Far Proximity Identification in Wireless Systems

Tao Wang†, Jian Weng‡, Jay Ligatti† and Yao Liu†
†University of South Florida, Tampa, FL
‡Jinan University, Guangzhou, China
{taow@mail, yliu@cse, ligatti@cse}.usf.edu, cryptjweng@gmail.com

Abstract—As wireless mobile devices are more and more pervasive and adopted in critical applications, it is becoming increasingly important to measure the physical proximity of these devices in a secure way. Although various techniques have been developed to identify whether a device is close, the problem of identifying the *far proximity* (i.e., a target is at least a certain distance away) has been neglected by the research community. Meanwhile, verifying the far proximity is desirable and critical to enhance the security of emerging wireless applications. In this paper, we propose a secure far proximity identification approach that determines whether or not a remote device is far away. The key idea of the proposed approach is to estimate the far proximity from the unforgeable "fingerprint" of the proximity. We have validated and evaluated the effectiveness of the proposed far proximity identification method through experiments on real measured channel data. The experiment results show that the proposed approach can detect the far proximity with a successful rate of 0.85 for the non-Line-of-sight (NLoS) scenario, and the successful rate can be further increased to 0.99 for the Line-of-sight (LoS) scenario.

Index Terms—Far proximity identification; Fingerprinting; Channel impulse response

1 Introduction

As mobile platforms are more and more pervasive and adopted in critical applications, it is becoming increasingly important to measure the physical proximity of mobile devices in a secure way. For example, Implantable Medical Devices (IMDs) like pacemakers may grant access to an external control device only when that device is close enough [33]. As another example, contactless-payment systems (like Google Wallet), which enable users to make payments by placing a mobile device in the close proximity of a payment terminal, may require the mobile devices to be within several centimeters or even millimeters of the payment terminals.

Thus, verifying the *close proximity* has triggered significant attention and activity from the research community, and multiple techniques have been proposed to achieve the efficient identification of close proximity (e.g., [5], [7], [13], [14], [18], [32], [38]), including the well-known distance bounding protocols and their variants (e.g., [5], [32], [38]).

Although various techniques have been developed to identify whether a device is close, the problem of identifying the *far proximity* (i.e., a target is at least a certain distance away) has been neglected by the research community. Meanwhile, verifying the far proximity is desirable and critical to enhance the security of emerging wireless applications. By enforcing far proximity, in addition to traditional access control and cryptographic approaches, we can enhance the security of various critical wireless applications, such as satellite communication, long-haul wireless TV, radio, and alarm broadcasting, and Marine VHF radio for rescue and communication services [2].

For example, GPS devices receive signals, presumably from satellites in space, to determine their locations. Ideally, the GPS devices could verify that received signals are from far-away sources, to avoid being deceived by a nearby adversary's signals. In cellular networks, mobile phones may at times expect to receive signals from particular cell towers. It has been demonstrated that adversaries can set up a fake short-range cell tower to fool nearby mobile phones [24], [40]. To avoid being deceived by such a fake cell tower, it is desirable that mobile phones can authenticate that the signals they receive originate from a tower at an expected, further distance away.

Existing close proximity identification techniques (e.g., [7], [13], [18]) qualitatively decide whether or not a target is nearby, but they cannot be directly extended to address the far proximity identification problem. The qualitative decision that a target is not nearby doesn't quantitatively guarantee that the target is at least a certain distance away (i.e., in the far proximity).

Distance bounding protocols (e.g., [5], [32], [38] demonstrated their success in quantitatively estimating the distance between two wireless devices. However, they cannot be directly applied to enforce far proximity identification. In distance bounding protocols, a local device sends a challenge to a remote device, and the remote device replies with a response that is computed as a function of the received challenge. The local device then measures the round-trip time between sending its challenge and receiving the response, subtracts the processing delay from the round-trip time, and uses the result to calculate the distance between itself and the remote device. However, by delaying its response to a challenge, a dishonest remote device can appear to be arbitrarily further from the local device than it actually is.

In this paper, we develop a secure far proximity identification approach that can determine whether a remote device is far away. The key idea of the proposed approach is to estimate the proximity from the unforgeable "fingerprint" of the proximity. We develop a technique that can extract the fingerprint of a wireless device's proximity from the physical-layer features of signals sent by the device (i.e. channel impulse response). Since channel estimation is mandatory for all wireless systems to achieve reliable communications, mobile devices can easily extract a proximity fingerprint from an estimated channel impulse response. The

proximity fingerprints are closely related to the distance between the local and remote devices. They are easy to extract but difficult to forge. We also develop a novel technique that uses the proximity fingerprint to identify the lower bound of the distance between the local and the remote devices. We further propose a fine-grained proximity identification algorithm and derive both lower and upper bounds of the proximity between the local and the remote devices. Besides, we identify typical types of attacks against proposed schemes and propose the corresponding defense approaches.

The contributions of this paper are: (1) we develop a novel fingerprinting technique that enables the local device to extract the fingerprint of a wireless device's proximity from the physical-layer features of signals sent by the device; (2) we discover the theoretical relationship between the proximity and its fingerprint, and we developed a technique that can use such a relationship to estimate the lower and upper bounds of the distance between the local and remote devices; and (3) we validate and evaluate the effectiveness of the proposed far proximity identification method through experiments on the real-world data. The experiment results show that the proposed approach can detect the far proximity with a success rate of 0.85 for the non-Line-of-sight (NLoS) scenario, and the success rate can be further increased to 0.99 for the Line-of-sight (LoS) scenario.

The rest of the paper is organized as follows. Section 2 describes our assumptions and system and threat models. Section 3 presents the proposed far proximity identification techniques. Section 4 identifies typical types of attacks and proposes the corresponding countermeasures. Sections 5 and 6 discuss the experimental evaluation and related work. Section 7 concludes this paper.

2 System and Threat Models

To facilitate the presentation, we refer to the local device, which verifies the proximity, as the *verifier* and the remote device, whose proximity is being verified, as the *prover*. The verification system consists of a verifier and a prover. Both are equipped with radio interfaces that can transmit and receive wireless signals.

The verifier aims to determine whether or not a prover is at least a certain distance away, and it analyzes the signals emitted by the prover to achieve this goal. The verifier can work in both *active* or *passive* modes. In the active mode, the verifier sends a message to the prover to initialize the proximity identification, and the prover cooperates with the verifier by sending wireless signals back to the verifier to enable the verification. In the passive mode, instead of actively sending out signals, the verifier monitors the wireless channel to capture the prover's signal. Once the prover's signals are captured, the verifier can identify the prover's proximity.

We assume that the prover is untrusted. The prover may provide the verifier with fake messages and wrong configuration information regarding its hardware and software settings, such as device type, signal processing delay, and protocols in use. The prover may intentionally delay its replies to the verifier's messages or send bogus replies at any time to mislead the verifier. However, we assume that the verifier can receive wireless signals sent by the prover. As the long-haul wireless applications (e.g., space communications and TV broadcasting) usually have the stronger LoS feature, we assume that there are no metal shields on the straight line between the verifier and the prover to block wireless signals from the prover.

3 Far Proximity Verification

A simple and naive method to identify whether a prover is far away is to examine the received signal strength (RSS). A signal decays as it propagates in the air. Thus, it seems that strong RSS indicates a short signal propagation length and a close transmitter, whereas weak RSS strength implies a far-away transmitter. However, a dishonest prover can increase or decrease its transmit power to pretend to be close to, or far from, the verifier. The root reason for the failure of the naive method is that RSS can be easily forged. In this paper, we discover unforgeable and unclonable *fingerprints* of the proximity and propose techniques that can identify the far proximity based on these fingerprints.

3.1 Proximity Fingerprints

Because of the multipath effect [10], a signal sent by the prover generally propagates to the verifier in the air along multiple paths due to reflection, diffraction, and scattering. Each path has an effect (e.g., distortion and attenuation) on the signal traveling on it [27]. A *channel impulse response* characterizes the overall effects imposed by the multipath propagation, and it reflects the physical feature of a wireless link [10]. Because it is difficult to change the physical feature, channel impulse responses have been used as "**link signatures**" to uniquely identify the wireless link between a wireless transmitter and a receiver [6], [27], [43].

Figure 1 (a) shows a simple example of multipath propagation. The signal sent by the prover is reflected by an obstacle (i.e., a building), and thus it travels along Path 1 (the direct path from the prover to the verifier), and Path 2 (the reflection path). The signal copy that travels along one path is usually referred to as a *multipath component* [10]. Let $r1$ and $r2$ denote the multipath components that travel along Path 1 and Path 2 respectively. Figure 1 (b) is an example of the corresponding channel impulse response, which shows that $r1$ arrives at the verifier first and the peak of the signal amplitude of $r1$ is A_{r1}, and $r2$ arrives after $r1$, and its peak is A_{r2}.

Intuitively, if the prover increases (decreases) the transmit power, both A_{r1} and A_{r2} will increase (decrease), but the prover cannot adjust its transmit power such that it arbitrarily manipulates only one of A_{r1} and A_{r2}, because it is difficult for the prover to identify and modify the physical paths over which multipath components propagate [27]. On the other hand, the length of the signal propagation path is closely related to the amplitude of the received signal. A far-away prover results in weaker A_{r1} and A_{r2} than a close prover. Based on this intuition, we give the definition of proximity fingerprint below.

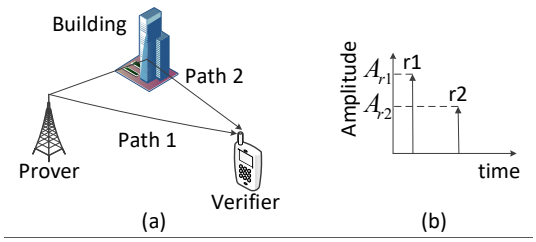


Fig. 1. An example of the multipath effect.

Definition 1 (Proximity Fingerprint) *Let A_{r1} and A_{r2} be the amplitudes of the first and the second received multipath components,*

respectively. The proximity fingerprint f is the ratio of A_{r1} to A_{r2}, i.e., $f = \frac{A_{r1}}{A_{r2}}$.

In Lemma 1, we prove that increasing or decreasing the transmit power does not affect the proximity fingerprint.

Lemma 1. Let P_t denote the transmit power. Let P_{r1} and P_{r2} be the amplitudes of the first and the second received multipath components. If the prover changes P_t to nP_t ($n > 0$), then both P_{r1} and P_{r2} will change to $\sqrt{n}P_{r1}$ and $\sqrt{n}P_{r2}$.

