Mimicry Attacks Against Wireless Link Signature and New Defense Using Time-Synched Link Signature

Song Fang, Yao Liu, and Peng Ning

***Abstract*— Wireless link signature is a physical layer authentication mechanism, using the multipath effect between a transmitter and a receiver to provide authentication of wireless signals. This paper identifies a new attack, called mimicry attack, against the existing wireless link signature schemes. An attacker can forge a legitimate transmitter's link signature as long as it knows the legitimate signal at the receiver's location, and the attacker does not have to be at exactly the same location as the legitimate transmitter. We also extend the mimicry attack to multiple-input multiple-output (MIMO) systems, and conclude that the mimicry attack is feasible only when the number of attacker' antennas is equal to or larger than that of the receiver's antennas. To defend against the mimicry attack, this paper proposes a novel construction for wireless link signature, called time-synched link signature, by integrating cryptographic protection and time factor into wireless physical layer features. Experimental results confirm that the mimicry attack is a real threat and the newly proposed time-synched link signatures are effective in physical layer authentication.**

***Index Terms*— Link signature, MIMO, time-synched.**

I. Introduction

WIRELESS physical layer security is becoming increasingly important as wireless devices are more and more pervasive and adopted in critical applications. There have been multiple proposals in recent years to provide enhanced wireless security using physical layer characteristics, including fingerprinting wireless devices (e.g., [1]–[4]), authenticating and identifying wireless channels (e.g., [5], [6]), and deriving secret keys from wireless channel features only observable to the communicating parties (e.g., [7], [8]).

Among the recent advances in wireless physical layer security is (wireless) link signature. Link signature uses the unique wireless channel characteristics (e.g., the multipath effect) between a transmitter and a receiver to provide authentication of the wireless channel. Three link signature schemes [5], [6], [9] have been proposed so far. Since its initial introduction, link signature has been recognized as a physical layer authentication mechanism for applications where wireless channel characteristics is unique for individual nodes (e.g., [2], [7], [10]–[12]). In this paper, we identify the *mimicry attack* against these link signature schemes.

We start our investigation with the link signature scheme in [5]. It is assumed in [5] that an attacker "cannot 'spoof' an arbitrary link signature" and that the attacker "will not have the same link signature at the receiver unless it is at exactly the same location as the legitimate transmitter". However, we show in this paper that an attacker *can* forge an *arbitrary* link signature as long as it knows the legitimate signal at the receiver's location, and the attacker does not have to be at exactly the same location as the legitimate transmitter in order to forge its link signature.

We also extend the mimicry attack to the link signature scheme in [9]. Since the last link signature scheme in [6] is essentially an integration of the techniques in [5] and [6], all existing link signature schemes are vulnerable to the mimicry attack. Furthermore, we find that if the receiver has two antennas to cooperatively authenticate the transmitter, the attacker with only one antenna cannot successfully launch the mimicry attack. However, we discover that the mimic attack is still feasible if the attacker also has two antennas.

Then we explore the feasibility of the mimicry attack into MIMO systems. If the number of the receiver's receive antennas is larger than that of the attacker's transmit antennas, the receiver can detect the mimicry attack, otherwise, the receiver can be fooled that the attacker's link signatures are the same with the ones of the authenticated transmitter's.

The mimicry attack can apply to the following example scenarios when link signatures are used for authentication: (1) launching location spoofing attacks: an attacker can utilize a fake location to fool a target receiver by creating a fake wireless link signaturer; (2) bypassing motion detection systems: an attacker could maintain its wireless signature unchanged while it is actually moving, thus from the perspective of the target receiver, who utilizes the wireless link signature to determine whether the transmitter moves or not, the attacker appears to remain stationary; (3) bypassing wireless transmitter authentication systems: an attacker can impersonate a legitimate transmitter by forging its wireless link signature.

To provide physical layer authentication capability and defend against the threats identified in this paper, we develop

Manuscript received September 18, 2015; revised December 23, 2015; accepted March 2, 2016. Date of publication March 11, 2016; date of current version April 12, 2016. This work was supported in part by the National Science Foundation under Grant 1527144 and Grant 1553304, and in part by the Army Research Office under Grant W911NF-14-1-0324. An earlier version of this paper was presented at the 2012 IEEE International Conference on Computer Communications and the 2011 Conference on Computer and Communications Security Poster. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Lifeng Lai.

S. Fang and Y. Liu are with the Department of Computer Science and Engineering, University of South Florida, Tampa, FL 33620 USA (e-mail: songf@mail.usf.edu; yliu@cse.usf.edu).

P. Ning is with Samsung Research American, Mountain View, CA 94043 USA (e-mail: peng.ning@samsung.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIFS.2016.2541307

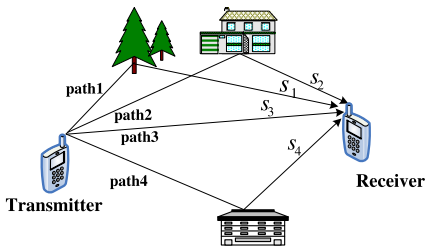


Fig. 1. Multipath example: The transmitted signal propagates over four paths, and the receiver receives corresponding signal copies s_1, s_2, s_3, and s_4.

a novel construction for link signature, which is called time-synched (i.e., time synchronized) link signature. Time-synched link signature integrates cryptographic protection as well as time factor into the wireless physical layer features, and provides an effective and practical solution for authenticating physical layer wireless signals. We also perform an extensive set of experimental evaluation of the mimicry attacks and the time-synched link signature scheme on the USRP2 platform [13] running GNURadio [14]. Our experiments show that the mimicry attack can deteriorate the success rate of distinguishing between the legitimate transmitter and the attacker to 0.5935, which is close to a blind guess. However, with an optimum threshold, the proposed time-synched link signature is able to restore the success rate to 0.9365.

Our contribution in this paper is three-fold. First, we identify the mimicry attack against existing link signature schemes and extend the mimicry attack to MIMO systems. Second, we develop the time-synched link signature scheme to defend against various threats against existing link signature schemes, including the mimicry attacks presented in this paper. Finally, we perform extensive experiments to confirm the threats of the mimicry attack and demonstrate the effectiveness of the time-synched link signature for physical layer authentication.

The rest of the paper is organized as follows. Section II first gives some background information for link signatures. Sections III introducea the mimicry attacks and Sections IV explores the feasibility of the mimicry attacks in MIMO systems. In Section V, we present our proposed time-synched link signature. Next, Section VI gives our experimental confirmation of the mimicry attack as well as evaluation of the time-synched link signature, and Section VII discusses related work. Finally, Section VIII concludes this paper.

II. Preliminaries

In this section, we give some preliminary information on link signatures, including multipath effect, channel impulse response, and how these are used for wireless link signatures.

A. Multipath Effect, Channel Impulse Response

Wireless signal usually propagates in the air along multiple paths due to reflection, diffraction, and scattering [5]. For example, as shown in Figure 1, the receiver receives multiple copies of the transmitted signal from different paths, each of which may have a different delay due to the path it traversed on. The received signal is indeed the sum of these time delayed signal copies. Each path imposes a *response* (e.g., distortion and attenuation) on the signal traveling along it [5], and the superposition of all responses between two nodes is referred to as a *channel impulse response* [15].

The multipath effects between different pairs of nodes are usually different, and so are the channel impulse responses [5]. Due to this reason, a channel impulse response between two nodes is also called a *link signature*, and has been proposed to provide robust location distinction and location-based authentication [5], [6], [11]. Specifically, when a transmitter and attackers are in different locations, to determine whether a received signal is from the transmitter, the receiver can estimate the link signature of the received signal and compare it with the known value from the transmitter. The received signal is accepted only if the estimated link signature is similar to the known value.

B. Estimating Channel Impulse Responses

A popular method for estimating channel impulse responses is the *training sequence based estimation* [16]. The transmitter first sends a training sequence (i.e., a sequence of bits) over the wireless channel. The receiver then uses the training sequence and the corresponding received signal samples to estimate channel impulse responses, where the data value of the training sequence can be pre-shared [16] or reconstructed from the received signal through demodulation [5].

Note that at the physical layer channel estimation can be processed in either frequency domain (e.g. [5], [6]) or time domain (e.g., [16]). Because of the linear relationship between the two domains, frequency and time domain based methods are inter-convertible. In the following, we describe the channel estimation method in the time domain.

1) Mathematical Formulation: To transmit the training sequence, the transmitter converts it into M physical layer symbols (i.e., complex numbers that are transmission units at the physical layer [15]). The transmitter then sends the M symbols to the wireless channel.

Let $\mathbf{x} = [x_1, x_2, \ldots, x_M]$ denote the transmitted symbols in the training sequence. Assume that there exist L paths. Thus, the receiver can receive L copies of $\mathbf{x}$, each traveling on one path and undergoing a response (i.e., distortion and attenuation) caused by the corresponding path. The vector $\mathbf{y}$ of received symbols is the convolution sum of the L copies of $\mathbf{x}$. Let $\mathbf{h} = [h_1, h_2, \ldots, h_L]^T$ be the channel impulse response, where h_i is the response of the i-th path, and $\mathbf{n}$ denote the channel noise. Thus, the received symbols $\mathbf{y}$ can be represented by $\mathbf{y} = \mathbf{h} * \mathbf{x} + \mathbf{n}$ [16], where $*$ is the convolution operator. With the matrix form, we have

$$\mathbf{y} = \begin{bmatrix} x_1 & 0 & \cdot & 0 \\ x_2 & x_1 & \cdot & \cdot \\ \cdot & x_2 & \cdot & 0 \\ x_L & \cdot & \cdot & x_1 \\ \cdot & \cdot & \cdot & \cdot \\ x_M & \cdot & \cdot & x_{M-L+1} \\ 0 & x_M & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & \cdot & x_M \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \\ \cdot \\ \cdot \\ h_L \end{bmatrix} + \mathbf{n} \qquad (1)$$

Rewriting Equation (1) in a compact matrix form gives us

$$\mathbf{y} = \mathbf{X}\mathbf{h} + \mathbf{n}, \tag{2}$$

where $\mathbf{X}$ is a $(L + M - 1) \times L$ Toeplitz matrix, containing L delayed versions of the transmitted symbols $\mathbf{x}$, and $\mathbf{y}$ is a vector consisting of $(L + M - 1)$ received symbols.

