijack the ongoing authorized transmissions by overshadowing the authorized transmission signals with much stronger unauthorized transmission signals right after the legitimate auth-preamble signals. The unauthorized devices may also try to remove the friendly jamming signals to maintain their wireless connections by exploiting advanced digital process techniques, such as the MIMO based attack approach [12].

IV. Fast Friendly Jamming

A. Overview

The friendly jammer uses auth-preamble to distinguish authorized transmissions from unauthorized ones. Fig. 1 shows the overall design of fast friendly jamming. The ally transmitter prepends specially generated auth-preamble signals before its wireless transmission signals. The friendly jammer monitors channels and tries to verify received auth-preamble signals. Transmissions that are not accompanied by valid auth-preambles will fail the verification and be jammed by the friendly jammer. The friendly jammer plays two important roles in this process: (1) it disables the wireless communications between unauthorized transmitters and unauthorized receivers, and (2) it prevents the ally receiver from accepting an unauthorized transmitter's signals.

The friendly jammer relies on auth-preambles to decide its action, the generation and verification of auth-preambles

[2] The equation here is for single-tap channels.

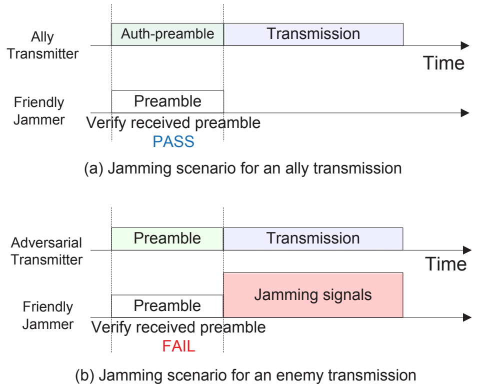


Fig. 1. Jamming scenarios of fast friendly jamming.

are crucial to fast friendly jamming. To realize fast friendly jamming, we propose to generate the auth-preamble signals by using the shared key and the time, and verify the received auth-preamble signals directly instead of demodulating auth-preamble signals into bits.

B. Auth-Preamble Generation

The auth-preamble signals need to be difficult for an adversary to predict, mimic and replay, while they should be easy for the friendly jammer to verify. As mentioned earlier, methods of using modulated bits as auth-preamble signals increase the reaction delay, as the friendly jammer needs to perform the time-consuming demodulation operations. Moreover, the modulated bits also give adversaries opportunities to mimic the auth-preamble signals due to the strong patterns in the modulated signals (e.g., phases). Therefore, we propose to use the shared key and the timing information to control the generation of auth-preamble signal symbols directly.

As shown in Section II, the auth-preamble signal symbols are discrete base-band signals which can be represented by complex numbers. Assuming the auth-preamble contains l signal symbols (complex numbers) and the symbol rate of the ally transmitter is r sps (symbol per second). Upon receiving an up-layer packet at time t_u (in seconds and is a float number), the ally transmitter first uses the shared key and $\lfloor t_u \rfloor$ as the input to a pseudo-random number generator (PRNG) to generate $2r$ floating numbers, denoted as $a(0), a(1), \ldots, a(2r-1)$. Then, floating numbers are used as the real and imaginary parts of r complex numbers, denoted as $x(0), x(1), \ldots, x(r-1)$. Each complex number $x(i)$ can be formed by $x(i) = a(2i) + a(2i+1) \cdot j$, where $i = 0, 1, \ldots, r-1$. Finally, the ally transmitter selects l consecutive complex numbers from the x sequence starting from the $\lfloor f(t_u) \cdot r \rfloor$-th symbol as auth-preamble signals, where $f(t_u)$ is the fractional part of t_u.

The generated auth-preamble signals should be prepended before packet transmission signals. The auth-preamble and packet transmission signals need to be up-converted to RF signals before sending out from the antenna. Note that the final transmission may also contain a normal preamble after the auth-preamble for channel training or synchronization purpose.

C. Auth-Preamble Verification

1) Amplitude Differential based Correlation: The ally transmitter uses the generated pseudo-random complex numbers as auth-preamble signals, which brings challenges for the friendly jammer to verify the received copy of these signals. As mentioned earlier, the auth-preamble signals keep changing and resemble random noise, the channel and hardware effects are unknown. Therefore, traditional approaches cannot be applied for the verification of the auth-preamble signals.