Proof: The amplitude P_r of a received signal can be modeled as [10]

$$P_r = \begin{cases} \sqrt{P_t k (\frac{d_0}{d})^\alpha} & d > d_0, \\ \sqrt{P_t k} & d \leq d_0, \end{cases} \quad (1)$$

where P_t is the transmit power, d is the length of the path along which the signal propagates from the transmitter to the receiver ($d > d_0$), k is a scaling factor whose value depends on the antenna characteristics and the average channel attenuation, d_0 is a reference distance for the antenna far-field, and α is the path loss exponent. The values of k, d_0, and α can be obtained either analytically or empirically [10]. Assume $d_1 > d_0$ and $d_2 > d_0$. Thus, according to Equation 3.2.1, P_{r1} and P_{r1} can be approximated by

$$P_{r1} = \sqrt{P_t k (\frac{d_0}{d_1})^\alpha}, \quad P_{r2} = \sqrt{P_t k (\frac{d_0}{d_2})^\alpha}, \quad (2)$$

where d_1 and d_2 are the lengths of the path along which the first and the second received multipath components travel respectively. If P_t is changed to nP_t ($n > 0$), then P_{r1} and P_{r2} will accordingly change to $\sqrt{n}P_{r1}$ and $\sqrt{n}P_{r2}$, and the proximity fingerprint (the ratio of P_{r1} to P_{r2}) remains the same. □

Note that due to the bandwidth limitation, the verifier can only distinguish two signals when their arrival time difference is larger than the resolvable time (i.e., 1/B, where B is the channel bandwidth). Therefore, P_{r1} and P_{r2} may not be the amplitude of received signals from exact first and second paths. Nevertheless, Lemma1 always holds as long as the prover cannot modify P_{r1} and P_{r2} simultaneously.

Key Features of Proximity Fingerprints: Lemma 1 shows that the prover cannot adjust its transmit power to arbitrarily manipulate the proximity fingerprint, but it appears that an attacker (i.e., a dishonest prover or a third-party adversary against benign provers) could affect the proximity fingerprint by intentionally placing a reflector nearby the prover to generate a fake path, in addition to the direct signal path from the prover to the verifier.

However, at the verifier's view, the direct and fake paths are still one unresolvable path if the difference between the arrival times of the signals traveling on both paths is much smaller than the symbol duration, which is the transmission time of a wireless physical-layer unit [10]. To be successful, an attacker has to place the reflector far enough away from the prover (i.e., δc meters, where δ is the symbol duration and c is the speed of light [10]), such that the difference between the two path arrival times is resolvable at the verifier. More crucially, at this distance the attacker must make sure that the prover's signal can exactly hit his reflector and be bounced back to the target verifier. However, it is quite uncertain for the prover's signal to be delivered to the reflector, then reflected by the reflector to the verifier due to the random scattering effect caused by long distance propagation [10].

For example, GPS satellites have a typical symbol duration of 0.01 second [1]. It is impractical for the satellite's signal to exactly hit a reflector that is 3,000,000 meters away, and moreover be reflected by the reflector to hit a target GPS navigation device on earth.

To summarize, proximity fingerprints are caused by wireless reflections somewhere, which the verifier does not need to know and identify. The verifier can easily extract A_{r1} and A_{r2} from the channel impulse response and compute the proximity fingerprint as A_{r1}/A_{r2}. Note that estimating the channel impulse responses is a must-have function for most modern wireless systems [10], [23]. But in order for the attacker to be successfully, the attacker has to know (1) how to pinpoint a far-away place to put a reflector or an active wireless device, and (2) exactly where to direct the reflector to shoot a needle in a haystack. Thus, significant practical hurdles exist for attacking proximity fingerprints. In this way, verifiers can easily extract proximity fingerprints, but it is difficult for attackers to forge or manipulate a specific fingerprint.

The attacker may also launch active attacks to undermine the verification of proximity fingerprints. In later section (4), we will discuss these active attacks and the corresponding countermeasures.

Impact of Directional Antennas: When directional antennas are used, the multipath effect may be reduced. However, directional antennas cannot provide perfect laser-like radio signals. For example, the beamwidth of a 3-element Yagi Antenna, the most common type of directional antenna, is 90 degrees in the vertical plane and 54 degrees in the horizontal plane [16]. Thus, it is not possible to completely eliminate the multipath effect, and accordingly the multipath propagation has been also considered in designing wireless communication systems equipped with directional antennas (e.g., [37], [42]). The proximity fingerprint can be calculated based on a very limited number of paths (i.e., two paths), and thus it is compatible to wireless systems with directional antennas in use.

Proximity fingerprint with single or many peaks: In cases where channel has more than two resolvable peaks, we still select the first two peaks to estimate the distance. That's because the first two peaks are usually largest ones in channel and more resilient to the channel noise.

The scheme cannot be applied to the scenario where CIR only has one peak. But such situation happens only when the signals are transmitted through the ideal environment (e.g. free space propagation) or within a narrow bandwidth. Practical GPS or cellular networks do not assume free space propagation. In addition, even the transmit signals have a narrow bandwidth (e.g. 5MHz or 10MHz), it's still highly likely that the CIR has multiple resolvable peaks due to the long propagation distance (e.g. thousands of meters). For example, if the signal bandwidth is 5MHz, the resolvable time is about 0.2 microsecond and thus the minimum path difference required to distinguish two peaks is about 60 meters. Since the transmitter is usually thousands of meters away from the receiver, it's highly possible that there exist two propagation paths whose distance difference is larger than 60 meters.

3.2 Far Proximity Identification Using Proximity Fingerprints

Based on the study of proximity fingerprint, we now reveal the relationship between the proximity fingerprint and the actual proximity, and we propose far proximity identification techniques that can provide fine granularity and lower bounds on proximity

(i.e., the prover is at least a certain distance away from the verifier) using the proximity fingerprint

3.2.1 Far Proximity Identification

To calculate the proximity of the prover, we first model the fingerprint of the proximity. We consider signal propagation in two typical wireless environments, i.e., the outdoor and the indoor environments.

There are multiple signal propagation models that characterize the path loss of wireless signals, such as the free space path loss model, ray tracing path loss models, the simplified path loss model, and empirical path loss models [10]. The common feature of these models is that they all indicate that the power of the transmitted signal decreases as the propagation distance increases. In the channel impulse response, each resolvable multipath component is the superposition of multiple non-resolvable signals arriving within the resolvable time. Because the empirical path loss model is able to characterize the path loss in complex propagation environments, we would like to apply the empirical model to quantitatively estimate the amplitude of resolvable multipath component in channel response. In the following discussion, without loss of generality, we focus on two well-known propagation models (Okumura Model [10] and ITU Indoor Propagation Model [23]) for both outdoor and indoor environments.

We assume that there are no large metallic obstacles that can significantly block the straight line propagation between the verifier and the prover. Thus, the first received multipath component normally travels along the straight line due to the penetration and diffraction-around-object effect, and the propagation distance is approximately d meters, where d is the distance between the verifier and the prover. The second received multipath component travels along a reflection path. Assume that the difference between the arrival times of the first and the second multipath components is Δt. The propagation distance of the second arrived multipath component is thus $d + \Delta tc$ meters, where c is the speed of light.

Outdoor signal propagation: One of the most common models for outdoor signal propagation in urban, suburban, and rural areas is the Okumura Model [10]. According to the Okumura model, the signal path loss in decibels (dB) in urban areas can be modeled as

$$\begin{aligned} L(\text{dB}) &= 69.55 + 26.16\log_{10}(f_c) - 13.82\log_{10}(h_{te}) \\ &- a(h_{re}, f_c) + (44.9 - 6.55\log_{10}(h_{te}))\log_{10}(d), \end{aligned}$$

where d is the length of the path along which the signal propagates from the transmitter to the receiver, f_c is the central frequency, h_{te} and h_{re} are the transmitter's and the receiver's antenna heights respectively, and $a(h_{re}, f_c)$ is a correction factor computed using h_{re} and f_c [10]. Based on the Okumura Model, we give Lemma 2

Lemma 2. The proximity fingerprint in the outdoor environment is $\sqrt{(\frac{d_2}{d_1})^{\frac{\gamma}{10}}}$, where d_1 and d_2 are the lengths of the paths along which the first and the second received multipath components travel respectively, $\gamma = 44.9 - 6.55\log_{10}(h_{te})$, and h_{te} is the transmitter's antenna height.

Proof: The received signal power P_r can be represented as P_r(dB) = P_t(dB) - L (dB), where P_t is the transmit power. To facilitate the calculation, we change the unit of P_r from dB to watt (W). The relationship between P_r(dB) and P_r(W) is $P_r(\text{dB}) = 10\log_{10} P_r(\text{W})$. Thus, we can derive

$$P_r(\text{W}) = 10^{\frac{1}{10}(P_t(\text{dB}) - L(\text{dB}))} = \frac{10^{\frac{1}{10}P_t(\text{dB})}}{10^{\frac{1}{10}L(\text{dB})}} = \frac{P_t(\text{W})}{L(\text{W})}$$

Similarly, we can derive $L(\text{W})$ as

$$L(\text{W}) = 10^{\frac{1}{10}L(\text{dB})} = 10^{\frac{1}{10}(\beta + \gamma\log_{10}(d))},$$

where $\beta = 69.55 + 26.16\log_{10}(f_c) - 13.82\log_{10}(h_{te}) - a(h_{re}, f_c)$ and $\gamma = 44.9 - 6.55\log_{10}(h_{te})$.

The amplitude of a signal is the square root of the received signal power. Thus, the amplitudes A_{r1} and A_{r2} of the first and the second received multipath components can be represented by

$$A_{r1} = \sqrt{P_{r1}(\text{W})} = \sqrt{\frac{P_t(\text{W})}{10^{\frac{1}{10}(\beta + \gamma\log_{10}(d_1))}}}$$

$$A_{r2} = \sqrt{P_{r2}(\text{W})} = \sqrt{\frac{P_t(\text{W})}{10^{\frac{1}{10}(\beta + \gamma\log_{10}(d_2))}}}$$

where d_1 and d_2 are the lengths of the paths along which the first and the second received multipath components travel respectively. Note that both multipath components have the same values for γ and β, because they are from the same signal source (i.e., the prover) and exhibit the same frequency f_c. Thus, the proximity fingerprint f can be written as

$$f = \frac{A_{r1}}{A_{r2}} = \sqrt{(\frac{d_2}{d_1})^{\frac{\gamma}{10}}}. \tag{3}$$

According to the Okumura model, the signal path loss models in suburban and rural areas are, respectively,

$$L_{suburban}(\text{dB}) = L(\text{dB}) - 2[\log_{10}(f_c/28)]^2 - 5.4,$$

and

$$L_{rural}(\text{dB}) = L(\text{dB}) - 4.78[\log_{10}(f_c)]^2 + 18.33\log_{10}(f_c) - K,$$

where K ranges from 35.94(countryside) to 40.94 (desert). By using the same analytical approach, we can obtain the similar result that the proximity fingerprint in the suburban and rural areas is $\sqrt{(\frac{d_2}{d_1})^{\frac{\gamma}{10}}}$. □

Indoor signal propagation: The path loss in the indoor environment can be usually represented by the ITU Indoor Propagation Model [23] as shown below

$$L(\text{dB}) = 20\log f_c + \lambda\log d + P_f(N_f),$$

where λ is the empirical path loss at the same floor, N_f denote the number of floors between the transmitter and receiver, and $P_f(N_f)$ denotes the floor penetration loss. Based on the ITU indoor model, we give Lemma 3

Lemma 3. The proximity fingerprint in the indoor environment is $\sqrt{(\frac{d_2}{d_1})^{\frac{\lambda}{10}}}$, where d_1 and d_2 are the lengths of the paths along which the first and the second received multipath components travel respectively, and λ is the empirical floor penetration loss factor.