2) Estimation: Two types of estimators are generally used to estimate $\mathbf{h}$ from Equation (2): least-square (LS) and linear minimum mean squared error (LMMSE) [17]. If the statistical distribution of the channel impulse responses and noise are unknown, the LS estimator is usually used. If the statistical distribution of the channel impulse responses and noise are known, the LMMSE estimator is often used. For the LS estimator, the estimation result is given by $\hat{\mathbf{h}}_{LS} = (\mathbf{X}^H\mathbf{X})^{-1}\mathbf{X}^H\mathbf{y}$, where $\mathbf{X}^H$ is the conjugate transpose of $\mathbf{X}$ and $()^{-1}$ is the matrix inverse operation [18]. For the LMMSE estimator, the estimation result is: $\hat{\mathbf{h}}_{LMMSE} = \mathbf{R}_h(\mathbf{R}_h + \sigma_n^2(\mathbf{X}\mathbf{X}^H)^{-1})^{-1}\hat{\mathbf{h}}_{LS}$, where $\mathbf{R}_h$ is the channel correlation matrix (i.e., the statistical expectation of $\mathbf{h}\mathbf{h}^H$) and σ_n^2 is the variance of the noise [19].

III. Mimicry Attack

In this section, we present the mimicry attack against link signature schemes [5], [6], [9].

A. Overview

Let $\mathbf{y}_t$ and $\mathbf{y}_a$ denote the received symbols that are from the transmitter and the attacker, respectively. The attacker's goal is to make $\mathbf{y}_a$ approximately the same as $\mathbf{y}_t$. When the receiver attempts to extract the link signature from $\mathbf{y}_a$, it will get a link signature that is very similar to the one estimated from $\mathbf{y}_t$. As a result, the attacker can impersonate the transmitter to bypass link signature based authentication.

The attacker needs to meet two requirements to launch a mimicry attack: First, the attacker needs to know the transmitter's symbols (i.e., $\mathbf{y}_t$) at the receiver's location. Second, the attacker needs to manipulate its own symbols to be transmitted such that when they arrive at the receiver they are similar to those from the transmitter (i.e., $\mathbf{y}_a \approx \mathbf{y}_t$).

B. Obtaining Transmitter's Symbols

There are multiple ways for the attacker to obtain the transmitter's symbols at the receiver's location. For example, the attacker may learn $\mathbf{y}_t$ by placing a sensing device in the proximity of the receiver. For the sake of presentation, we call this device the *symbol sensor*. It records the symbols received from the transmitter and reports them to the attacker through any available communication channel. Note that the characteristic of the wireless channel becomes uncorrelated every half a carrier wavelength over distance [20]. Normally, the symbol sensor would be placed within a range of half a carrier wavelength away from the receiver (e.g., for a 2.4 GHz signal, its wavelength equals to 12.5 cm). Thus, the symbols that the symbol sensor receives are roughly the same as those received by the receiver, and can be used as $\mathbf{y}_t$.

The attacker can also use the mathematical model $\mathbf{y}_t = \mathbf{h}_t * \mathbf{x} + \mathbf{n}$ to derive $\mathbf{y}_t$, where $\mathbf{h}_t$ is the link signature between the transmitter and the receiver. Specifically, the symbol sensor can receive symbols from the transmitter, estimate the link signature from these symbols, and report the link signature to the attacker. The attacker can use the reported link signature as an approximation of $\mathbf{h}_t$ to calculate $\mathbf{y}_t$. In this case, the symbol sensor only needs to report the derived link signatures from time to time, and the attacker can calculate $\mathbf{y}_t$ directly by using the estimated link signature $\mathbf{h}_t$ rather than wait for the sensor to report $\mathbf{y}_t$.

C. Manipulating Transmitted Symbols

The symbols $\mathbf{y}_a$ received from the attacker can be represented as $\mathbf{y}_a = \mathbf{h}_a * \mathbf{x}_a + \mathbf{n}_a$, where $\mathbf{x}_a$, $\mathbf{h}_a$, and $\mathbf{n}_a$ are the symbols transmitted by the attacker, the link signature of the attacker, and the channel noise, respectively. To make $\mathbf{y}_a$ equal to $\mathbf{y}_t$, the attacker can treat $\mathbf{x}_a$ as a unknown variable, and solve it from the equation $\mathbf{h}_a * \mathbf{x}_a + \mathbf{n}_a = \mathbf{y}_t$, where link signature $\mathbf{h}_a$ of the attacker can be obtained from the symbol sensor as well. For previous wireless link signature based authentication schemes [1]–[6], the channel impulse response is assumed to be unchanged in a short time or change slowly. Similarly, we assume that the attacker's link signature does not change between obtaining the attacker's link signature and launching the mimicry attack. The solution to this equation enables $\mathbf{y}_a$ to be similar to or the same as the transmitter's symbols $\mathbf{y}_t$. As a result, the link signatures that are estimated from $\mathbf{y}_a$ will also be close to those estimated from $\mathbf{y}_t$.

Let $\mathbf{x}_a = [x_{a1}, x_{a2}, \ldots, x_{aM}]^T$ denote the symbols transmitted by the attacker, and $\mathbf{h}_a = [h_{a1}, h_{a2}, \ldots, h_{aL}]^T$ denote the link signature of the attacker. We have

$$\begin{aligned}
\mathbf{y}_t &= \mathbf{h}_a * \mathbf{x}_a + \mathbf{n}_a = \mathbf{X}_a\mathbf{h}_a + \mathbf{n}_a \\
&= \begin{bmatrix}
h_{a1} & 0 & \cdot & 0 \\
h_{a2} & h_{a1} & \cdot & \cdot \\
\cdot & h_{a2} & \cdot & 0 \\
\cdot & \cdot & \cdot & h_{a1} \\
h_{aL} & \cdot & \cdot & h_{a2} \\
0 & h_{aL} & \cdot & \cdot \\
\cdot & 0 & \cdot & \cdot \\
0 & 0 & \cdot & h_{aL}
\end{bmatrix}
\begin{bmatrix} x_{a1} \\ x_{a2} \\ \cdot \\ \cdot \\ x_{aM} \end{bmatrix} + \mathbf{n}_a \\
&= \mathbf{H}_a\mathbf{x}_a + \mathbf{n}_a.
\end{aligned}$$

where $\mathbf{H}_a$ is the Toeplitz matrix of the attacker's link signature. Therefore, $\mathbf{y}_t = \mathbf{h}_a * \mathbf{x}_a + \mathbf{n}_a \Leftrightarrow \mathbf{y}_t = \mathbf{H}_a\mathbf{x}_a + \mathbf{n}_a$. We can solve $\mathbf{x}_a$ from $\mathbf{y}_t = \mathbf{H}_a\mathbf{x}_a + \mathbf{n}_a$. Since $\mathbf{n}_a$ is unknown, we use the standard least square approach [18] to solve $\mathbf{x}_a$. Specifically, we minimizes $||\mathbf{y}_t - \mathbf{H_a}\hat{\mathbf{x}}_\mathbf{a}||^2$, where $\hat{\mathbf{x}}_a$ is the approximate solution of $\mathbf{x}_a$. The minimization yields

$$\hat{\mathbf{x}}_a = (\mathbf{H}_a^H\mathbf{H}_a)^{-1}\mathbf{H}_a^H\mathbf{y}_t. \tag{3}$$

Elements in $\mathbf{x}_a$ are already physical layer symbols, and thus they can be transmitted directly. The attacker does not need to modulate them again for transmission.

D. Extending Attack to Multiple Tone Probing Based Link Signature

There are two other link signature schemes [6], [9] besides the one we just attacked [5]. The scheme in [9], referred to as

the multiple tone probing based link signature, uses complex gain at different frequencies to build a link signature, and the scheme in [6] is an integration of the techniques in [5] and [9]. In the following, we show that we can extend the mimicry attack to also compromise the multiple tone probing based link signature, thus making all existing link signature schemes vulnerable.

In multiple tone probing, K carrier waves are simultaneously transmitted to the receiver, and the transmitted signal is $s(t) = \sum_{\kappa=1}^{K} e^{j2\pi f_\kappa t}$ [6], [9], where f_κ is the frequency of the κ-th carrier. Each carrier wave undergoes an attenuation at its center frequency [6]. Thus, the received signal is $r(t) = \sum_{\kappa=1}^{K} H_\kappa e^{j2\pi f_\kappa t}$, where H_κ is the complex channel gain that reflects the amount of attenuation on the κ-th carrier wave. The vector $\mathbf{h} = [H_1, H_2, \ldots, H_K]$ of the complex channel gain is used as the link signature [6], [9].