To solve this problem, we propose *amplitude differential based correlation*, which enables the friendly jammer to verify received auth-preamble signals without demodulation. The basic idea is to use the *amplitude ratio* between two consecutive signals to tolerate the channel and hardware effects. For example, assuming that the transmitted auth-preamble signal is $x(i)$ and the corresponding received one is $y(i)$. Observing that $|e^{j\gamma}e^{j2\pi\Delta f t}| = 1$ and the received signals y have sufficient SNR, from (1), we have

$$\begin{aligned} |y(i)| &\approx |he^{j\gamma}e^{j2\pi\Delta f t_i}x(i)| \\ &\approx |hx(i)|. \end{aligned}$$

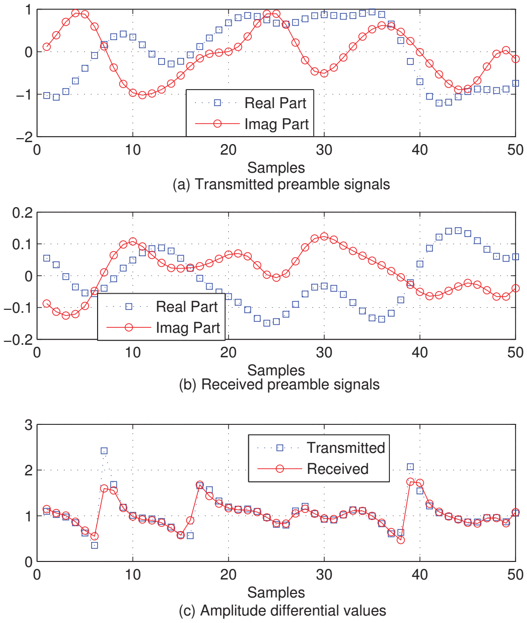
Further observe that in slow fading environments, the change of channel attenuation h over a short period of time (e.g., a few milliseconds) is negligible. We denote the amplitude differential value between two consecutively received signals $y(i)$ and $y(i+1)$ as $AD_{y(i)}$. As the channel is single-tap, we have

$$\begin{aligned} AD_{y(i)} &= |\frac{y(i+1)}{y(i)}| \approx |\frac{hx(i+1)}{hx(i)}| \approx |\frac{x(i+1)}{x(i)}| \\ &\approx AD_{x(i)}, i = 0, 1, \ldots, l-2. \end{aligned} \quad (2)$$

It is easy to see that the amplitude differential values between consecutive signals do not contain the channel and hardware effects (i.e., channel attenuation h, phase shift γ, and frequency offset Δf), and thus the amplitude differential values of the received signals and the corresponding transmitted signals are roughly the same. Fig. 2 shows transmitted and received auth-preamble signals, and their amplitude differential values from our experiment based on GNURadio and USRP. We can see that due to channel and hardware effects, the received auth-preamble signals are very different from the transmitted ones, but their amplitude differential values are very close to each other.

In order to verify the received auth-preamble signals, the friendly jammer first uses the shared key and timing information to generate the auth-preamble signals locally and compute their amplitude differential values beforehand. Then, it computes $AD_{y(0)}, AD_{y(1)}, \ldots, AD_{y(l-2)}$ from the received auth-preamble signals $y(0), y(1), \ldots, y(l-1)$. Finally, the friendly jammer correlates these two amplitude differential sequences to verify the auth-preamble signals. But before correlating, the friendly jammer needs to decide the correlation window for the locally generated signals.

Assuming that the friendly jammer received the auth-preamble signals at time t_a, the auth-preamble duration is T_d. To make sure that the transmitted auth-preamble falls into the correlation window of the locally generated signals,



(a) Transmitted preamble signals

(b) Received preamble signals

(c) Amplitude differential values

Fig. 2. Transmitted auth-preamble signals, received auth-preamble signals and their amplitude differential values.

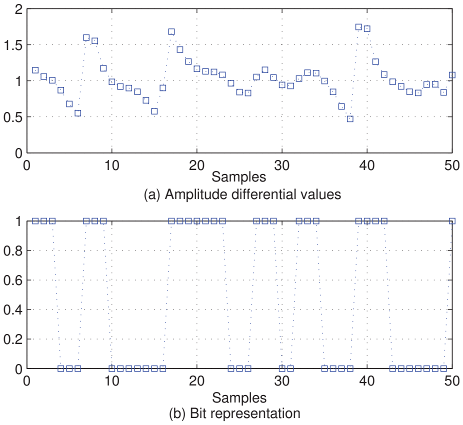
considering the clock drift ΔT, the friendly jammer needs to generate the auth-preamble signals in the time window of $[t_a - \Delta T, t_a + \Delta T + T_d]$, denoted as $g(0), g(1), \ldots, g(m-1)$ and computes their amplitude differential values, denoted as $AD_{g(0)}, AD_{g(1)}, \ldots, AD_{g(m-2)}$.