Proof: As discussed earlier, the received signal power P_r can be represented as P_r(dB) = P_t(dB) - L (dB). By converting the unit of P_r from dB to W, we can obtain

$$P_r(\text{W}) = \frac{P_t(\text{W})}{L(\text{W})} = \frac{P_t(\text{W})}{10^{\frac{1}{10}(20\log f_c + \lambda\log d + P_f(N_f))}}$$

The proximity fingerprint, the ratio of A_{r1} to A_{r2} can be written as

$$f = \frac{\sqrt{P_{r1}(\mathrm{W})}}{\sqrt{P_{r2}(\mathrm{W})}} = \sqrt{(\frac{d_2}{d_1})^{\frac{\lambda}{10}}} \tag{4}$$

□

Far proximity identification: Assume there are no large metallic obstacles that can significantly block the signal propagation between the verifier and the prover. The path that the first received multipath component usually travels along (i.e., Path 1) is roughly straight between the verifier and the prover due to penetration and diffraction-around-obstacles features of wireless signals [10]. Thus, d_1 approximately equals to the distance between the verifier and the prover. The lower bound of d_1 is given in Lemma 4.

Lemma 4. Let d be the distance between the prover and the verifier. We have $d \geq \frac{c}{B(f^{\frac{2}{\alpha}}-1)}$, where c is the speed of light, B is the bandwidth of the communication system, α is the path loss exponent, and f is the proximity fingerprint.

Proof: Let t denote the time at which the prover's signal starts to propagate to the verifier. Let t_1 and t_2 denote the arrival times of the first and the second received multipath components, respectively. Therefore, $d_1 = (t_1 - t)c$ and $d_2 = (t_2 - t)c$, and we have the following:

$$d_2 = (t_2 - t)c = (t_1 - t)c + (t_2 - t_1)c = d_1 + \Delta c,$$

where $\Delta = t_2 - t_1$. From Equations 3 and 4, we know that for both the outdoor and indoor environments, the proximity fingerprint f can be generalized by the same expression $f = \sqrt{(\frac{d_2}{d_1})^{\alpha}}$, where α equals to $\frac{\gamma}{10}$ and $\frac{\lambda}{10}$ for the outdoor and indoor propagation respectively. The first received multipath component travels along the straight line between the verifier and the prover. Hence, the distance d between the verifier and the prover is equal to d_1. According to [10], for resolvable multiple path components, $\Delta \geq \frac{1}{B}$, where B is the bandwidth of the wireless communication system. Thus,

$$f = \sqrt{(\frac{d_2}{d})^{\alpha}} = \sqrt{(\frac{d + \Delta c}{d})^{\alpha}}$$

and we have

$$f \geq \sqrt{(\frac{d + \frac{c}{B}}{d})^{\alpha}},$$

and

$$d \geq \frac{c}{B(f^{\frac{2}{\alpha}} - 1)}. \tag{5}$$

□

Fine-grained proximity identification: A more accurate time difference estimation between arrivals can be obtained from the measured channel impulse response. Figure 2 shows an example of a real-measured channel impulse response obtained from the CRAWDAD data set [36], which contains channel impulse responses collected in an indoor environment with obstacles (e.g., cubicle offices and furniture) and scatters (e.g., windows and doors). As shown in Figure 2, the time difference between the first and second peaks is about 75 nanoseconds. Since signals arriving within the resolution time cannot be distinguished from each other, the estimation error is considered as $\pm\frac{1}{2B}$, where B is the bandwidth of the communication system. Assume the time difference observed from the channel impulse response is δt, we can model the time difference range as $(\delta t - \frac{1}{2B},\ \delta t + \frac{1}{2B})$.

We would like to utilize such fine-grained time difference to further refine the proximity identification. In addition, with the range of the time difference, we can yield both lower and upper bounds of the proximity between the verifier and the prover. Assume the estimated time difference is within $(\delta t - \frac{1}{2B},\ \delta t + \frac{1}{2B})$. We have Lemma 5 as stated below:

Lemma 5. Let d be the distance between the prover and the verifier. d can be estimated within the range of $(\frac{(\delta t - \frac{1}{2B})c}{f^{\frac{2}{\alpha}}-1},\ \frac{(\delta t + \frac{1}{2B})c}{f^{\frac{2}{\alpha}}-1})$, where c is the speed of light, α is the path loss exponent, and f is the proximity fingerprint.

Proof: In the proof of Lemma 4, we generalize the proximity fingerprint f as $f = \sqrt{\frac{d_2}{d_1}^{\alpha}}$, where d_1 and d_2 are the propagation distance of the first and second multipath components respectively. Since the first received multipath component approximately travels along the straight-line between the verifier and the prover, we have $d = d_1$.

Assume the time difference between two multipath components is Δt. We have the relationship between d_1 and d_2 as $d_2 = d_1 + \Delta t c$, where c is the speed of light. By substituting the relationship into the proximity fingerprint, we can derive the distance as the following equation 6:

$$d = \frac{\Delta t c}{f^{\frac{2}{\alpha}} - 1}. \tag{6}$$

Since the time difference Δt is within the range of $(\delta t - \frac{1}{2B},\ \delta t + \frac{1}{2B})$, where δt is the time difference observed from the corresponding channel impulse response, we can have the upper bound of the distance between the prover and verifier by substituting $\Delta t \leq \delta t + \frac{1}{2B}$.

$$d \leq \frac{(\delta t + \frac{1}{2B})c}{f^{\frac{2}{\alpha}} - 1}.$$

Similarly, lower bound of the proximity can be obtained by substituting $\Delta t \geq \delta t - \frac{1}{2B}$ into the equation 6.

$$d \geq \frac{(\delta t - \frac{1}{2B})c}{f^{\frac{2}{\alpha}} - 1}.$$

Therefore, we have the proximity range between the prover and the verifier as $(\frac{(\delta t - \frac{1}{2B})c}{f^{\frac{2}{\alpha}}-1},\ \frac{(\delta t + \frac{1}{2B})c}{f^{\frac{2}{\alpha}}-1})$. □

Choosing α: For the outdoor signal propagation, according to the Okumura model, $\gamma = 44.9 - 6.55\log_{10}(h_{te})$, where h_{te} is the height of the transmitter's antenna. If the verifier has specific types of targets, for example, the verifier aims to verify the proximity of a satellite, a cellular base station, or a TV tower, then the verifier can directly compute γ by looking up the typical values of h_{te} from the corresponding wireless device handbooks. Alternatively, the verifier can also get an estimate of γ by using the typical transmitter antenna height in the outdoor environment (e.g., the typical transmitter antenna height ranges between 1 to 200 meters [23], and thus γ approximately lies between 44.9 and 29.83). After obtaining γ, the verifier can compute $\alpha = \frac{\gamma}{10}$. For the indoor signal propagation, $\alpha = \frac{\lambda}{10}$, where λ is the indoor path loss factor that doesn't rely on the antenna height and it can be obtained through empirical experiments.

Note that the path loss exponent α for both outdoors and indoors can be actually regarded as an attenuation factor that

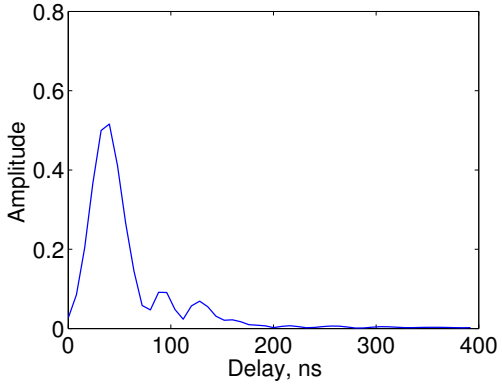


Fig. 2. An example of the real-measured channel impulse response obtained from the CRAWDAD data set

reflects the attenuation caused by the propagation path. Previous studies have performed extensive empirical experiments to measure typical values of such an attenuation factor in different wireless environments [10]. For example, the attenuation factor is 2.0 for vacuum free space, 2.7–3.5 for urban areas, 3.0–5.0 for suburban areas, and 1.6–1.8 for indoors [10]. In the following discussion, without loss of generality, we use these typical empirical values of the attenuation factor as the example α. Nevertheless, the verifier can obtain α empirically using existing readily-available approaches (e.g., [3], [22]), and a real-measured attenuation factor can help to improve the accuracy of the proximity lower bound estimation.

3.2.2 Experimental Examples

Figure 3 shows the estimated lower bound of the proximity as a function of the proximity fingerprint f. The speed of light c is 2.99792458×10^8, and the bandwidth B is 20 Mbps. From Figure 3, we can see that the proximity lower bound of the prover decreases as the proximity fingerprint f increases. The indoor environment has the smallest α, and with $f = 5$ the verifier can know that the prover is at least 3.01 meters away. The suburban environment has the largest α, and with $f = 5$ the verifier can know that the prover is at least 16.59 meters away.

As mentioned, figure 2 shows an example of a real-measured channel impulse response obtained from the CRAWDAD data set. The channel impulse response was measured when the distance between the transmitter and the receiver is 4.09 meters. From Figure 2, we can see that each received multipath component leads to a triangle in shape with a peak [27]. The second multipath component arrives at the receiver about 75 nanoseconds after the arrival of the first one. The proximity fingerprint is 5.6499. The channel impulse response was measured indoors, and thus α ranges between 1.6 and 1.8.

We use Lemma 4 to estimate the lower bound of the proximity of the transmitter, and Figure 4 shows the result. We can observe that the estimated lower bound increases as α increases. However, when α reaches the maximum value (i.e., 1.8) of the indoor environment, the real distance is still bounded by (i.e., greater than) the estimated lower bound. Specifically, when $\alpha = 1.8$, the lower bound of the proximity is 3.84 meters. This means the transmitter should be at least 3.84 meters away from the receiver. The actual distance between the transmitter and the receiver is 4.09 meters, which is slightly greater than the lower bound 3.84 meters.

Note that long-haul communications may desire a much relaxed tightness of the proximity lower bound. For example, GPS satellites running on the Low Earth Orbit have an altitude of approximately 2,000,000 meters (1,200 miles). With a proximity lower bound of 1,000,000 meters (i.e., the bound is less than the actual proximity by 50%), it would be possible to prevent most attackers from impersonating the satellites, because it is usually very difficult for the attacker to achieve such a long transmission range.

3.3 System Design

In what follows, we show how the theoretical result of Lemma 4 can be used in a practical communication system to achieve the far proximity identification.