The mimicry attack identified in this paper can also be adapted to attack the multiple tone probing based link signature. Let $\mathbf{h}_a = [H_{a_1}, H_{a_2}, \ldots, H_{a_K}]$ denote the multiple tone link signature between the attacker and the receiver, and $\mathbf{h}_t = [H_{t_1}, H_{t_2}, \ldots, H_{t_K}]$ denote the one between the transmitter and the receiver. With the knowledge of $\mathbf{h}_t$, the attacker can generate a signal in the following form,

$$s_a(t) = \sum_{\kappa=1}^{K} \frac{H_{t_\kappa}}{H_{a_\kappa}} e^{j2\pi f_\kappa t} = \sum_{\kappa=1}^{K} \frac{\|H_{t_\kappa}\|}{\|H_{a_\kappa}\|} e^{j(2\pi f_\kappa t + \theta_{a_\kappa} - \theta_{t_\kappa})},$$

where $\|.\|$ denote the magnitude of a complex number, θ_{a_κ} and θ_{t_κ} are the phases of H_{a_κ} and H_{t_κ}, respectively. After channel attenuation, the corresponding received signal is

$$r_a(t) = \sum_{\kappa=1}^{K} \frac{H_{t_\kappa}}{H_{a_\kappa}} H_{a_\kappa} e^{j2\pi f_\kappa t} = \sum_{\kappa=1}^{K} H_{t_\kappa} e^{j2\pi f_\kappa t},$$

which equals to the signal $r_t(t)$ received from the transmitter. As a result, the multiple tone link signature estimated from $r_a(t)$ is the same as that estimated from $r_t(t)$.

Since the link signature scheme in [6] is essentially an integration of the scheme in [5] and [9], the above result also makes the scheme in [6] vulnerable to mimicry attacks.

IV. Mimicry Attacks Against MIMO

One may wonder whether the mimicry attack still works in MIMO wireless communication systems. To answer this question, we first explore a simple communication scenario, where the receiver has multiple antennas while the attacker just owns one antenna.

A. Mimicry Attacks With One Antenna

As shown in Figure 2, the receiver is equipped with two receive antennas (antennas A and B). The link signatures between the transmitter and the two antennas of the receiver are $\mathbf{h}_1$ and $\mathbf{h}_2$ respectively. There also exists an attacker, who launches the mimicry attack to impersonate the transmitter. Assume that the attacker knows the link signatures $\mathbf{h}_1'$ and $\mathbf{h}_2'$ between himself and the two antennas of the receiver, respectively. The attacker can learn these link signatures before launching the mimicry attack via multiple methods, such as

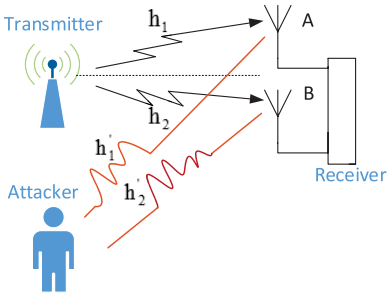


Fig. 2. Mimicry attacks when the receiver has two antennas while the attack has only one antenna.

putting a eavesdropper near the receiver. Hence, if the mimicry attack is successful, for antenna A, we can get

$$\mathbf{X}_a \mathbf{h}_1' = \mathbf{X}\mathbf{h}_1, \tag{4}$$

where $\mathbf{X}_a$ is the Teoplitz matrix of the transmitted sequence $\mathbf{x}_a$. We omit the noise to simplify the equations. Likewise, for antenna B, we have

$$\mathbf{X}_a \mathbf{h}_2' = \mathbf{X}\mathbf{h}_2. \tag{5}$$

In a successful mimicry attack, both Equations (4) and (5) are satisfied. In other words, if we can find a solution of $\mathbf{X}_a$, the receiver will incorrectly think that the attacker is the transmitter. However, Equations (4) and (5) have only one unknown variable $\mathbf{X}_a$, and $\mathbf{h}_1'$ and $\mathbf{h}_2'$ are linearly independent from each other due to the spatial uncorrelation property of wireless channels [15]. Thus, these two equations form an overdetermined linear system. In such a system, it is infeasible for the attacker to find an exact solution of $\mathbf{X}_a$ to make Equations (4) and (5) hold at the same time.

Hence, when the receiver utilizes two antennas to cooperatively authenticate the transmitter, the attacker with only one antenna may fail to launch the mimicry attack. This implies that extra antennas at the receiver can help to mitigate the mimicry attack.

B. Mimicry Attacks With Two Antennas

In this section, we investigate the feasibility of the mimicry attack when the attacker and the receiver both have two antennas, and we discover that the mimicry attack is feasible in such MIMO systems.

As shown in Figure 3, the receiver has antenna A and antenna B, and the attacker has antenna 1 and antenna 2. Thus, there exist 4 pairs of antennas and we denote the corresponding link signatures by $\mathbf{h}_{1a}$, $\mathbf{h}_{1b}$, $\mathbf{h}_{2a}$ and $\mathbf{h}_{2b}$. If the mimicry attack is successful, for antenna A, we can get

$$\mathbf{X}_{a1}\mathbf{h}_{1a} + \mathbf{X}_{a2}\mathbf{h}_{2a} = \mathbf{X}\mathbf{h}_1, \tag{6}$$

where $\mathbf{X}_{a1}$ and $\mathbf{X}_{a2}$ are the Teoplitz matrices of the sequences $\mathbf{x}_{a1}$, $\mathbf{x}_{a2}$ transmitted by antenna 1 and antenna 2, respectively. Likewise, for antenna B, we have

$$\mathbf{X}_{a1}\mathbf{h}_{1b} + \mathbf{X}_{a2}\mathbf{h}_{2b} = \mathbf{X}\mathbf{h}_2. \tag{7}$$

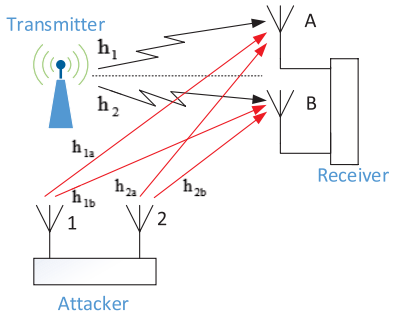


Fig. 3. Mimicry attacks when the receiver has two antennas while the attack also has two antennas.

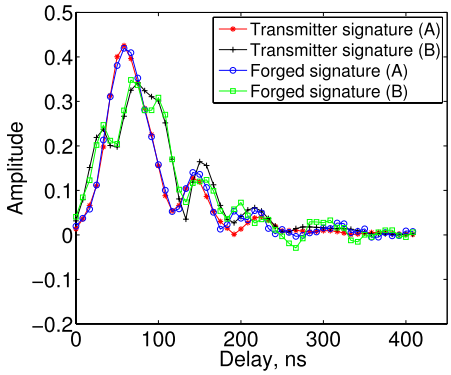


Fig. 4. Mimicry attack in MIMO systems using CRAWDAD data.

Equations (6) and (7) have two unknown variables $\mathbf{X}_{a1}$, $\mathbf{X}_{a2}$. Hence, the system formed by the two equations is not overdetermined and the attacker can find an exact solution to satisfy both equations.

Rewrite Equations (6) and (7), we get $\mathbf{H}_{1a}\mathbf{x}_{a1}+\mathbf{H}_{2a}\mathbf{x}_{a2} = \mathbf{X}\mathbf{h}_1$ and $\mathbf{H}_{1b}\mathbf{x}_{a1} + \mathbf{H}_{2b}\mathbf{x}_{a2} = \mathbf{X}\mathbf{h}_2$, where $\mathbf{H}_{ij}$ ($i \in \{1,2\}$ and $j \in \{a,b\}$) is a $(M+L-1)\times M$ Teoplitz matrix and the transmitted sequences $\mathbf{x}_{a1}$, $\mathbf{x}_{a2}$ are $M \times 1$ vectors. Thus, we can get

$$\begin{bmatrix} \mathbf{H}_{1a} & \mathbf{H}_{2a} \\ \mathbf{H}_{1b} & \mathbf{H}_{2b} \end{bmatrix} \begin{bmatrix} \mathbf{x}_{a1} \\ \mathbf{x}_{a2} \end{bmatrix} = \begin{bmatrix} \mathbf{X}\mathbf{h}_1 \\ \mathbf{X}\mathbf{h}_2 \end{bmatrix} \tag{8}$$

Let $\mathbf{H}$ denote $\begin{bmatrix} \mathbf{H}_{1a} & \mathbf{H}_{2a} \\ \mathbf{H}_{1b} & \mathbf{H}_{2b} \end{bmatrix}$, then we can solve the transmitter sequences $\mathbf{x}_{a1}$ and $\mathbf{x}_{a2}$ using the LS estimator, and the result is

$$\begin{bmatrix} \mathbf{x}_{a1} \\ \mathbf{x}_{a2} \end{bmatrix} = (\mathbf{H}^H\mathbf{H})^{-1}\mathbf{H}^H \begin{bmatrix} \mathbf{X}\mathbf{h}_1 \\ \mathbf{X}\mathbf{h}_2 \end{bmatrix} \tag{9}$$

C. Mimicry Attacks in General Scenarios

We further extend the previous result to a general situation, where the receiver has P antennas to cooperatively authenticate the transmitter and the attacker has Q antennas to launch the mimicry attack. Assume the chosen link signatures that the attacker would like to mimic are represented by $\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_P$, and the attacker's real link signature between the antenna of the attacker and the antenna of the receiver is denoted as $\mathbf{h}_{ij}$, where $i \in \{1,2,\ldots,P\}$ and $j \in \{1,2,\ldots,Q\}$. Thus, we have

$$\begin{cases} \mathbf{H}_{11}\mathbf{x}_{a1} + \mathbf{H}_{21}\mathbf{x}_{a2} + \cdots + \mathbf{H}_{Q1}\mathbf{x}_{aQ} = \mathbf{X}\mathbf{h}_1 \\ \mathbf{H}_{12}\mathbf{x}_{a1} + \mathbf{H}_{22}\mathbf{x}_{a2} + \cdots + \mathbf{H}_{Q2}\mathbf{x}_{aQ} = \mathbf{X}\mathbf{h}_2 \\ \qquad\qquad\qquad\qquad\qquad\qquad \vdots \\ \mathbf{H}_{1P}\mathbf{x}_{a1} + \mathbf{H}_{2P}\mathbf{x}_{a2} + \cdots + \mathbf{H}_{QP}\mathbf{x}_{aQ} = \mathbf{X}\mathbf{h}_P, \end{cases} \tag{10}$$

where $\mathbf{x}_{a1}$, $\mathbf{x}_{a2}$, $\cdots$, $\mathbf{x}_{aQ}$ are sequences transmitted by the attacker's Q antennas. Note that the matrix H_{ji} must be full rank. Thus, we can see that Q should be equal to or larger than P in order to solve the transmit sequences from (10). Specifically,

- When $Q > P$, (10) is an under-determined linear system, and $\mathbf{x}_{ai}$ has an infinite number of solutions.
- When $Q = P$, (10) has the same number of equations and unknowns, and $\mathbf{x}_{ai}$ has a single unique solution.