Both transmitted auth-preamble signals x and locally generated auth-preamble signals g are generated using the same key and g covers the time period of x. Thus, x should be a sub-sequence of g, which means that AD_y is a sub-sequence of AD_g. Therefore, to verify the auth-preamble signals, the friendly jammer does a shift correlation on AD_y with AD_g. Assuming the correlation result starts from $AD_{g(i)}$ is $\Gamma(i)$, we have

$$\Gamma(i) = \sum_{z=0}^{l-2} AD_{g(i+z)} \cdot AD_{y(z)}, i = 0, 1, \ldots, m-l. \tag{3}$$

The correlation result spikes when AD_y is aligned with AD_g correctly. To detect the spike, the friendly jammer imposes a threshold on the difference of the first and the second largest correlation outputs. If the difference is greater than the threshold, a spike is detected and the current transmission passes the verification, the friendly jammer will stay silent until the ongoing transmission finishes. If no spike is found during the correlation process, the verification fails and the current transmission is regarded as an unauthorized one, then the friendly jammer will start to jam the transmission.

2) Efficient Correlation of Amplitude Differential Values: The correlation approach in (3) involves time consuming operations, such as float number multiplications and additions. To reduce the correlation time, we propose to use an approximate but efficient method to perform the correlation between two sequences of amplitude differential values (i.e., AD_y and AD_g). Specifically, we transform each sequence of amplitude differential values into a bitmap by converting each amplitude differential value greater than a threshold to “1” bit (or “0” bit otherwise), as shown in Fig. 3. Given two equal-length bitmaps B_1 and B_2, the correlation process in (3) can be expressed as $\Gamma = |B_1 \wedge B_2|$, where $|B_1 \wedge B_2|$ is the weight (i.e., number of “1”) of bitmap $B_1 \wedge B_2$. Since the correlation between two bitmaps can be computed through bit-wise operations, this method can be executed efficiently.



(a) Amplitude differential values

(b) Bit representation

Fig. 3. Amplitude differential values and their bit representation. The chosen threshold is 1.

V. Analysis

A. Against Different Kinds of Attacks

1) Auth-Preamble Replay Attack: In the replay attack, the adversary records legitimate auth-preamble signals and replays the recorded signals right before its own transmission signals. To avoid collisions, the auth-preamble signals can only be replayed when the current transmission is finished. Therefore, if the authorized transmission duration (including the preamble) T_s achieves that $T_s > 2\Delta T$, the replayed auth-preamble will fall out of the correlation window at the friendly jammer, and the replay attack can be thwarted. The derivation is omitted due to space issue.

2) Auth-Preamble Hijack Attack: In this attack, the adversary will keep monitoring the channels. When detecting an authorized transmission, the adversary sends its transmission signals at a carefully calculated time so that the unauthorized transmission signals append right after the legitimate auth-preamble signals to “overwrite” the authorized transmission signals. However, to “overwrite” the authorized signals, the power of unauthorized transmission must be much stronger than the authorized one. Therefore, the friendly jammer can use the sudden increasing RSSI as an indicator of the auth-preamble hijack attack and jam the transmission accordingly.

3) MIMO Based Attack: Tippenhauer et al. proposed a MIMO based attack to remove the jamming signals from single friendly jammer to reveal the confidential transmission signals [12]. The same technique can also be used to eliminate the friendly jamming signals to recover the unauthorized transmission signals. To remove friendly jamming signals, the adversary's two antennas need to be placed at different locations which are equidistant to the friendly jammer. However, when multiple (> 2) friendly jammers are deployed to monitor the same area in the network, it is impossible to find such two locations that are the same distance away from all friendly jammers. Therefore, multiple jammers can be deployed to defeat this type of attack.

B. *Impact of Multiple Friendly Jammers*

In practice, the ally system may adopt multiple friendly jammers to enhance the jamming performance. For example, the friendly jammers can work together to defeat the earlier mentioned MIMO based attacks, to increase the jamming power against DSSS based unauthorized devices, and to jam more channels collaboratively to defeat FHSS based unauthorized devices. Note that multiple friendly jammers can be deployed easily, because they do not interfere with the authorized transmissions.

C. *Communication Overhead*

In fast friendly jamming, the auth-preamble signals introduce additional communication overhead. Assume that the auth-preamble has l symbols, the packet payload has n bytes, and b bits are modulated in one symbol, which means the payload has $8n/b$ symbols. The overhead rate $\psi = lb/8n$.