The verifier's objective is to find out the proximity lower bound of the prover, i.e., to verify that the prover is at least a certain distance away. According to Lemma 4, the proximity lower bound is computed by $\frac{c}{B(f^{\frac{2}{\alpha}}-1)}$. Thus, the verifier can simply compute this bound with the knowledge of the speed of light c, the system bandwidth B, the path loss exponent α, and the proximity fingerprint f. The speed of light c is a universal physical constant and the bandwidth B is a system configuration parameter, and both of them are known to the verifier. The path loss exponent α can be either obtained empirically, or can be determined using the typical values. The proximity fingerprint f is the only remaining factor that the verifier needs to decide to compute the lower bound.

As we discussed earlier, the fingerprint f is the ratio of A_{r1} to A_{r2}, where A_{r1} and A_{r2} are the amplitudes of the first and the second received multipath components. A_{r1} and A_{r2} can be extracted from the channel impulse response. A wireless packet is usually preceded by a preamble, a special data content that indicates the beginning of an incoming packet. When the prover sends a packet to the wireless channel, the verifier will first capture the preamble using the match filtering technique [12]; then the verifier knows that there is an incoming packet and continues to receive the payload. The preamble not only enables packet capture, but also enables the estimation of the channel impulse response at the verifier.

After receiving the preamble, the verifier can use existing channel estimation techniques (e.g., least-square (LS) and linear minimum mean squared error (LMMSE) estimators [4]) to estimate the channel impulse response from the preamble, and thereby obtain the values of A_{r1} and A_{r2} and the proximity fingerprint $f = A_{r1}/A_{r2}$. It is worth pointing out that using the preamble is not the only way to obtain A_{r1} and A_{r2}. The verifier can also use blinding estimation methods (e.g., [39]) to estimate the channel impulse response from the entire content of the preamble and the payload. In addition, the verifier can use hybrid methods (e.g., [15]) that combine preamble-based estimation and blind estimation together to improve the estimation accuracy. After obtaining the proximity fingerprint f and demodulating the payload and authentication information, the verifier then verifies the prover's proximity using Lemma 4.

Dealing with the Wireless Uncertainty: Wireless channels can be affected by random environmental factors like temperature, humidity, and vegetation. Thus, the estimated channel impulse response may be time-varying and fluctuate around a center value. To improve the proximity authentication accuracy, instead of using only one channel impulse response to estimate the proximity, we

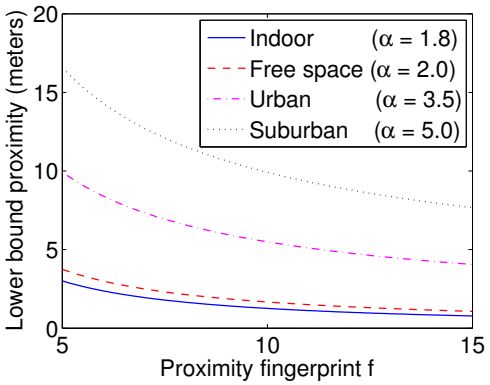


Fig. 3. The estimated lower bound of the proximity as a function of the proximity fingerprint f: The indoor environment has the smallest α, and with $f = 5$ the verifier can know that the prover is at least 3.01 meters away. The suburban environment has the largest α, and with $f = 5$ the verifier can know that the prover is at least 16.59 meters away.

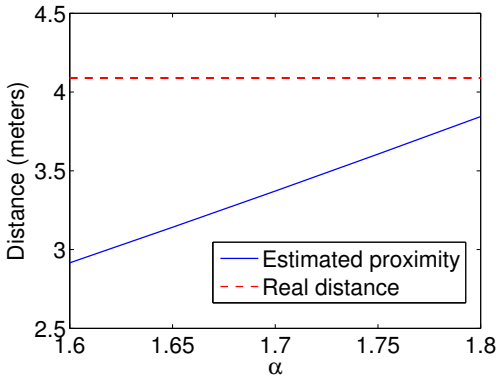


Fig. 4. Estimated lower bound v.s. the real distance regarding different path loss exponent α

propose to estimate the proximity based on multiple channel impulse responses, which are collected over a certain time window. Each channel impulse response can yield a set of amplitude ratios, and the corresponding set of estimated proximity. Suppose there are L multiple paths and n channel impulse responses, the verifier will obtain a total of $L * n$ estimates of the prover's proximity. The verifier then uses the mean value of these estimates as the proximity authentication output, so that the impact of random noises can be mitigated with the boosted size of the sample space.

In addition, the path loss exponent α plays an important role in authenticating the proximity. To make α resilient against environmental changes, we propose to use training phases to calibrate α periodically. In a training phase, the verifier estimates the proximity of an authenticated beacon transmitter, whose real distance is already known to the verifier. The verifier compares the estimated proximity with the real distance between the beacon transmitter and itself. Based on the comparison result, the verifier adjusts α so that the difference between the estimation output and real distance can be minimized. After the training phase, the verifier uses calibrated α to identify the proximity of an unknown wireless device.

By averaging over multiple channel measurements and using a real time α, the verifier can cope with environment changes and improve the proximity authentication performance.

3.4 Implication

Lemma 4 indicates that the prover is at least $\frac{c}{B(f^{\frac{2}{\alpha}}-1)}$ meters away from the verifier. This range is determined by the speed of light c, the system bandwidth B, the path loss exponent α, and the proximity fingerprint f. Note that c, α, and B are system constants that are determined by the physical features of the propagation medium. Thus, they are are not manipulatable. Also, there are significant practical hurdles to manipulate the proximity fingerprint f, as described in Section III-A. Therefore, the provers can prove (or cannot repudiate) that it is at least $\frac{c}{B(f^{\frac{2}{\alpha}}-1)}$ meters away.

On the other hand, Lemma 4 reveals a good feature of the proposed technique, i.e., it supports passive proximity identification. As we discussed earlier, c, α, and B are system constants that are already known to the verifier. The proximity fingerprint f is computed based on channel impulse responses. Existing channel estimation techniques are typically passive, and they do not rely on active two-way interactions between the prover and the verifier to estimate channel impulse responses. With the knowledge of c, α, B, and f, the verifier can directly compute the proximity lower bound of the prover. The passive verification not only reduces communication overhead, but also increases the difficulty for an adversary to recognize an on-going proximity identification activity.

4 Attacks and Countermeasures

A proximity fingerprint itself is unforgeable, because it is extracted from channel impulse responses, which have been regarded as "signatures" to uniquely identify the wireless link between a transmitter and a receiver. However, an attacker may launch attacks targeting at the verification decision process such that the verifier gets an incorrect verification decision. Specifically, the attacker is able to intercept, interfere, or even jam the signal transmission between the prover and verifier, and aims at causing false negative/positive errors to fool the verifier to get a wrong decision on a dishonest/benign prover. In addition, a dishonest prover may manipulate the channel estimation process to create a fake channel impulse response at the verifier or collaborate with attackers to generate a mixed received signal to fool the verifier with a wrong decision.

To fool the verifier, the attacker may try to collaborate with another active wireless device or equip with multiple antennas to create a fake second path by transmitting signals from a different direction. In this case, the attacker must make sure that there is no multipath effect for the signals traveling on the direct path (e.g., the path from the prover to the verifier) and the fake path (e.g., the path from the active wireless device to the verifier). Otherwise, the attacker cannot control and guarantee that the fake path is exactly the second received path at the verifier side. Eliminating the multipath effect completely is normally regarded as infeasible.

In this paper, we focus on three other major attacks against the far proximity fingerprinting and they are:

- **Jam-and-replay attack:** The attacker may jam the prover and replay an intercepted signal to fool the verifier taking the attacker's proximity as the prover's proximity.
- **Flipping attack:** The attacker may collaborate with malicious provers to generate mixed received signals at the

verifier, and thus result in the false negative and positive errors.

- **Spoofing attack:** A dishonest prover may try to create a fake channel impulse response at the verifier by manipulating the channel estimation process.

4.1 Dealing with Jam-and-replay Attacks

4.1.1 Attack methodology

In the jam-and-replay attack, the attacker first intercepts the transmit signal from the prover, and at the same time jams the transmission to prevent the verifier from receiving the original signal from the prover. Then the attacker replays the intercepted signal from the prover at the attacker's own location, such that the verifier is fooled into taking the attacker's proximity as the prover's proximity. Because the attacker jams the original transmission between the prover and verifier, traditional anti-replay mechanisms such as sequence numbers do not work.

4.1.2 Defense approach

A common method of addressing jam-and-replay attacks is to explore timestamps (e.g., [18]). In such a method, the sender includes a timestamp in the transmitted message, which indicates the time when a particular bit or byte called the anchor (e.g., the start of the message header) is transmitted over the air. Upon receiving a frame, the receiver can use this timestamp and its local message receiving time to estimate the message traverse time. An overly long time indicates that the message has been forwarded by an intermediate attacker.

Timestamps-based method requires clock synchronization between the sender and the receiver, but it generally has a low synchronization requirement in common wireless applications. For example, in an 11 Mbps 802.11g wireless network, the transmission of a typical 1500-byte TCP message requires 1.09 (i.e., $\frac{1500*8}{11\times10^3}$) milliseconds. Thus, the attacker at least doubles the transmission time of the message to 2.18 milliseconds. As long as the verifier and the prover have coarsely synchronized clocks that differ in the order of milliseconds, the verifier can detect jam-and-replay attacks. In practice, multiple schemes can be applied to satisfy such clock synchronization requirement to detect the attack [21]. For example, in IEEE 802.11 standard, it specifies the timing synchronization function (TSF) to fulfill timing synchronization among users. Since the TSF is based on a 1 MHz clock and "ticks" in microseconds, it can achieve the time accuracy in the range of few microseconds (us), which is orders of magnitude smaller than milliseconds (ms). Note that the synchronization requirement can be further relaxed in GPS applications. GPS satellites have a transmission rate ranging between 20 bits/s and 100 bits/s [1]. The transmission of a standard 1500-bits GPS navigation message [1] takes 15 – 75 seconds, and accordingly the synchronization accuracy can be reduced to the order of seconds.

In addition, to launch jam-and-replay attacks, the attacker must send jamming signals to jam the wireless transmission. Jamming attacks have been extensively studied in the literature, and various techniques regarding jamming detection and countermeasures have been proposed (e.g., [10], [17], [34]). The prover and the verifier can also use existing jamming detection or anti-jamming techniques to discover the presence of jam-and-replay attacks, or to defend against such attacks.

4.2 Dealing with Flipping Attacks

4.2.1 Attack methodology

The attacker can collaborate with malicious provers or interfere legitimate provers to generate a mixed received signals at the verifier, and thus result in a flipped decision (i.e., in far proximity $\rightarrow$ out of far proximity and out of far proximity $\rightarrow$ in far proximity). Specifically, the attacker uses its own proximity to "pollute" the prover's proximity. Assume the prover and the attacker are d_p and d_a meters away from the verifier respectively, where $d_p \gg d_a$ or $d_a \gg d_p$. The attacker sends signals to the verifier along with the transmission of the prover. Suppose the attacker and the prover do not overwrite each other's signal (e.g., by using a very high transmission power). Thus, the verifier receives a mixed signal formed by both the prover and the attacker's signals. Intuitively, the proximity fingerprint extracted from the mixed signal will reflect proximity features of both the attacker and the prover, and thus the corresponding proximity lower bound d estimated based on the proximity fingerprint can greatly deviate from the real proximity lower bound d_p of the prover.