This result indicates that the attacker must utilize at least the same number of antennas as the receiver to make the mimicry attack feasible.

D. Simulation Results

To validate the feasibility of the mimicry attack against MIMO systems, we use the CRAWDAD data set [21], which contains over 9,300 link signatures measured in an indoor environment with obstacles and scatters. First, we pick up two nodes as the two antennas of the receiver (i.e., nodes 40 and 38), and two other nodes as the attacker's two transmit antennas (i.e., nodes 24 and 25). Then, we record the link signatures between each pair of transmit and receive antennas. Also, we choose another node (i.e., node 23) from the data set as the transmitter. The attacker aims to fool the receiver by mimicking the transmitter's link signatures (i.e., the one between nodes 23 and 40, and that between nodes 23 and 38).

The attacker computes the corresponding transmitted sequences $\mathbf{x}_{a1}$, $\mathbf{x}_{a2}$ based on Equation (10), and sends them to the receiver. The receiver estimates the corresponding link signatures based on the received symbols and the public training sequence. As shown in Figure 4, both the link signatures estimated by the receiver' antennas A and B are correspondingly close to the transmitter's two real link signatures.

Based on the CRAWDAD set, we randomly pick one link signature for the link between the transmitter and antenna A as the comparison base $\mathbf{h}_1$, and another link signature for the link between the transmitter and antenna B as the comparison base $\mathbf{h}_2$. The Euclidean distance between $\mathbf{h}_1$ and the other link signatures for the link between the transmitter and antenna A ranges between 0.0820 and 0.2345, whereas the Euclidean distance between estimated link signature and the transmitter's link signature at antenna A is 0.0922, which falls in the above range. Also, the Euclidean distance between $\mathbf{h}_2$ and the other link signatures for the link between the transmitter and antenna B ranges between 0.1086 and 0.2705, whereas the Euclidean distance between estimated link signature and the transmitter's link signature at antenna B is 0.1416, which also lies in the above range. Therefore, the attacker can successfully fool the receiver into believing that his link signatures are the same as those of the transmitter. The simulation result verifies that it is still

possible to launch mimicry attacks to forge link signatures in MIMO systems.

V. Time-Synched Link Signature

In this section, we develop a novel time-synched link signature to defend against the mimicry attacks. A key feature of this new mechanism is the integration of cryptographic protection and time factor into wireless link signatures.

A. Assumptions and Threat Model

We assume that there are a *Transmitter* and a *Verifier*, sharing a secret key K that is only known to them. Note that this assumption does not contradict with the goal of wireless link signatures, because the main purpose of link signatures is to provide the authentication of locations and this cannot be achieved by using a shared key. The Transmitter sends packets, or more precisely, physical layer *frames*, to the Verifier, who then verifies if these frames are directly transmitted by the Transmitter. We assume that the attacker can eavesdrop, overhear, and jam wireless communications. Also, the attacker is assumed to be able to transmit with a higher power to generate a capture effect to overwhelm the signal sent by the transmitter. However, we assume that the attacker cannot compromise the Transmitter or the Verifier, and thus does not know their secret.

The attacker's goal is to generate or forward frames to the Verifier and convince it that the frames were transmitted directly by the Transmitter. By doing so the attacker may want to convince the Verifier to derive incorrect physical layer characteristics about the transmission (e.g., wrong Received Signal Strength, leading to incorrect estimate of distance).

Given that a cryptographic authentication mechanism (e.g., digital signature, Message Integrity Code (MIC)) can be added to a message to detect forged messages, the main threat is from the frames that are originally generated by the Transmitter but forwarded by the attacker. We focus on the case when the attacker can jam and replay the Transmitter's frames (i.e., the jam-and-replay attack [22]). In the other cases where the Verifier can receive the original transmission by the Transmitter, a duplication detection mechanism (e.g., sequence number) along with authentication can properly detect the frames forwarded by the attacker.

We assume that the attacker can launch *frame repeater attacks*. That is, the attacker is able to receive a frame transmitted by the Transmitter and then forward the frame to the Verifier. Such frame repeaters are widely available commercially (e.g., various brands of 802.11 repeaters). We also assume that the attacker can launch physical layer *symbol repeater attacks*. That is, the attacker can observe the transmission of each physical layer symbol, which may represent one or multiple bits in the frame, and then forward the symbol to the Verifier directly. Such repeaters can be developed using noise canceling techniques and proper positioning of antennas, as described in [23]. Compared with frame repeater attacks, symbol repeater attacks are much harder to defend against.

Link signatures are specific to wireless communication channels, and usually require a training phase for two nodes to learn the actual value. The attacker may target at either the *training phase* to mislead the Transmitter and the Verifier about their link signature, or the *operational phase* (as described in Section III) when the link signature is used for physical layer authentication. Thus, a secure link signature has to protect both the training and the operational phases.

B. Design Strategy

The fundamental reason for the mimicry attack is that the (sniffing) attacker can establish a set of equations based on two pieces of information: (1) the training sequence and (2) the Transmitter's signal (i.e., physical layer symbols) at the Verifier's location. These allow the attacker to manipulate the transmitted physical layer symbols so that a frame sent by the attacker has a valid link signature. To defend against this attack, our strategy is to deprive the attacker at least one of these two pieces of information. It is in general very difficult to prevent a passive attacker from receiving signals (and then extracting valid link signatures). However, it is possible to prevent the attacker from knowing the training sequences. Thus, our initial idea is to use *unpredictable*, *dynamic*, and *authenticated* training sequences for extracting link signatures from wireless packets (frames).

1) Detecting Frames Forwarded by Attackers: It is not hard to realize that simply using unpredictable, dynamic, and authenticated training sequences is still insufficient. The attacker can receive and analyze the Transmitter's signal to learn the training sequence. If the Verifier cannot receive the original transmission (e.g., due to jam-and-replay attack), the attacker can still forge link signatures by manipulating and forwarding a frame received from the Transmitter.

To handle this threat, we propose to bring "time" into the scheme. We assume the Transmitter and the Verifier have synchronized clocks. (As we will show in the proposed scheme, in the training phase the Transmitter and the Verifier will synchronize their clocks to meet this assumption.) The Transmitter may include a timestamp in the transmitted frame, which indicates the time when a particular bit or byte (e.g., the Start of Frame Delimiter (SFD) field in an IEEE 802.11 or 802.15.4 frame [24]) is transmitted over the air. We assume that the Transmitter can use authenticated timestamping techniques (e.g., [25]) to ensure that the timestamp precisely represents the point in time when the SFD field is transmitted in air. As a result, upon receiving a frame, the Verifier can use the timestamp included in the frame and the time when it receives the frame, which should also be obtained through Medium Access Control (MAC) layer timestamping [25], to estimate the traveling time of the frame. An overly long time indicates that the frame has been forwarded by an intermediate attacker. Also, an attacker may revise the timestamp to conceive the Verifier that the calculated delay is small. To avoid such attacks, the Transmitter will send the MIC of the entire frame along with the frame to the Verifier. Thus any revision of the timestamp by an attacker will be detected by the Verifier.

Using MAC layer timestamping can defend against the frame repeater attack fairly well. For example, in an 802.11g wireless network, which supports 54 Mbps bandwidth, the transmission of a 100-byte frame takes about $14.8\mu s$. To maximize the chance to detect retransmitted frames, we may force certain critical frames (e.g., those used to extract physical layer properties such as Received Signal Strength (RSS)) to have a large frame size. In the case of 802.11g, the maximum frame size is 2,346 octets (bytes), which will take about $347.6\mu s$ to transmit. A frame repeater will have to double the transmission time, giving the Verifier a good chance to detect the extra delay and thus detect the attack.

2) Defending Against Physical Layer Symbol Repeater Attacks: A physical layer symbol repeater attack is much harder to detect than frame repeater attacks. If the attacker knows where the training sequence is located in the frame, she can start repeating the physical layer symbols right after she finishes receiving all symbols corresponding to the training sequence. This reduces the delay that the physical layer symbol repeater has to tolerate to the transmission time of only the training sequence, which could be much shorter than the transmission time of the entire frame.

Note that the calculation process of the symbol repeater attack is the same as that of the frame forwarding attack mentioned earlier. For the symbol repeater attack, the attacker starts forwarding manipulated symbols right after she recognizes the training sequence. For the frame forwarding attack, the attacker starts forwarding manipulated symbols once she receives a entire frame. In both attacks, the attacker has to know the training sequence and the processes of calculating the symbols to be transmitted by the attacker are exactly the same, as described in Section III-C.