In our experiment, the 128-symbol auth-preamble allows the friendly jammer to distinguish authorized transmissions from unauthorized ones accurately. For a packet of 1500 bytes with BPSK modulation, the corresponding communication overhead is 1.06%.

VI. Experimental Evaluation

A. *Experiment Setup*

Our implementation is based GNURadio and N210 USRP. The prototype system contains a transmitter, a receiver, and a friendly jammer. Each node is a USRP connected to a host PC running GNURadio. We use XCVR2450 daughter boards operating on 2.4 GHz as the RF front ends.

For the software parameter configuration, the transmitter generates pseudo-random float numbers with precision of 0.1 and uniformly distribution between $[-1, 1]$, then uses these floating numbers to form auth-preamble signals, as described in Section IV-B. The packet payload length is 1500 bytes. BPSK is used for payload modulation. The bit rate is $250kbps$, and sample per symbol is 4. Our implementation uses both GNURadio and MATLAB for signal processing.

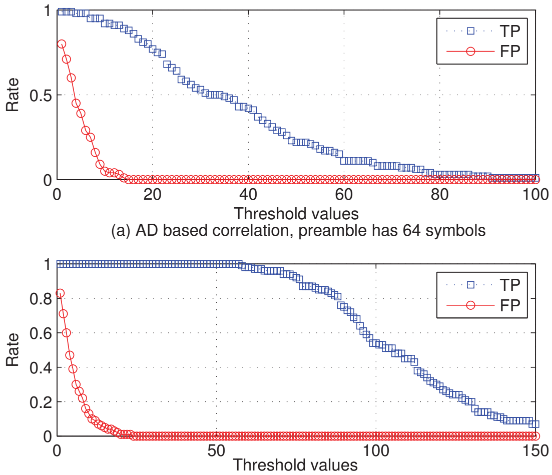
In the evaluation, we will first exam the accuracy of the proposed technique, and then measure its execution time.

B. *Auth-Preamble Verification Accuracy*

In this part of experiments, we let the transmitter send auth-preamble signals, and the friendly jammer tries to verify the received auth-preamble signals using amplitude differential based correlation.

We repeat the experiment for 100 times. In each time, the transmitter transmits legitimate auth-preamble signals and the bogus auth-preamble signals (generated using a wrong key) with the modulated packet payload signals. The friendly jammer monitors the channel and computes the amplitude differential values for both the received auth-preamble signal samples and m interpolated locally generated auth-preamble signal samples. Considering the clock drift, we set the correlation window length for the locally generated auth-preamble signals as $m = 10^4$. Note that these signals can be generated beforehand to reduce the reaction time. The correlation can be transformed to bit-wise operations and executed efficiently. Assume that the computed amplitude differential values are denoted as AD_y and AD_g, respectively. If the difference of the first and the second largest correlation outputs is greater than a given threshold, the received auth-preamble signals are identified as legitimate auth-preamble signals.

In the experiments, the received auth-preamble lengths are 64 symbols (256 samples) and 128 symbols (512 samples). We evaluate the proposed techniques using the true positive rate (i.e., the rate that legitimate preambles are correctly identified) and false positive rate (i.e., the rate that bogus preambles are incorrectly identified as legitimate preambles). When using different thresholds, results for amplitude differential based correction and for the efficient variation (in Section IV-C2) are shown in Fig. 4 and Fig. 5, respectively.



(a) AD based correlation, preamble has 64 symbols

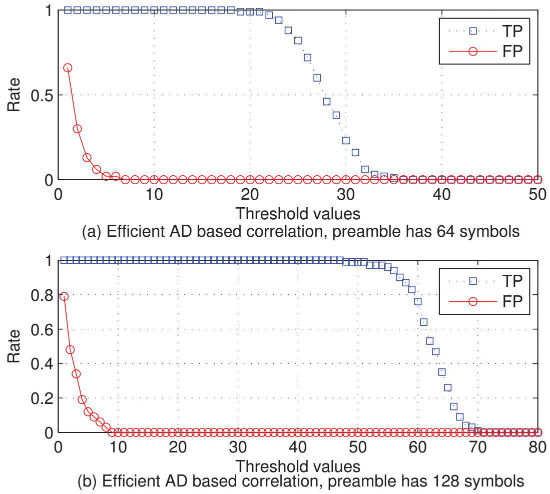
(b) AD based correlation, preamble has 128 symbols

Fig. 4. True positive and false positive of amplitude differential based correlation. TP is true positive and FT is false positive.