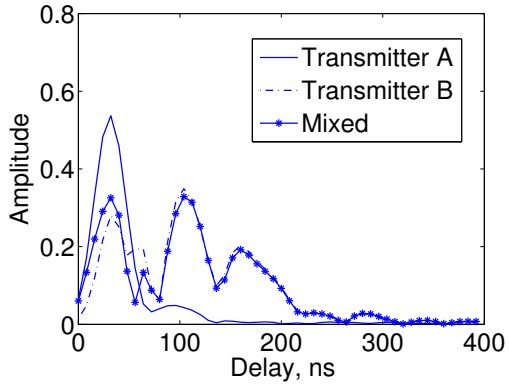


Fig. 5. Example of flipping attacks: the mixed channel impulse response reflects proximity features of both the transmitter A and B.

We conducted experiments using the CRAWDAD dataset to examine the impact of the attacker. The dataset includes over 9,300 real channel impulse response measurements in a 44-node wireless network [36]. Two transmitters (nodes 31 and 35) and a receiver (node 44) from the data set are used for the experiments. Figure 5 shows the real measured channel impulse responses between the transmitters and the receiver. Transmitter A is positioned 1.83 meters away from the receiver. The proximity fingerprint, the amplitude ratio of the first received multipath component to that of the second one, is about 11.06. Accordingly, the estimated lower bound of the transmitter's proximity is 1.67 meters. Compared to transmitter A, transmitter B is farer away from the receiver. The distance between transmitter B and the same receiver is 14.15 meters. Transmitter B's proximity fingerprint is about 1.80 and the estimated proximity lower bound is 12.99 meters. We let transmitters A and B send signals to the receiver at the same time. Thus, the receiver receives a mixed signal from both transmitters. The mixed signal can result in a channel impulse response as shown in Figure 5. The proximity fingerprint estimated from this channel impulse response is 2.44, and the corresponding estimated proximity lower bound is 8.85 meters, which falls between the true bounds of transmitter A (1.67) and transmitter B (12.99).

The experiment results show that it is possible for an attacker to use its own proximity to significantly affect the estimated

proximity lower bound of the prover. Consequently, a nearby attacker may fool the receiver into believing that a far-away prover is not far away, and vice versa.

4.2.2 Defense approach

A basic solution to deal with flipping attacks is to use existing jamming detection approaches. The attacker's signals cause the wireless interference to the transmission, and thus they can be regarded as jamming signals. Jamming attacks have been extensively studied in the literature, and various techniques regarding jamming detection have been proposed (e.g., [9], [11], [17], [29]–[31], [34], [41]). The verifier may use existing jamming detection techniques to discover the presence of flipping attacks.

However, one drawback of jamming detection techniques is that they require the attacker to constantly jam the prover's transmission for a relatively long time, such that the receiver can collect enough jammed signal samples. By analyzing those samples, the receiver can obtain important statistical values, including packet loss rate, bit error rate, and received signal strength. Those values enable the receiver to make a decision regarding whether or not the communication system is under jamming attacks. However, if the attacker jams the transmission for a short time, the receiver may not be able to get an accurate estimate of those statistical values, thereby reporting an incorrect decision.

We propose advanced defense techniques to deal with flipping attacks. Traditional jamming detection techniques require constant long-term jamming, because the detection decision is based on statistical values obtained from a certain amount of jammed signal samples. To deal with a short-term flipping attacker, we design defense techniques without relying on statistical values. Intuitively, due to the absence or imperfect synchronization, there exists a tiny clock discrepancy between the attacker and the prover. This clock discrepancy is actually a time-varying random variable. When there exist flipping attacks, the attacker's and the prover's signals mix together, and the clock discrepancy introduces randomness to the mixed signal.

We performed experiments using the CRAWDAD dataset to compare the channel impulse responses obtained from non-mixed and mixed signals. We considered a normal scenario and an attack scenario. In the first scenario, only the prover transmits signals to the verifier, while in the second one, the attacker launches flipping attacks by interfering the prover's transmission. Figure 6 shows 5 channel impulse responses in the normal scenario and each of them is extracted from a short-term signal that lasts for about 250 nanoseconds. We can observe that those channel impulse responses are similar to each other in shape. Figure 7 shows 5 channel impulse responses extracted from the mixed signals in the attack scenario. Unlike the normal scenario, the channel impulse responses exhibit random shapes and they are quite different from each other.

Therefore, we can detect the presence of flipping attacks through checking the consistency among channel impulse responses. Specifically, we can compute the difference between successive channel estimations. The channel is considered as polluted, if two successive estimated channels change significantly. Specifically, assume we have two successive channel estimations $\mathbf{h_i}$ and $\mathbf{h_j}$, their difference is denoted as the Euclidean distance $d_{ij} = ||\mathbf{h_i} - \mathbf{h_j}||$. Then we compare d_{ij} with a threshold τ, for a constant $\tau > 0$. When $d_{ij} > \tau$, the channel is considered as polluted and a flipping attack is detected.

The flipping attack detection can be viewed as a choice between two events F_0 and F_1, where F_0 indicates the event of a normal scenario, while F_1 indicates the event of a flipping attack. The density functions conditioned on F_0 and F_1 can be denoted as $f_{d_{ij}}(d_{ij}|F_0)$ and $f_{d_{ij}}(d_{ij}|F_1)$ respectively. Accordingly, we can obtain the probability of false alarm and missed detection as $P_f = \int_{x=\tau}^{\infty} f_{d_{ij}}(x|F_0)dx$ and $P_m = \int_{x=0}^{\tau} f_{d_{ij}}(x|F_1)dx$ respectively. As both probabilities are functions of the threshold τ, there is a tradeoff between the false alarm rate and missed detection rate. In practice, we may empirically select a threshold τ to achieve a high detection rate and at the same time, maintain a relatively low false alarm rate. In addition, to further minimize the false alarm caused by the normal channel fading, we compare the differences among n channel estimations($n > 2$). The channel will be treated as polluted, only when p out of n channel estimations are dramatically changed, where p is the threshold of the flipping attack indicator.

4.3 Dealing with Spoofing Attacks

4.3.1 Attack methodology

Instead of creating real-world fake paths, the attacker may target the channel estimation process, such that the verifier obtains a fake impulse response specified by the attacker. Let $\mathbf{S_v}$ denote the symbols (i.e. transmission units in physical layer) received by the verifier, and $\mathbf{S_v}$ can be represented as $\mathbf{S_v} = \mathbf{h} * \mathbf{S_p} + \mathbf{n_v}$, where $\mathbf{S_p}$ are the preambles, $\mathbf{h}$ is the actual channel impulse response between the verifier and the prover, and $\mathbf{n_v}$ is the channel noise, respectively. Upon receiving $\mathbf{S_v}$, the verifier uses $\mathbf{S_p}$ and $\mathbf{S_v}$ as the input of the channel estimation algorithm (e.g., LS and LMMSE), and the output of the algorithm is the estimated channel impulse response.

In general, the prover and the verifier must agree on the same preambles to achieve the accurate channel estimation. However, it is possible for a dishonest prover to specify a fake channel impulse response by modifying the transmit preambles. Let $\mathbf{S'_p}$ denote the preambles to be transmitted by a dishonest prover, and let $\mathbf{S'_v}$ denote the corresponding received symbols. $\mathbf{S'_v}$ can be represented by $\mathbf{S'_v} = \mathbf{h} * \mathbf{S'_p} + \mathbf{n'_v}$ [28]. Further let $\mathbf{h_a}$ denote the fake channel estimation result chosen by the dishonest prover. The goal of the dishonest prover is to find symbols $\mathbf{S'_p}$, such that when $\mathbf{S'_p}$ arrive at the verifier, the corresponding received symbols $\mathbf{S'_v}$ can result in an estimated channel impulse response that is equal to $\mathbf{h_a}$. To achieve this goal, the prover let $\mathbf{S'_v} = \mathbf{h_a} * \mathbf{S_p} + \mathbf{n_v}$, i.e., $\mathbf{h} * \mathbf{S'_p} + \mathbf{n'_v} = \mathbf{h_a} * \mathbf{S_p} + \mathbf{n_v}$.

Upon receiving $\mathbf{S'_v}$, the verifier uses $\mathbf{S_p}$ and $\mathbf{S'_v}$ to estimate the channel impulse response. Because $\mathbf{S'_v} = \mathbf{h_a} * \mathbf{S_p} + \mathbf{n_v}$, the estimated channel impulse response will be equal to $\mathbf{h_a}$. We rewrite the equation $\mathbf{h}*\mathbf{S'_p}+\mathbf{n'_v} = \mathbf{h_a}*\mathbf{S_p}+\mathbf{n_v}$ as $\mathbf{h}*\mathbf{S'_p}+\mathbf{n} = \mathbf{S}$, where $\mathbf{S} = \mathbf{h_a} * \mathbf{S_p}$ and $\mathbf{n}$ $(= \mathbf{n'_v} - \mathbf{n_v})$ is the white Gaussian channel noise. By using the standard least square approach [4], we can obtain that $\mathbf{S'_p} = (\mathbf{H}^H\mathbf{H})^{-1}\mathbf{H}^H\mathbf{S}$, where $\mathbf{H}$ is the Toeplitz matrix of $\mathbf{h}$. By sending $\mathbf{S'_p}$ to the verifier, the prover can fool the verifier to obtain a fake channel results ha. Because all elements in ha are chosen by the dishonest prover, the verifier will obtain fake amplitude ratios that are specified by the prover.