To defend against such physical layer symbol repeater attacks, we propose to integrate a third idea into the scheme, that is, to make the location of the training sequence *unpredictable until the end of the frame transmission*. Specifically, we propose to insert the training sequence at a *randomly* selected location in the payload, and place this location, which can be represented as the offset from the start of the frame header, at the end of the frame. In order for a physical layer symbol repeater to mimic the link signature of the Transmitter, she has to manipulate the physical layer symbols corresponding to the training sequence in a frame. If the location of the training sequence is not revealed until the end of the frame, the attacker will have to wait until the end of the transmission to learn it. This forces a physical layer symbol repeater attack to degenerate into a frame repeater attack, which can be handled as discussed earlier.

3) Minimum Frame Length: If a frame payload is too short, the Verifier may have difficulty seeing the extra delay caused by a frame repeater. One solution is to pad extra bits into the frame payload if the frame length is less than a minimum frame length. The minimum frame length can be determined based on the errors of the time synchronization and time measurement. Assume that the maximum errors in clock discrepancy and transmission time are e_δ and e_τ, respectively. Further assume that the maximum time measurement errors in the Transmitter and the Verifier are e_T and e_V, respectively. Thus, the maximum error that the Verifier has to tolerate is $e_{all} = e_\delta + e_\tau + e_T + e_V$. Assume that the data rate of the wireless communication is R. It is easy to see that when the frame length is greater than the minimum frame length $L_{min} = R \cdot e_{all}$, the Verifier is guaranteed to detect frames forwarded by frame repeaters.

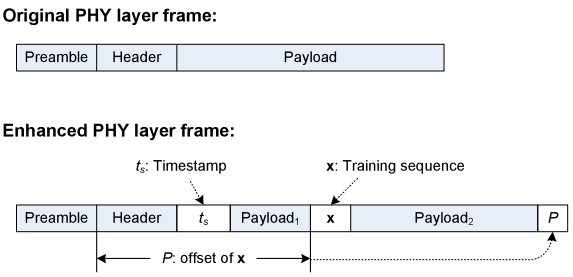


Fig. 5. PHY layer frame format.

It has been demonstrated in an implementation of Radio Frequency (RF) distance bounding protocol [26] that nanosecond processing delay is feasible to achieve. The time-synched link signature requires much less precision in time synchronization between the Transmitter and the Verifier. For example, even assuming e_{all} is between $1\mu s$ and $10\mu s$, in a 54 Mbps 802.11g wireless network, L_{min} will range between 7 bytes and 68 bytes.

4) Overall Design: Figure 5 illustrates how these ideas can be integrated into a physical layer protocol. The upper portion of Figure 5 shows the layout of a typical physical layer frame, which consists of a series of preamble symbols, the frame header, and the payload. To detect frames forwarded by attackers, we include in each frame a timestamp t_s, which indicates the transmission time of the frame. To defend against physical layer repeater attacks, we include the randomly generated offset P of the training sequence in each frame at the end of the frame (to force the attacker to wait until the end of frame transmission).

Assume the Transmitter and the Verifier share a secret key K. Given the shared secret key, there are may ways to generate an unpredictable, dynamic, and authenticated training sequence. One simple method is to piggyback the authentication of the frame with the generation of the training sequence, that is, to use the MIC of the entire frame as the training sequence $\mathbf{x}$. In situations where there is a mismatch between the MIC and the training sequence (e.g., when a longer training sequence is needed), we can simply generate the training sequence as $\mathbf{x} = F(K, t_s)$, where F is a pseudo-random function, and compute the frame MIC separately. The use of K and t_s makes $\mathbf{x}$ dynamic and unpredictable, and the frame MIC allows $\mathbf{x}$ to be authenticated.

In the following, we present the detailed procedure of time-synched link signature, including the training phase and the operational phase.

C. Training Phase

The training phase is intended for the Verifier to collect enough information from the Transmitter so that the Verifier

can verify the link signatures of the future frames from the Transmitter. The Verifier should obtain the valid link signature from the Transmitter whenever the link signature between them may change. This can be accomplished by executing the training phase protocol periodically or whenever one of them moves.

In the training phase, the Verifier needs to synchronize its clock with the Transmitter, and obtain the link signature for the current communication channel. At the same time, the Verifier needs to confirm that there is no successful attack during the training phase.

We use the classic time synchronization technique to estimate the clock discrepancy between the Transmitter and the Verifier as well as the frame traverse time. This approach has been used in the past for secure time synchronization (e.g., [22], [25]). For the sake of presentation, we refer to the point in time when the SFD field of a frame is transmitted or received as the *transmission time* or the *receiving time* of this frame. Specifically, the Verifier sends a *request frame* to the Transmitter, and at the same time records the frame transmission time t_1 in the Verifier's local clock. When the Transmitter receives the request frame, it records the receiving time t_2 of this frame, and then sends a *reply frame* to the Verifier, in which t_2 and the transmission time t_3 of the reply frame, which are both measured in the Transmitter's clock, are included. Finally, the Verifier receives the reply frame and records the receiving time t_4 in its clock. The clock discrepancy δ between the Verifier and the Transmitter and the one-way frame traverse time τ can then be estimated as follows (e.g., [22], [25], [27]):

$$\delta = \frac{(t_2 - t_1) - (t_4 - t3)}{2}; \quad \tau = \frac{(t_2 - t_1) + (t_4 - t_3)}{2}. \quad (11)$$

The Transmitter and the Verifier face a subtle difficulty in time synchronization due to the need of authentication: The timestamps t_1 and t_3 should be the actual transmission time of the request and reply frames; however, the MIC computation requires the timestamp value before the actual transmission. Fortunately, a solution has been previously developed for this problem [25]. It is observed that in the physical layer protocol component, all computation is deterministic if the wireless channel is available for transmission. Thus, we can estimate how much time the deterministic processing will take before (the SFD field of) the frame is transmitted and thus determine the transmission time before computing the frame MIC. If the frame transmission does not happen due to channel unavailability, the estimation, the computation of the MIC, and the transmission can be repeated.

To defend against potential frame repeater and physical layer symbol repeater attacks, we use the design given in Section V-B. That is, the Transmitter pads the reply frame payload so that after all necessary components of the frame are included, the frame length is at least the minimum frame length L_{min}. The Transmitter uses the MIC of the entire frame as the link signature training sequence, and places it at a random offset in the frame payload. Finally, the Transmitter places the random offset at the end of the frame.

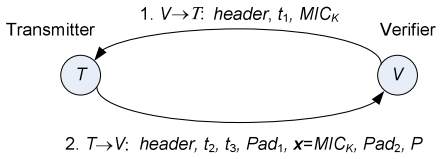


Fig. 6. Training phase protocol.

Figure 6 shows the training phase protocol between the Transmitter and the Verifier.

1) Training Request: The Verifier sends the first training request frame to the Transmitter, which includes the frame header, the transmission time t_1 of this frame, and the frame MIC that covers the entire frame (excluding the preambles). Upon receiving of the request frame, the Transmitter immediately records the receiving time t_2 of the frame, and authenticates the request frame by verifying the MIC.

We can also filter some bogus requests before verifying the MIC. Though a clock discrepancy between the Transmitter and the Verifier is expected, there is usually a maximum clock discrepancy δ_{max}. If $(t_2 + \delta_{max} - t_1)$ is too large, it is likely that the request frame is a replay of a previous request frame, and should be discarded without verification.

2) Training Reply: Upon verifying an incoming training request frame, the Transmitter should send back a training reply frame. The Transmitter should include time t_2 and the actual transmission time t_3 of the reply frame in the frame. The Transmitter also pads the frame payload to at least the minimum frame length L_{min} and randomly selects an offset P to place the training sequence as discussed earlier. The Transmitter then leaves a placeholder (e.g., all 0's) in place of the training sequence and computes the frame MIC using the shared key K. Finally, the Transmitter places the frame MIC as the training sequence $\mathbf{x}$ in the reply frame and sends it over the air.

Once the Verifier receives the training reply frame, the Verifier first computes the clock discrepancy δ and the one-way transmission time τ according to Equation (11). If τ is greater than a threshold τ_{max}, which is the maximum possible direct transmission time, the Verifier should consider the reply frame as possibly forwarded by the attacker and discard it. Otherwise, the Verifier locates the frame MIC by following the offset P at the end of the frame, authenticates the frame MIC using the shared key K, and uses the frame MIC, which is also the training sequence $\mathbf{x}$, to extract the link signature. The Verifier may run the training phase protocol several times to get a better quality link signature. As a result, the Verifier obtains the valid link signature to perform physical layer authentication of future frames from the Transmitter.

D. Operational Phase

Once the Verifier obtains the clock discrepancy and the valid link signature from the Transmitter, the two nodes can go into the operational phase, during which the Verifier can use this link signature to verify frames that require physical layer authentication.

1) Transmitter: To defend against the threats discussed in Section V-A, the Transmitter follows the design shown in Figure 5. Specifically, the Transmitter randomly selects an offset in the frame payload to include the field for the training sequence.[1] The Transmitter also includes the transmission time t_s, places the offset P at the end of the frame, and computes the frame MIC using the shared secrete key K, with a placeholder (e.g., all 0's) for the training sequence. The Transmitter then uses the frame MIC as the training sequence $\mathbf{x}$, puts it in the frame, and sends the frame over the air. Similar to the training phase, the Transmitter estimates the frame transmission t_s based on the current time and the estimated duration for the deterministic MIC computation.

2) Verifier: When the Verifier receives the frame, it immediately records the receiving time t_r. The Verifier then retrieves the frame transmission time t_s from the received frame and estimates the frame traverse time $\tau = t_s - t_r - \delta$, where δ is the clock discrepancy between the Verifier and the Transmitter learned in the training phase. If τ is greater than the threshold τ_{max}, the maximum possible direct transmission time, the Verifier should consider the frame possibly forwarded by the attacker and discard it. Otherwise, the Verifier locates the frame MIC by using the offset P at the end of the frame, verifies the frame MIC using the shared key K, and then uses the frame MIC as the training sequence to extract the link signature. Finally, the Verifier compares this link signature with the one derived during the training phase. The frame is accepted if this link signature does not deviate from the valid one learned in the training phase. Otherwise, the frame is considered forged and discarded.