We can see that for both amplitude differential based correction and its efficient variation, when the auth-preamble has 128 symbols, there is a range of threshold values which achieve 100% true positive rate with 0% false positive rate. This means the amplitude differential based correlation can distinguish authorized and unauthorized transmissions accurately.

C. *Execution Time*

In this part of experiments, we want to compare the running time of efficient amplitude differential based correlation with the traditional demodulation approach. As the bit-wise operations are much faster compared to the complex float



(a) Efficient AD based correlation, preamble has 64 symbols

(b) Efficient AD based correlation, preamble has 128 symbols

Fig. 5. True positive and false positive of efficient amplitude differential based correlation.

number operations, the dominating time-consuming factor for traditional demodulation approach is the demodulation operations; while for the proposed efficient amplitude differential based correlation approach, it is the computation of amplitude differential values.

To measure the demodulation time, we modify the benchmark receiver in GNURadio by connecting the receive path to demodulation related blocks (e.g., channel filter and demodulator) only and connecting the output directly to a null sink. Similarly, for counting amplitude differential value computation time, we connect the receive path to the amplitude differential values computation blocks and direct the output to a null sink. We measure the time of demodulating certain number of input signals and computing amplitude differential values for the same number of input signal samples.

When the number of input signals is small, the block setup time may dominate the real signal processing time. To make the results more accurate, we set the input signal length from $2 \cdot 10^6$ to 10^7 and run each test for 100 times. We remove the greatest and the smallest ten execution times, the average execution time of remaining tests is shown in Fig. 6 .

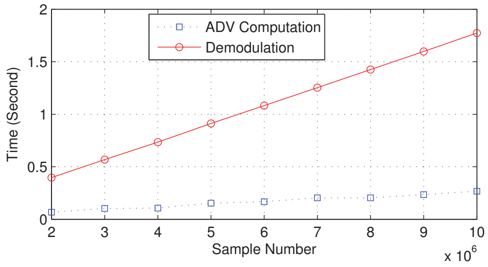


Fig. 6. Time comparison. ADV is the amplitude differential values.

It is easy to see that the computation of the amplitude differential values is in general 6-7 times faster than the demodulation operation. In other words, using efficient amplitude differential based approach rather than the demodulation approach, the friendly jammer can reduce reaction delay by $81.9\% - 85.7\%$.

VII. Conclusion

In this paper, we proposed fast friendly jamming, a novel design to allow the friendly jammer to distinguish the authorized and unauthorized wireless transmissions through verifying auth-preamble signals on the physical layer. We have implemented a prototype of the proposed techniques and performed real-world experiments to evaluate the performance. The experiment results show that the proposed techniques can reduce the reaction delay of the friendly jammer by $81.9\% - 85.7\%$ as compared to the traditional demodulation methods, and enable the accurate distinction between authorized and unauthorized transmissions. In our future work, we will generalize fast friendly jamming to enhance the friendly jamming capability to multi-tap channel scenario.

Acknowledgment

The authors would like to thank the anonymous reviewers for their insightful comments. This work was supported in part by the US National Science Foundation under Grants ECCS-1002258, CNS-1016260, ECCS-1307949, EARS-1444009 and the Army Research Office under grant W911NF-14-1-0324.

References

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

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

[3] 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.

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

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

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

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

[8] 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.

[9] M. Rieback, B. Crispo, and A. Tanenbaum. Keep on blockinin the free world: Personal access control for low-cost rfid tags. In *Security Protocols*. Springer, 2007.

[10] W. Shen, P. Ning, X. He, and H. Dai. Ally friendly jamming: How to jam your enemy and maintain your own wireless connectivity at the same time. In *IEEE Symposium on Security and Privacy*, 2013.

[11] W. Shen, P. Ning, X. He, H. Dai, and Y. Liu. MCR decoding: A MIMO approach for defending against wireless jamming attacks. In *IEEE CNS workshop on Physical-layer Methods for Wireless Security*, 2014.

[12] N. O. Tippenhauer, L. Malisa, A. Ranganathan, and S. Capkun. On limitations of friendly jamming for confidentiality. In *IEEE Symposium on Security and Privacy*, 2013.

[13] M. Wilhelm, I. Martinovic, J. B Schmitt, and V. Lenders. Short paper: reactive jamming in wireless networks: how realistic is the threat? In *WiSec*, 2011.

[14] M. Wilhelm, I. Martinovic, J. B Schmitt, and V. Lenders. Wifire: A firewall for wireless networks. In *ACM SIGCOMM Computer Communication Review*, 2011.

[15] 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.