4.3.2 Defense approach

To launch a successful spoofing attack, the dishonest prover must first know the actual channel $\mathbf{h}$ between the prover and the verifier. Otherwise, the attacker can only generate a unpredictable channel

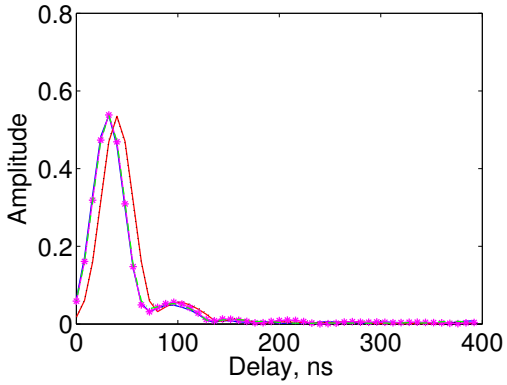


Fig. 6. Channel impulse response measured in normal scenario

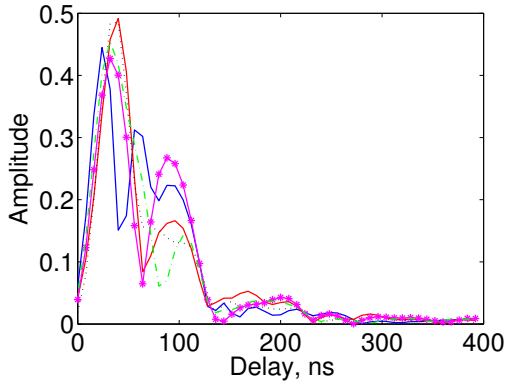


Fig. 7. Channel impulse response measured in the scenario of flipping attacks

estimation by randomizing the preambles. To defend against the attack, we propose to introduce a passive auxiliary node, which we refer to as the helper. Specifically, the prover and the verifier agree on two different preambles $\mathbf{S_{p1}}$ and $\mathbf{S_{p2}}$, and the verifier and the helper use $\mathbf{S_{p1}}$ and $\mathbf{S_{p2}}$ to estimate the channel impulse responses from two successive transmissions of the prover. The basic idea is that the dishonest prover cannot maintain the consistent channel impulse responses at both the verifier and the helper with two different preambles $\mathbf{S_{p1}}$ and $\mathbf{S_{p2}}$, especially when the prover has no idea of the channel between the prover and the passive helper.

To facilitate the presentation, without loss of generality, we omit the noise part in the following equations. To generate a fake channel impulse response $\mathbf{h_a}$, the prover solves the fake preamble $\mathbf{S'_{p1}}$ from the equation $\mathbf{h} * \mathbf{S'_{p1}} = \mathbf{h_a} * \mathbf{S_{p1}}$, so that the verifier will regard $\mathbf{h_a}$ as the channel impulse response when uses $\mathbf{S_{p1}}$ to estimate the channel. The helper also uses $\mathbf{S_{p1}}$ to estimate its channel impulse response from $\mathbf{h_h} * \mathbf{S'_{p1}} = \mathbf{h_{ah}} * \mathbf{S_{p1}}$, where $\mathbf{h_h}$ and $\mathbf{h_{ah}}$ are the actual and the estimated channel impulse response between the helper and the prover. For the attacker's next transmission, both the verifier and the helper will use the preamble $\mathbf{S_{p2}}$ to estimate the channel. Similarly, to fool the verifier, the attacker must generate another fake preamble $\mathbf{S'_{p2}}$ such that it satisfies $\mathbf{h} * \mathbf{S'_{p2}} = \mathbf{h_a} * \mathbf{S_{p2}}$. However, the attacker can hardly fool the verifier and the helper at the same time. Because the attacker cannot know the channel between the attacker and the helper, the fake preamble $\mathbf{S'_{p2}}$ will not necessary satisfy the helper node's equation $\mathbf{h_h} * \mathbf{S'_{p2}} = \mathbf{h_{ah}} * \mathbf{S_{p2}}$. Thus, the channel estimation result at the helper will be different from the previous channel estimation result $\mathbf{h_{ah}}$.

If the successive estimated channel impulse responses show dramatic changes in a short time at the helper, the spoofing attacks are detected and the helper triggers an alert at the verifier. To avoid false alarms caused by normal channel fading, we can further increase the number of preambles. The verifier and the helper agree on n preambles ($n > 2$), and the alert will be triggered only when p out of n channel estimations are detected as inconsistent, where p is the threshold of the spoofing attack alerts.

4.4 Impact of a Cloned Prover

In an extreme scenario, a dishonest prover may use a collaborator at a different location to forge the proximity. The collaborator claims to be the prover and sends signals to the verifier. As a result, the verifier will take the collaborator's proximity as the prover's proximity. In this case, the verifier will send the data to a higher layer for authenticity verification. Such verification will fail if the prover does not disclose its personal information (e.g., user ID, network address, private key) to the collaborator. However, the verification does succeed if the collaborator has all the information of the prover, in which case the collaborator is essentially a full copy of the prover. Therefore, it is not practical to detect such a same-identity attack in any proximity detection system, including all distance bounding protocols. To defeat such attacks, the verifier needs to enforce existing security mechanisms like duplicated nodes detection (e.g., [25]) or hardware biometrics authentication (e.g., [6]), which are orthogonal to this work.

5 Experimental Evaluation

The proposed far proximity identification approach identifies whether a prover is at least a certain distance away from the verifier. To evaluate the performance of the proposed far proximity detection approach, two key questions are of particular interest. The first question is how likely it is that the detection method makes an error. An error happens when a prover is identified as at least β meters away, while the real distance d between the verifier and the prover is less than the identified lower bound β. The second question is how tight the estimated lower bound is. Let ϵ denote the difference between the real distance d and the lower bound β, i.e., $\epsilon = d - \beta$. Ideally, to obtain a good estimation accuracy, one would like to achieve a small ϵ. In this section, we perform experiments using real-world channel data to evaluate the performance of the proposed approach in a real wireless environment.

5.1 Experiment Setup

Wireless propagation can be either line-of-sight (LoS) or non-LoS (NLoS). In LoS scenarios, there exist no major or very few obstacles residing between the transmitter and receiver, and thus LoS scenarios usually feature better signal quality. In NLoS scenarios, there exist a number of major obstacles between the transmitter and receiver, and NLoS scenarios are more complicated with higher signal distortion and sharper changes in signal strength.

Far proximity identification often applies to long-haul wireless communications (e.g., GPS) in outdoor environments, which are usually open and have a much stronger feature in LoS than NLoS. Compared to outdoor environments, indoor environments like offices, residential homes, and shops, are more complicated due to the frequent occurrences of walls, people, furniture, cubicles, etc.

Thus, indoor environments usually have a fairly large number of NLoS propagation paths. In our experiment, we choose the more challenging indoor environment for our evaluation to examine the worst-case performance of the proposed method.

5.1.1 Data Set

We validate the proposed far proximity identification technique using the CRAWDAD data set [26], which contains more than 9,300 real channel impulse response measurements (i.e., link signatures) in a 44-node wireless network [36]. There are $44 \times 43 = 1,892$ pairwise links between the nodes, and multiple measurements are provided for each link [36]. The map of the 44 node locations is shown in [27]. The measurement environment is an indoor environment with obstacles (e.g., cubicle offices and furniture) and scatters (e.g., windows and doors). More information regarding the CRAWDAD data set can be found in [26], [36].

5.1.2 Evaluation Metrics

We herein use *error rate* and *tightness of the bound* as metrics to evaluate the performance of the proposed technique in the real world. In addition, the proximity lower bound is computed based on a key factor, the proximity fingerprint. Thus, the proximity fingerprints plays a vital role in proximity identification. To further validate the the feasibility of using proximity fingerprints for proximity identification, we also perform experiments to reveal the relationship between the real distance and the proximity fingerprints. Our evaluation metrics are summarized below.

- **Error rate**: The error rate is the ratio of the number of failed trials (i.e., error happens in the trail) to the total number of trials. An error happens when the real distance between the verifier and the prover is less than the identified proximity lower bound.
  The error rate indicates how possible a nearby adversary will be considered as a remote legitimate device. A small error rate indicates a nearby adversary can hardly pretend to be far away from the prover, and vice versa.
- **Tightness of the bound**: Tightness is the normalized difference between the estimated lower bound and the real distance (i.e., $\frac{d-\beta}{d}$, where β is the estimated proximity lower bound, and d is the real distance between the verifier and the prover).
  Tightness indicates how close between the estimated result and the real distance. A small tightness indicates a remote device will not be falsely considered as a nearby adversary, and vice versa.
- **Proximity Fingerprints**: The proximity fingerprint is the ratio of the amplitude of the first received multipath component to that of the second one.

5.2 Experiment Results

Based on the CRAWDAD data set, we perform experiments under both LoS and NLoS scenarios to show the error rate, tightness of the bound, and the relationship between the proximity fingerprint and the distance.

We distinguish two types of channel impulse responses: if a LoS path exists and there are no obstacles between the transmitter and the receiver, we mark the corresponding channel impulse responses as LoS channel impulse responses. Otherwise, we mark them as NLoS channel impulse responses. Thus, we obtain two sets of data. The first set is formed by all LoS channel impulse responses, and the second one is formed by all NLoS channel impulse responses. We perform our experiments using both sets.

5.2.1 Proximity fingerprint vs. distance

The proximity fingerprint is an important parameter in computing the proximity lower bound. According to Lemma 4, the theoretical proximity lower bound is calculated as $\frac{c}{B(f^{\frac{2}{\alpha}}-1)}$. From this formula, we can easily derive that as the proximity fingerprint f increases (other parameters remain the same), the proximity lower bound decreases and vice versa. Note that the proximity lower bound reveals the least distance between the verifier and the prover. Thus, the increase of the proximity fingerprint f may also indicate the decrease of the real distance and vice versa. We plot the proximity fingerprint as a function of the distance in Figure 8. We can see that the proximity fingerprint in the NLoS scenario slightly differs from that of the LoS scenario in magnitude due to the reflection loss. However, for both scenarios, their proximity fingerprints exhibit the same tendency, i.e., they both decrease as the distance increases. This observation is consistent with our theoretical result.

5.2.2 Error Rate

Error rate vs. pathloss: To obtain the error rate, we experiment as follows. Let N_{LoS} denote the number of channel impulse responses in the LoS data set. For each channel impulse response in the data set, we compute the proximity fingerprint and the corresponding proximity lower bound using Lemma 4. We also compute the real distance between the transmitter and the receiver based on their coordinates. If the lower bound is less than the real distance, we mark the trial as successful. Otherwise, we mark the trial as failed. Accordingly, the error rate is calculated as $\frac{N_f}{N_{LoS}}$, where N_f is the number of failed trails and N_{LoS} is the total number of trials. We perform the experiment again using the NLoS data set and obtain the corresponding error rate for the NLoS scenario.

The channel impulse responses are collected from an indoor environment, and the corresponding pathloss exponent α empirically ranges between 1.6 and 1.8. Thus, we perform our experiment for different values of α in this range. Figure 9 plots the error rate as a function of α. The pathloss exponent α reflects how a signal is distorted and attenuated during its propagation, and a large α can result in higher signal distortion and attenuation. Accordingly, from Figure 9 we can observe that the error rate increases as α increases. However, when α reaches the maximum value for indoor environments, the achieved error rate in the LoS scenario is as low as 0.075. For the minimum α of 1.6, the proposed approach has a reduced error rate of 0.05.

For the NLoS scenario, we can still achieve an error rate between 0.17 and 0.22. Note that NLoS scenarios are the worst-case scenarios. Far proximity identification is typically used in outdoor environments, which have the stronger LoS feature. As shown in Figure 9, the error rate of LoS scenarios is much lower than that of the NLoS scenarios.