E. Security Analysis

Now let us examine the ability of the time-synched link signature to defend against the malicious threats.

First of all, the time-synched link signature uses a training sequence authenticated with a shared secret key only known to the Transmitter and the Verifier, and the training sequence changes from frame to frame due to the involvement of the timestamp in the computation of the training sequence. Thus, the training sequence is authenticated, dynamic, and unpredictable. This effectively prevents the attacker from forging frames with training sequences of its choice. The only choice left for the attacker is to reuse and manipulate valid frames from the Transmitter.

The use of random offset for the training sequence in the frame payload forces the attacker to wait for the end of the frame transmission to understand where the training sequence is located in the frame. As a result, the attacker cannot launch physical layer symbol repeater attacks and at the same time manipulate the training sequence correctly to bypass link signature verification. The attacker may still perform the frame repeater attack. However, due to the enforcement of the minimum frame length, a frame forwarded by a frame repeater will introduce at least the amount of delay caused by the receiving of the frame, which is detectable by the Verifier.

The attacker may launch a probabilistic mimicry attack by randomly guessing the location of the training sequence and forging the frame symbols accordingly. Indeed, the attacker may also try to overestimate the length of the training sequence and perform the forgery. If the assumed training sequence $\mathbf{y}'_t$ is a superset of the actual one $\mathbf{y}_t$ (i.e., $\mathbf{y}_t$ is a subsequence of $\mathbf{y}'_t$), due to the linear property of Equation (3), the forged symbols $\hat{\mathbf{x}}'_a$ will also include $\hat{\mathbf{x}}_a$ as a subsequence. This will allow the attacker's symbols to be accepted by the receiver. However, the attacker cannot delay the transmission of a frame for L_{min} or more; otherwise, its interference will be detected. This means that the probability for the attacker to succeed is at most $p = \frac{L_{min}-|x|+1}{F-|x|+1}$ when L_{min} is greater than or equal $|x|$, where $|x|$ and F are the length of the training sequence and the frame payload, respectively. When L_{min} is less than $|x|$, the probability of a successful mimicry attack degrades to 0. For example, in a 54Mbps 802.11g wireless network, if we can achieve $2.96\mu s$ precision in the time synchronization and measurement error (i.e., $e_{all} = 2.96\mu s$ and $L_{min} = 159.84$ bits) and use HMAC-SHA1 to generate the training sequence (i.e., $|x| = 160$ bits), the probabilistic mimicry attack is guaranteed to fail.

Nevertheless, the probabilistic mimicry attack does increase the requirement for time synchronization. In other words, the Transmitter and the Verifier need to obtain fine-grained time synchronization so that the success probability of a probabilistic mimicry attack becomes negligible.

VI. Experimental Evaluation

We have implemented the link signature scheme in [5], the mimicry attack, and the newly proposed time-synched link signature. We have also implemented the frame repeater attack, which can be used along with the mimicry attack. Our prototype uses USRP2 [13], which are equipped with AD and DA converters as the RF front ends, and XCVR2400 daughter boards operating in the 2.4 GHZ range as transceivers. The software implementation is based on GNURadio [14].

USRP2s are capable of processing signals up to 100MHz wide. Such a high bandwidth enables the use of them for capturing multipath effects and measuring link signatures. However, GNURadio configuration requires to set the values of interpolation (decimation) rate at the transmitter (receiver) and the number of samples per symbol. If the values of those parameters are set too high, the actual bandwidth will be significantly reduced. To guarantee the capture of multipath effect, we set those parameters the minimum values allowed by GNURadio (i.e., 5 for interpolation and decimation rate, and 2 for number of samples per symbol).

A. Evaluation Methodology

1) Evaluation Scenarios: Our prototype system consists of a transmitter, a receiver (i.e., the verifier in case of time-synched link signature), and an attacker. Each node is a USRP2 connected to a commodity PC. The receiver estimates

[1]Note that the *training* sequence is necessary for the Verifier to extract the link signature. It is used in the operational phase even though its name has "training" in it.

the received link signatures and compares them with the transmitter's link signatures.

We evaluate three scenarios: (1) normal scenario, (2) forgery scenario, and (3) defense scenario. In a normal scenario, the attacker simply sends original symbols to the receiver. In both the forgery and the defense scenarios, the receiver functions as the symbol sensor for the attacker. It estimates the link signatures for the attacker and provides this link signature and the received symbols from the transmitter to the attacker. Upon obtaining this information, the attacker launches the mimicry attack. However, the forgery scenario uses the previous link signature scheme in [5], while the defense scenario uses the newly proposed time-synched link signature.

2) Evaluation Metrics: Intuitively, the attacker wants to reduce the difference between its own link signatures and the transmitter's link signatures, whereas the defense method aims to increase this difference to alert the transmitter. Thus, the link difference between both the attacker's and the transmitter's link signatures can visually reveal the impact of mimicry attacks and the effectiveness of the defense method.

The receiver measures N link signatures of the transmitter, where we set N to 50 in our evaluation. Let $\mathcal{H}$ denote the set formed by the N link signatures. We collect 500 link signatures from the attacker, and calculate the link difference $d_{a,\mathcal{H}}$ between $\mathcal{H}$ and them. For the purpose of comparison, we also let the receiver collect 500 link signatures from the transmitter, and calculate the link difference $d_{t,\mathcal{H}}$ between $\mathcal{H}$ and those newly collected link signatures.

According to [5], the above link difference (i.e., $d_{a,\mathcal{H}}$ and $d_{t,\mathcal{H}}$) is calculated using $\frac{1}{\sigma}\min_{\mathbf{g}\in\mathcal{H}}\|\mathbf{g}-\mathbf{h}\|$, where $\mathbf{h}$ is a link signature of the attacker or the transmitter, and σ is the *historical average difference* between link signatures in $\mathcal{H}$ [5], and is given by $\sigma = \frac{1}{N(N-1)}\sum_{\mathbf{g}\in\mathcal{H}}\sum_{\mathbf{q}\in\mathcal{H}-\mathbf{g}}\|\mathbf{q}-\mathbf{g}\|$.

Link signature based authentication serves as a detector that decides whether or not a received signal is from the desired source. Thus, besides link difference, we also use detection rate P_D (i.e., the rate that an attacker's link signature is successfully detected by the receiver) and false alarm rate P_{FA} (i.e., a transmitter's link signature is incorrectly identified as the attacker's link signature) as two additional evaluation metrics. Finally, we measure the time delay introduced by the transmitter and the attacker to assess how well the frame repeaters can be detected.

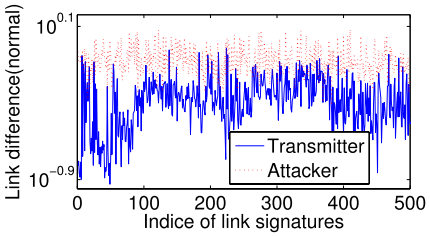


Fig. 7. Normal scenario.

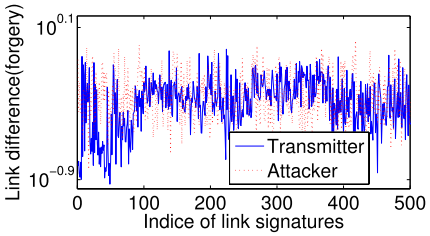


Fig. 8. Forgery scenario.

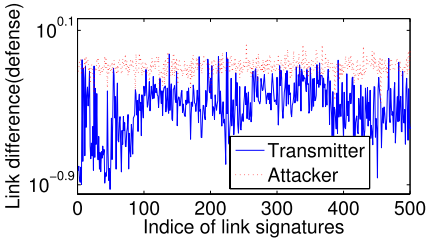


Fig. 9. Defense scenario.

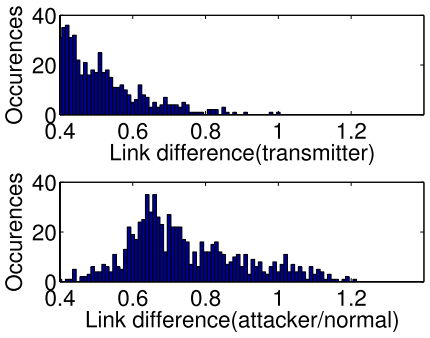


Fig. 10. Link difference for the transmitter and the attacker in normal scenario.

B. Evaluation Results

We now show how mimicry attacks affect the link difference, false alarm rate, detection rate, and the tradeoff between the detection and the false alarm rates in the normal, forgery, and defense scenarios.

1) Link Difference: Figures 7, 8, and 9 show the link difference for the attacker $d_{a,\mathcal{H}}$ and that for the transmitter $d_{t,\mathcal{H}}$ in the normal, forgery, and defense scenarios, respectively. In the normal scenario, we see in Figure 7 that $d_{a,\mathcal{H}}$ is generally larger than $d_{t,\mathcal{H}}$. The histograms $d_{a,\mathcal{H}}$ and $d_{t,\mathcal{H}}$ are shown in Figure 10. Most of the transmitter's link difference is less than 0.6, whereas most of the attacker's link difference is larger than 0.6. Thus, based on the value of link difference, the receiver can achieve a high accuracy in distinguishing between the transmitter and the attacker.