Error rate vs. distance: We then perform experiments to examine how the real distance affects the error rate. For each channel impulse response in the LoS data set, we compute the distance between the corresponding transmitter and the receiver. Let d_{max} and d_{min} denote the maximum and minimum distance among all computed distances. We calculate the error rate using the set formed by channel impulse responses whose corresponding

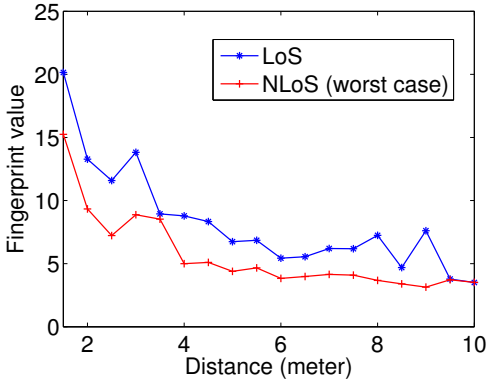


Fig. 8. Relationship between the distance and the proximity fingerprint.

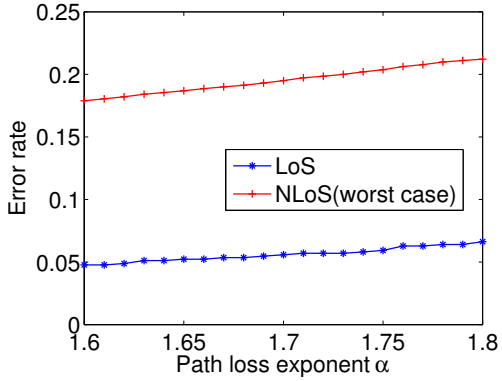


Fig. 9. Error rate as a function of pathloss exponent α.

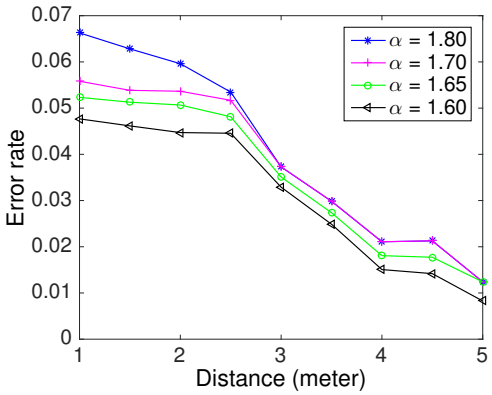


Fig. 10. Error rate as a function of various distances in the LoS scenario

distance are larger than a threshold distance. The threshold is initially set as d_{min} and increases each time until it reaches d_{max}. We perform the experiments again using the NLoS set.

Figure 10 shows the error rate as a a function of various distances in the LoS scenario. The error rate decreases as distance increases. The obvious reason is that a larger distance indicates a longer distance between the transmitter and the receiver, and thus a higher chance that the estimated proximity lower bound is less than the distance. When distance approaches the maximum distance between the sender and the receiver, the corresponding error rate is 0.01. When distance approaches the minimum distance, the error rate slightly increases but it is still a small rate that ranges between 0.05 and 0.07 for different α.

Figure 11 plots the error rate of the NLoS scenario for $\alpha = 1.80$, which results the worst error rate as compared to other values of α. Contrary to the LoS scenario, the error rate of the NLoS scenario increases as distance increases. That's because in the NLoS scenario a longer distance between the transmitter and the receiver indicates a higher chance that there are more obstacles, and thus a reduced proximity detection accuracy. The "worst worst case" happens when distance approaches the maximum distance d_{max} for the worst case NLoS scenario. However, as we can observe from Figure 11, the achieved error rate of the "worst worst case" is about 0.25. This means that we can successfully obtain the proximity lower bound for a majority number (75%) of verifiers. As distance decreases, the error rate decreases quickly. When distance approaches the minimum distance, the achieved error rate is about 0.15. Again, the experiment is performed in an indoor environment (e.g., WiFi and Bluetooth), which has a short signal propagation distance. Outdoor wireless applications (e.g., space communications and TV broadcasting) usually have the stronger LoS feature, and therefore can substantially benefit from the proposed method in terms of significantly reducing the error rate.

5.2.3 *Tightness of the proximity bound*

Our second evaluation metric is the tightness of the bound. To evaluate the tightness, we perform the following experiments using LoS and NLoS data sets. In all experiments, the pathloss exponent α is set to the minimum and maximum values of 1.6 and 1.8. For each channel impulse response in the LoS data set, we compute the distance between the corresponding transmitter and the receiver and the proximity lower bound. Based on the bound and the actual distance, we can calculate the tightness of the bound. We then sort all the tightness values and compute the empirical cumulative distribution function (CDF) for them. We perform the experiment again using NLoS data set and obtain the CDFs of the NLoS tightness values.

Figure 12 shows the CDF curves of the tightness computed using channel impulse responses collected in LoS and NLoS scenarios. For the LoS scenario with $\alpha = 1.8$, we can observe that 95% of the tightness values are less than 0.2. The indoor environment typically features a short propagation path, and thus a 0.2 tightness indicates a small absolute difference in distance. For example, if the distance between the transmitter and receiver is 5 meters, the achieved tightness can be around 1 meter. In particular, the maximum distance d_{max} between the transmitter and the receiver is about 11 meters, and the corresponding proximity bound is 9.56 meters, which is very close to the actual distance.

For the NLoS scenario with $\alpha = 1.8$, we can observe from Figure 12 that 90% of the tightness values are less than 0.3. Compared to the LoS Scenario, the NLoS scenario has a reduced performance due to the existence of obstacles. Again, the experiment is conducted based on short-range communications, and a 0.3 tightness still suggests a small absolute difference in distance. When α decreases to 1.6, the achieved tightness increases. That's because the corresponding estimated proximity lower bound decreases, and a decreased bound grows the difference between the bound and the real distance, and thus augments the tightness. However, for $\alpha = 1.6$, we can still observe that a great majority of the tightness values are fairly small, e.g., 95% and 80% of the tightness values are less than 0.25 and 0.3 in the LoS and the NLoS (worst-case) scenarios respectively. Note that such tightness

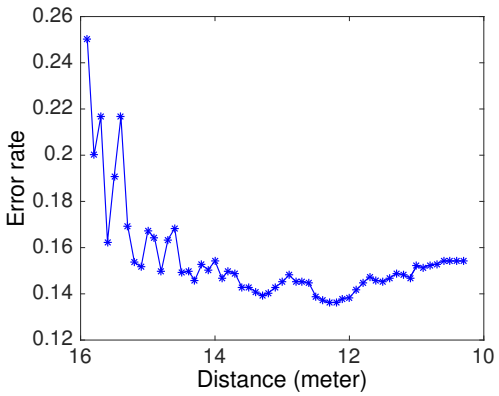


Fig. 11. Error rate as a function of various distances in the NLoS scenario

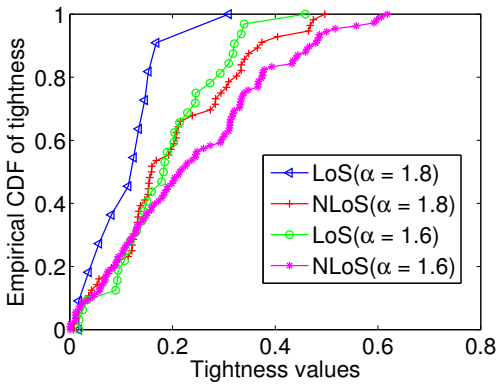


Fig. 12. The empirical CDF curves of the tightness.

is usually sufficient to prevent attackers from impersonating the transmitters in typical long-haul outdoor wireless applications. For example, GPS satellites running on the Low Earth Orbit have an altitude of approximately 2,000,000 meters (1,200 miles). With a proximity lower bound of 1,000,000 meters (i.e., a tightness of 0.5), it would be possible to prevent most attackers from impersonating the satellites, because it is usually very difficult for the attacker to achieve such a long transmission range.

5.2.4 *The experiment for a longer distance scenario*

We further conduct an experiment to evaluate the performance of the proposed scheme in the scenario of a longer distance between the prover and the verifier on top of the Universal Software Radio Peripherals (USRPs), which are the radio frequency transceivers.

In the experiment, the transmit rate is set as 10 Mbps and the distance between the prover and the verifier is set as 50 meters, which are the maximum rate and largest distance we can achieve due to the hardware limitations of USRPs (i.e. processing capability and transmission power). The experiment is done in the outdoor scenario, and the corresponding path loss exponent α is chosen as 5.0. We measure the channel impulse response for 1000 times, and for each measurement we estimate the corresponding lower bound proximity. Note that during the measurements, the environment may change since there are people walking between the transmitter and receiver.

From the experiment results, we can observe an error rate of 0.0939. We also draw the empirical CDF of tightness in Figure 13. As shown in this figure, the maximum tightness value is 0.3385. As discussed earlier, such tightness is usually sufficient to prevent attackers from impersonating a nearby transmitters in typical long-range outdoor wireless applications.

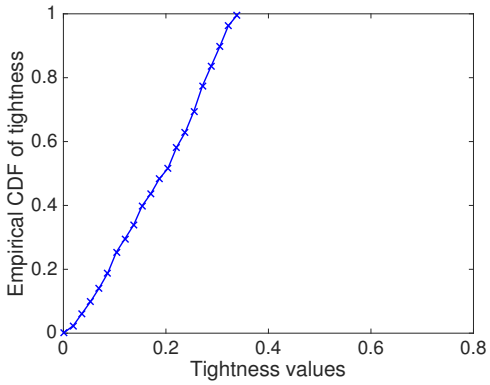


Fig. 13. Empirical CDF of tightness with a distance of 50 meters

6 Related Work

Related work falls into the following two areas.

(a) Distance Bounding Protocols: Distance bounding protocols are a class of protocols that determine an approximate distance between a local device and a remote device. (e.g., [5], [32], [38]). Distance bounding protocols and their variants are based on the common observation that the distance between the local and the remote devices is equal to the product of the speed of electromagnetic wave and the one-way signal propagation time. The approximate distance is obtained from a series of wireless packets exchanged between the local device and the remote device. Specifically, the local device sends a challenge to the remote device, which then replies with a response that is generated based on the challenge. The local device measures the round-trip time between sending the challenge and receiving the response, subtracts the processing delay from the round-trip time, and uses the result to compute the distance. Because the response is generated based on the challenge, the distance bounding protocol can prevent the remote device from pretending to be closer than it actually is by sending a fake response before it receives the challenge.

However, by delaying its response to a challenge, a remote device can appear to be arbitrarily further from the local device than it actually is. Hence, distance-bounding protocols cannot enforce lower bounds on proximity (i.e., requirements that the remote device be *at least* a certain distance from the local device). For this reason, the GPS-device and mobile-phone examples used for motivation in Section 1 cannot be enforced by distance-bounding protocols.