In the forgery scenario, the attacker launches mimicry attacks to make its own link signatures similar to the transmitter's link signatures. We see in Figure 8 that $d_{a,\mathcal{H}}$ decreases to the same level as $d_{t,\mathcal{H}}$, and $d_{a,\mathcal{H}}$ and $d_{t,\mathcal{H}}$ substantially overlap with each other. The histogram of $d_{a,\mathcal{H}}$ (i.e., the top graph in Figure 11) shows that the link difference distribution of the attacker gets very close to that of the transmitter. The mimicry attack reduces the difference between the attacker's link signatures and the transmitter's link signatures, leading to high false negative rate at the receiver.

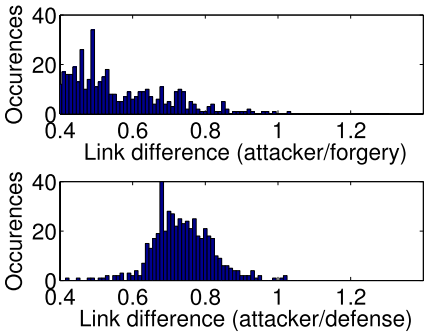


Fig. 11. Link difference for the attacker in forgery and defense scenarios.

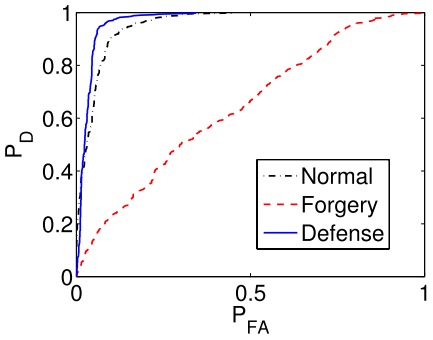


Fig. 13. Tradeoff between false alarm and detection rate.

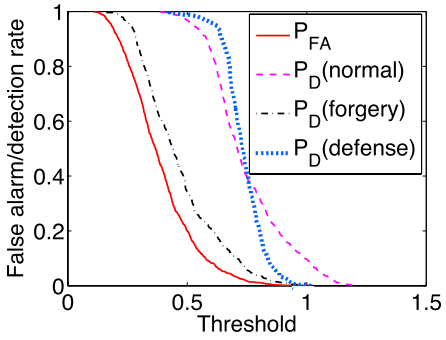


Fig. 12. False alarm rate P_{FA} and detection rate P_D.

In the defense scenario, as indicated in Figure 9, the use of time-synched link signature increases $d_{a,\mathcal{H}}$ of forged link signatures. In particular, the mean value of $d_{a,\mathcal{H}}$ under defense and forgery scenarios are 0.7368 and 0.4648, respectively. The histogram of $d_{a,\mathcal{H}}$ in the defense scenario (i.e., the bottom graph in Figure 11) shows that the link difference computed from a majority of forged signatures is smaller than 0.6. Thus, the receiver can again distinguish between the transmitter and the attacker with low error rate.

2) Detection and False Alarm Rates: As we mentioned earlier, a history of N transmitter's link signatures is measured and stored at the receiver, and the receiver computes the link difference d between a newly measured link signature and history link signatures. In our experiment, we follow the same detection rule as used in [5]. Specifically, if d is smaller than a certain threshold r, the receiver concludes that this link signature is from the transmitter. Otherwise, the receiver assumes that it is from the attacker.

Let N_{FA} denote the number of link signatures that are actually from the transmitter but incorrectly identified as from the attacker, and N_D denote the number of link signatures that are from the attacker and detected by the receiver. The false alarm rate P_{FA} is calculated as the ratio of N_{FA} to the total number of the transmitter's link signatures, and the detection rate P_D is computed as the ratio of N_D to the total number of the attacker's link signatures.

Figure 12 shows P_{FA} and P_D as a function of the threshold r. A large threshold can reduce false alarm rate P_{FA}, whereas a small threshold can increase detection rate. Therefore, an optimum threshold that can both minimize false alarm rate and maximize detection rate is usually desired by the receiver. Such optimum threshold actually occurs at the point where the distance between P_{FA} and P_D is the largest (i.e., $P_D - P_{FA}$ is the largest). The optimum threshold of the normal, defense, and forgery scenarios are 0.5811, 0.6329, and 0.4182, respectively. For the normal scenario, the corresponding P_{FA} and P_D achieved by the optimum threshold are $P_{FA} = 0.0936$ and $P_D = 0.9064$. The defense scenario slightly outperforms the normal scenario in terms of reducing P_{FA} and increasing P_D with the optimum threshold, leading to $P_{FA} = 0.0635$ and $P_D = 0.9365$.

The forgery scenario has the worst performance. With the optimum threshold, $P_{FA} = 0.4045$ and $P_D = 0.5935$. Note that, in our experiment, a link signature is either from the transmitter or from the attacker, and thus the probability that a blind guess hits the true source of this link signature is 0.5. The false alarm rate P_{FA} and detection rate P_D in the forgery scenario are just slightly better than a blind guess.

Figure 13 shows the receiver operating characteristic (ROC) curves for the normal, forgery, and defense scenarios, in which the P_{FA} and P_D are the x-axis and y-axis, respectively. The curve representing the defense scenario is on the top-left corner of the figure, indicating good performance of the time-synched link signature.

3) Frame Time Delay: The proposed time-synched link signature uses estimated frame traverse to filter out frames forwarded by the attacker. We measure the time delay of frames from the transmitter and the attacker, respectively, to examine this approach. In our experiment, the frame length is 190 bits and the transmission rate is set to 500Kbps. The transmitter sends 130 frames, and the attacker forwards all of them. Thus, the receiver receives 260 frames in total.

Figure 14 shows that the time delays of frames forwarded by the attacker significantly exceed those of the frames directly by the transmitter. Our further analysis indicates that the ratio of attacker's delay to the transmitter's delay ranges between 2.2 and 2.6, indicating that the forwarding by the attacker approximately doubles the time delay.

We would like to caution the reader that due to the limitation of USRP2, our implementation does not perform physical layer timestamping. Thus, the time delay measured in our experiments include the processing time on the PC and the USRP2 boards. In a real deployment, physical layer timestamping is necessary to increase the precision of time synchronization and time measurement.

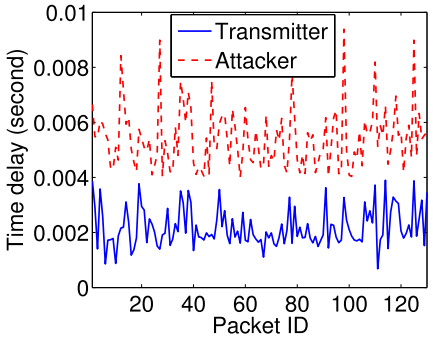


Fig. 14. Time delay of forwarded packets and original packets.

VII. Related Work

A. Wireless Transmitter Authentication

Existing techniques using non-cryptographic approaches to authenticate wireless transmitters can be classified into three categories [2]: software fingerprinting (e.g., [28]–[30]), location distinction (e.g., [5], [6], [9]), and radiometric identification (e.g., [2], [31]).

In software fingerprinting approaches, discrepancies in software configuration are used as fingerprints to distinguish between wireless nodes [2]. For example, Franklin et al. [28] proposed to use the implementation dependent differences among device drivers to identify 802.11 nodes. Kohno et al. [30] proposed to use clock skews in TCP and ICMP timestamps to fingerprint networked devices.

In location distinction based authentication, a signal is authenticated by verifying whether it originates from the expected location of the transmitter. RSS (e.g., [32]) and link signatures have been used to enable such location distinction [5]. The RSS based methods directly estimate the location of a signal origin using the RSS values. However, such methods can be defeated with an array antenna, which can fake arbitrary source locations [5]. The link signature based approaches authenticate the channel characteristics between the transmitter and the receiver [5], [6], [9]. In this paper, we showed that all these link signature scheme are vulnerable to mimicry attacks. Our newly proposed time-synched link signature is developed to fill this gap.

In radiometric identification approaches, the distinctive physical layer characteristics exhibited by wireless devices are utilized to distinguish between them. Transient based techniques (e.g., [31]) identify a wireless device by looking at the unique features "during the transient phase when the radio is turned on" [33]. Modulation based techniques (e.g., [2]) measure differentiating artifacts of individual wireless frames in the modulation domain to identify the device.

B. Attacks on Radiometric Identification

Recently, it was demonstrated in [33] and [34] that radiometric identification techniques were vulnerable to impersonation attacks. The results in [33] revealed that both transient and modulation based techniques are vulnerable to impersonation attacks, though transient-based techniques are harder to reproduce. Edman and Yener [34] showed that an attacker can significantly reduce the accuracy of such techniques by simply using a commodity RF hardware platform. These works are complementary to ours in this paper.

In our previous works [35] and [36], we only addressed the simple mimicry attack scenario, where both the receiver and the attacker have only one antenna. In this paper, we discussed the general case when both the receiver and the attacker have multiple antennas, and discovered that the mimicry attack is still feasible in MIMO systems, as long as the attacker can utilize at least the same number of antennas as the receiver. We also extended mimicry attacks to the multiple tone probing based link signature and showed that mimicry attacks can make all existing link signature schemes vulnerable. Furthermore, in [36], we only compared the link differences for the attacker and the transmitter in the normal, forgery and defense scenarios, respectively. In this paper, we further explored how to set an appropriate threshold that enables the proposed time-synched link signature scheme to achieve a high detection rate while keeping a low false alarm rate in the three scenarios.

VIII. Conclusion

In this paper, we identified the mimicry attack against the existing wireless link signature schemes. We then extended the mimicry attack in MIMO systems and concluded that the attacker utilizing at least the same number of antennas as the receiver's antennas can successfully launch the mimicry attack. To defend against the mimicry attack, we proposed the novel time-synched link signature construction by integrating cryptographic protection and time factor into wireless physical layer features. We also performed an extensive set of experiments to demonstrate both the feasibility of mimicry attacks and the effectiveness of time-synched link signature.

References

[1] D. B. Faria and D. R. Cheriton, "Detecting identity-based attacks in wireless networks using signalprints," in *Proc. ACM Workshop Wireless Secur. (WiSec)*, 2006, pp. 43–52.

[2] V. Brik, S. Banerjee, M. Gruteser, and S. Oh, "Wireless device identification with radiometric signatures," in *Proc. 14th ACM Int. Conf. Mobile Comput. Netw. (MobiCom)*, 2008, pp. 116–127.

[3] R. M. Gerdes, T. E. Daniels, M. Mina, and S. Russell, "Device identification via analog signal fingerprinting: A matched filter approach," in *Proc. 13th Annu. Symp. Netw. Distributed Syst. Secur. (NDSS)*, 2006, pp. 1–11.

[4] L. C. C. Desmond, C. C. Yuan, T. C. Pheng, and R. S. Lee, "Identifying unique devices through wireless fingerprinting," in *Proc. 1st ACM Conf. Wireless Netw. Secur. (WiSec)*, 2008, pp. 46–55.

[5] N. Patwari and S. K. Kasera, "Robust location distinction using temporal link signatures," in *Proc. 13th Annu. ACM Int. Conf. Mobile Comput. Netw. (MobiCom)*, 2007, pp. 111–122.

[6] J. Zhang, M. H. Firooz, N. Patwari, and S. K. Kasera, "Advancing wireless link signatures for location distinction," in *Proc. 14th ACM Int. Conf. Mobile Comput. Netw. (MobiCom)*, 2008, pp. 26–37.

[7] S. Mathur, W. Trappe, N. Mandayam, C. Ye, and A. Reznik, "Radio-telepathy: Extracting a secret key from an unauthenticated wireless channel," in *Proc. 14th ACM Int. Conf. Mobile Comput. Netw. (MobiCom)*, 2008, pp. 128–139.

[8] H. Liu, Y. Wang, J. Yang, and Y. Chen, "Fast and practical secret key extraction by exploiting channel response," in *Proc. IEEE INFOCOM*, Apr. 2013, pp. 3048–3056.

[9] Z. Li, W. Xu, R. Miller, and W. Trappe, "Securing wireless systems via lower layer enforcements," in *Proc. ACM Workshop Wireless Secur. (WiSec)*, 2006, pp. 33–42.

[10] Y. Liu, P. Ning, and H. Dai, "Authenticating primary users' signals in cognitive radio networks via integrated cryptographic and wireless link signatures," in *Proc. IEEE Symp. Secur. Privacy (S&P)*, May 2010, pp. 286–301.

[11] K. Zeng, K. Govindan, and P. Mohapatra, "Non-cryptographic authentication and identification in wireless networks," *IEEE Wireless Commun.*, vol. 17, no. 5, pp. 56–62, Oct. 2010.

[12] A. Kalamandeen, A. Scannell, E. de Lara, A. Sheth, and A. LaMarca, "Ensemble: Cooperative proximity-based authentication," in *Proc. 8th Int. Conf. Mobile Syst., Appl., Services (MobiSys)*, 2010, pp. 331–344.

[13] Ettus Research. *The USRP Product Family Products and Daughter Boards*, accessed on Apr. 2011. [Online]. Available: http://www.ettus.com/products

[14] *GNU Radio—The GNU Software Radio*, accessed on Sep. 2014. [Online]. Available: http://www.gnu.org/software/gnuradio/

[15] A. Goldsmith, *Wireless Communications*. Cambridge, U.K.: Cambridge Univ. Press, 2005.

[16] R. Safaya. *A Multipath Channel Estimation Algorithm Using a Kalman Filter*, accessed on Apr. 2011. [Online]. Available: http://www.ittc.ku.edu/research/thesis/documents/rupul_safaya_thesis.pdf

[17] M. Biguesh and A. B. Gershman, "Training-based MIMO channel estimation: A study of estimator tradeoffs and optimal training signals," *IEEE Trans. Signal Process.*, vol. 54, no. 3, pp. 884–893, Mar. 2006.

[18] K. S. Shanmugan and A. M. Breipohl, *Random Signals: Detection, Estimation and Data Analysis*. New York, NY, USA: Wiley, May 1988.

[19] O. Edfors, M. Sandell, J. J. van de Beek, S. K. Wilson, and P. O. Börjesson, "OFDM channel estimation by singular value decomposition," *IEEE Trans. Commun.*, vol. 46, no. 7, pp. 931–939, Jul. 1998.

[20] X. He, H. Dai, W. Shen, and P. Ning, "Is link signature dependable for wireless security?" in *Proc. IEEE INFOCOM*, Apr. 2013, pp. 200–204.

[21] Sensing and Processing Across Networks at Utah. *Measured Channel Impulse Response Data Set*, accessed on Sep. 2014. [Online]. Available: http://span.ece.utah.edu/pmwiki/pmwiki.php?n=Main.MeasuredCIRDataSet

[22] S. Ganeriwal, S. Čapkun, C.-C. Han, and M. B. Srivastava, "Secure time synchronization service for sensor networks," in *Proc. ACM Workshop Wireless Secur. (WiSec)*, Sep. 2005, pp. 97–106.

[23] J. I. Choi, M. Jain, K. Srinivasan, P. Levis, and S. Katti, "Achieving single channel, full duplex wireless communication," in *Proc. 16th ACM Mobicom (Mobicom)*, Sep. 2010, pp. 1–12.

[24] *IEEE Standard for Information Technology—Telecommunications and Information Exchange Between Systems—Local and Metropolitan Area Networks—Specific Requirements. Part 15.1: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Wireless Personal Area Networks (WPANs)*, IEEE Standard 802.15.4, 2005.

[25] K. Sun, P. Ning, and C. Wang, "TinySeRSync: Secure and resilient time synchronization in wireless sensor networks," in *Proc. 13th ACM Conf. Comput. Commun. Secur. (CCS)*, 2006, pp. 264–277.

[26] K. B. Rasmussen and S. Čapkun, "Realization of RF distance bounding," in *Proc. USENIX Secur. Symp.*, 2010, pp. 389–402.

[27] D. L. Mills, "Internet time synchronization: The network time protocol," *IEEE Trans. Commun.*, vol. 39, no. 10, pp. 1482–1493, Oct. 1991.

[28] J. Franklin, D. McCoy, P. Tabriz, V. Neagoe, J. V. Randwyk, and D. Sicker, "Passive data link layer 802.11 wireless device driver fingerprinting," in *Proc. Usenix Secur. Symp.*, 2006, pp. 1–12.

[29] J. Pang, B. Greenstein, R. Gummadi, S. Seshan, and D. Wetherall, "802.11 user fingerprinting," in *Proc. 13th Annu. ACM Int. Conf. Mobile Comput. Netw. (MobiCom)*, 2007, pp. 99–110.

[30] T. Kohno, A. Broido, and K. Claffy, "Remote physical device fingerprinting," *IEEE Trans. Dependable Secure Comput.*, vol. 2, no. 2, pp. 93–108, Apr./Jun. 2005.

[31] B. Danev and S. Čapkun, "Transient-based identification of wireless sensor nodes," in *Proc. ACM/IEEE Conf. Inf. Process. Sensor Netw. (IPSN)*, Apr. 2009, pp. 25–36.

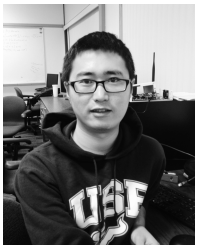
[32] R. Chen, J. M. Park, and J. H. Reed, "Defense against primary user emulation attacks in cognitive radio networks," *IEEE J. Sel. Areas Commun.*, vol. 26, no. 1, pp. 25–37, Jan. 2008.

[33] B. Danev, H. Luecken, S. Čapkun, and K. El Defrawy, "Attacks on physical-layer identification," in *Proc. 3rd ACM Conf. Wireless Netw. Secur. (WiSec)*, Mar. 2010, pp. 89–98.

[34] M. Edman and B. Yener, "Active attacks against modulation-based radiometric identification," Dept. Comput. Sci., Rensselaer Polytechn. Inst., Troy, NY, USA, Tech. Rep. TR 09-02, 2009.

[35] Y. Liu and P. Ning, "Poster: Mimicry attacks against wireless link signature," in *Proc. 16th ACM Conf. Comput. Commun. Secur. (CCS)*, 2011, pp. 801–804.

[36] Y. Liu and P. Ning, "Enhanced wireless channel authentication using time-synched link signature," in *Proc. IEEE INFOCOM*, Mar. 2012, pp. 2636–2640.



Song Fang received the B.S. degree in information engineering from the South China University of Technology, Guangzhou, China, in 2011, and the M.S. degree in communication and information engineering from the Beijing University of Posts and Telecommunications, Beijing, China, in 2014. He is currently pursuing the Ph.D. degree in computer science from the University of South Florida, Tampa, FL. His research interests are in the area of network security and system security.

Yao Liu received the Ph.D. degree in computer science from North Carolina State University, in 2012. She is currently an Assistant Professor with the Department of Computer Science and Engineering, University of South Florida, Tampa, FL. Her research is related to computer and network security, with an emphasis on designing and implementing defense approaches that protect emerging wireless technologies from being undermined by adversaries. Her research interests also lie in the security of cyber-physical systems, especially in smart grid security. She was the recipient of the best paper award for the Seventh IEEE International Conference on Mobile Ad-Hoc and Sensor Systems.

Peng Ning is currently a Professor with the Department of Computer Science, North Carolina State University, Raleigh, NC, USA. He is on leave at Samsung Mobile, Santa Clara, CA, USA, where he is leading the Samsung KNOX Research and Development Team. His research interests are primarily in mobile security, wireless security, and cloud computing security.