(b) Close Proximity Identification: There also exist traditional close proximity detection techniques (e.g., [8], [19]) that can detect the presence of nearby objects without any physical contact. These techniques use electromagnetic field changes to identify a close object. A proximity sensor generates an electromagnetic field or a beam of electromagnetic radiation (e.g., infrared). If an object moves into the field range of the sensor, a field change can result, and thus the sensor senses the presence of the object. For example, a sound alert is triggered when a vehicle moves into the close proximity of a worker or an obstacle. However, traditional techniques cannot identify the proximity of a specific object, because the proximity sensor reports all nearby objects as

long as those objects are in the field range.

Researchers later developed techniques that identify the close proximity of an individual target if the target can emit wireless signals (e.g., [7], [13], [20]). For example, based on the observation that a strong received signal usually indicates a close transmitter, Macii et al developed approaches that determine the proximity of the remote wireless device by measuring received signal strength [20]. However, the use of signal strength to determine proximity was found to be insecure, as a dishonest remote device can easily pretend to be close to the local device by boosting its transmit power.

More recent efforts overcome this drawback with the assistance of special hardware [7], [13]. Cai et al. proposed a scheme that identifies the presence of a close wireless device by using multiple antennas [7]. Halevi et al. proposed to use ambient sensors to detect whether a Near-Field-Communication (NFC) device is nearby or not [13]. Although those approaches can prevent attackers manipulating transmit power to deceive the local device, they cannot be directly extended to address the far proximity identification problem. They output a decision regarding whether a target is nearby, but such a decision cannot guarantee that the target is at least a certain distance away. Also, the requirement of special hardware such as multiple antennas and ambient sensors introduces extra cost and may reduce their compatibility.

Liu et al. proposed a new close proximity identification approach that does not rely on special hardware [18]. By using the wireless physical features that uniquely identify a wireless link between a transmitter and a receiver, the proposed technique enables the local device to distinguish between a nearby and a far-away remote device. An attacker cannot manipulate such physical features to pretend to be close to the local device. However, similar to all previous approaches, this approach is a decision-based, i.e. outputs a simple "yes" or "no" to indicate whether the remote device is very close or not. Hence, it does not provide the quantitative lower bound of the proximity, which is the primary contribution of this paper.

(c) CSI Based Distance Tracking Scheme: Existing CSI distance tracking ideas mainly focus on providing accurate distance estimation schemes. For example, Sen et al. proposed a distance estimation scheme that can extract the signal strength and the angle of only the direct path utilizing the channel state information, and thus provide an accurate estimation result in WiFi based localization systems [35]. On the other hand, the proposed scheme aims to estimate the lower bound of the proximity and focus on preventing a nearby adversary from impersonating a remote legitimate device. Existing CSI distance tracking schemes (e.g. [35]) may be vulnerable to such attacks, since attackers can pretend to be a further distance away from the receiver if they reduce the transmit power. In this way, the proposed scheme is complementary to existing schemes to provide an accurate and secure distance estimation scheme.

7 Conclusion

In this paper, we proposed a far proximity identification approach that determines the lower bound of the distance between the verifier and the prover. The key idea of the proposed approach is to estimate the proximity lower bound from the unforgeable fingerprint of the proximity. We developed a technique that can extract the fingerprint of a wireless device's proximity from the channel impulse response of the signals sent by the device. We also developed a technique that uses proximity fingerprint to calculate the proximity lower bound. We have examined the proposed approach through the real-world experimental evaluation using the CRAWDAD data set [26]. Our results indicate that the proposed approach is a promising solution for enforcing far proximity policies in wireless systems.

References

[1] Gps signals. http://en.wikipedia.org/wiki/GPS_signals. [Online; accessed 27-July-2013].

[2] Marine vhf radio. http://en.wikipedia.org/wiki/Marine_VHF_radio. [Online; accessed 13-July-2013].

[3] N. Alam, A. T. Balaie, and A. G. Dempster. Dynamic path loss exponent and distance estimation in a vehicular network using doppler effect and received signal strength. In *Proceedings of 2010 Vehicular Technology Conference Fall (VTC 2010-Fall)*, pages 1–5, 2010.

[4] M. Biguesh and A. B. Gershman. Training-based mimo channel estimation: A study of estimator tradeoffs and optimal training signals. *IEEE Transaction on Signal Processing*, 54(3):884–893, March 2006.

[5] S. Brands and D. Chaum. Distance bounding protocols. In *Proceedings of EUROCRYPT*, pages 344–359, 1994.

[6] V. Brik, S. Banerjee, M. Gruteser, and S. Oh. Wireless device identification with radiometric signatures. In *Proceedings of the 14th ACM International Conference on Mobile Computing and Networking (MobiCom '08)*, pages 116–127, 2008.

[7] L. Cai, K. Zeng, H. Chen, and P. Mohapatra. Good neighbor: Ad hoc pairing of nearby wireless devices by multiple antennas. In *Proceedings of the Annual Network and Distributed System Security Symposium (NDSS 2011)*, 2011.

[8] Z. Chen and R.C. Luo. Design and implementation of capacitive proximity sensor using microelectromechanical systems technology. *IEEE Transactions on Industrial Electronics*, 45(6):886–894, 1998.

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

[10] A. Goldsmith. *Wireless Communications*. Cambridge University Press, New York, NY, USA, 2005.

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

[12] K. Gunnam, G. Choi, M. Yeary, and Y. Zhai. A low-power preamble detection methodology for packet based rf modems on all-digital sensor front-ends. In *In Proceedings of the IEEE Instrumentation and Measurement Technology Conference*, 2007.

[13] T. Halevi, D. Ma, N. Saxena, and T. Xiang. Secure proximity detection for nfc devices based on ambient sensor data. In *Proceedings of the European Symposium on Research in Computer Security (ESORICS 2012)*, 2012.

[14] G. P. Hancke and M. G. Kuhn. An RFID distance bounding protocol. In *Proceedings of SecureComm'05*, pages 67–73, 2005.

[15] C. Jinho. Equalization and semi-blind channel estimation for space-time block coded signals over a frequency-selective fading channel. *IEEE Transactions on Signal Processing*, 52(3):774 – 785, 2004.

[16] L. B. Kuechle. Selecting receiving antennas for radio tracking. http://www.atstrack.com/PDFFiles/receiverantrev6.pdf.

[17] A. Liu, P. Ning, H. Dai, Y. Liu, and C. Wang. Defending DSSS-based broadcast communication against insider jammers via delayed seed-disclosure. In *Proceedings of the 26th Annual Computer Security Applications Conference ACSAC '10*, December 2010.

[18] Y. Liu, P. Ning, and H. Dai. Authenticating primary users' signals in cognitive radio networks via integrated cryptographic and wireless link signatures. In *Proceedings of 2010 IEEE Symposium on Security and Privacy (S&P '10)*, pages 286–301, May 2010.

[19] P. H. Lo, C. Hong, S. C. Lo, and W. Fang. Implementation of inductive proximity sensor using nanoporous anodic aluminum oxide layer. In *Proceedings of 2011 International Solid-State Sensors, Actuators and Microsystems Conference (TRANSDUCERS)*, pages 1871–1874, 2011.

[20] D. Macii, F. Trenti, and P. Pivato. A robust wireless proximity detection technique based on rss and tof measurements. In *Proceedings of 2011 IEEE International Workshop on Measurements and Networking (M&N'11)*, pages 31–36, 2011.

[21] A. Mahmood, R. Exel, H. Trsek, and T. Sauter. Clock synchronization over ieee 802.11—a survey of methodologies and protocols. *IEEE Transactions on Industrial Informatics*, 13(2):907–922, April 2017.

[22] G. Mao, B. D. O. Anderson, and B. Fidan. Path loss exponent estimation for wireless sensor network localization. *The International Journal of Computer and Telecommunications Networking*, 51(10):2467–2483, 2007.

[23] A. F. Molisch. *Wireless Communications, 2nd Edition*. Wiley India Pvt. Limited, 2007.

[24] C. Paget. Practical cellphone spying. *DEF CON 18*, 2010.

[25] B. Parno, A. Perrig, and V. Gligor. Distributed detection of node replication attacks in sensor networks. In *Proceedings of the IEEE Symposium on Security and Privacy (oakland'05)*, pages 49–63, 2005.

[26] N. Patwari and S. K. Kasera. CRAWDAD utah CIR measurements. http://crawdad.cs.dartmouth.edu/meta.php?name=utah/CIR.

[27] N. Patwari and S. K. Kasera. Robust location distinction using temporal link signatures. In *MobiCom '07: Proceedings of the 13th annual ACM international conference on Mobile computing and networking*, pages 111–122, New York, NY, USA, 2007. ACM.

[28] N. Patwari and S. K. Kasera. Temporal link signature measurements for location distinction. *IEEE Transactions on Mobile Computing*, 10(3):449–462, March 2011.

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

[30] Pöpper, M. Strasser, and S. Čapkun. Anti-jamming broadcast communication using uncoordinated spread spectrum techniques. *IEEE Journal on Selected Areas in Communications: Special Issue on Mission Critical Networking*, 2010.

[31] C. Pöpper, M. Strasser, and S. Čapkun. Jamming-resistant broadcast communication without shared keys. Technical report, ETH Zurich, September 2008. ETH Zurich D-INFK Technical Report 609.

[32] K. B. Rasmussen and S. Čapkun. Realization of rf distance bounding. In *Proceedings of the USENIX Security Symposium*, 2010.

[33] K.B. Rasmussen, C. Castelluccia, T.S. Heydt-Benjamin, and S. Čapkun. Proximity-based access control for implantable medical devices. In *Proceedings of the 16th ACM Conference on Computer and Communications Security (CCS '09)*, 2009.

[34] Robert A. Scholtz. *Spread Spectrum Communications Handbook*. McGraw-Hill, 2001.

[35] Souvik Sen, Jeongkeun Lee, Kyu-Han Kim, and Paul Congdon. Avoiding multipath to revive inbuilding wifi localization. In *Proceeding of the 11th annual international conference on Mobile systems, applications, and services*, pages 249–262. ACM, 2013.

[36] SPAN. Measured channel impulse response data set. http://span.ece.utah.edu/pmwiki/pmwiki.php?n=Main.MeasuredCIRDataSet.

[37] S. Sud. A low complexity spatial rake receiver using main beam multipath combining for a cdma smart antenna system. In *Proceedings of 2007 IEEE Military Communications Conference (MILCOM)*, pages 1–6, 2007.

[38] N. O. Tippenhauer and S. Čapkun. Id-based secure distance bounding and localization. In *Proceedings of 2009 European Symposium on Research in Computer Security(ESORICS'09)*, 2009.

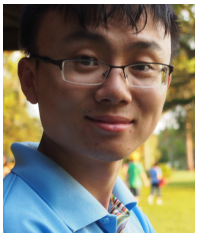
[39] M. K. Tsatsanis and G. B. Giannakis. Blind estimation of direct sequence spread spectrum signals in multipath. *IEEE Transactions on Signal Processing*, 5(45):1241 – 1252, 1997.

[40] R. Weinmann. The baseband apocalypse. *BlackHat DC*, 2011.

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

[42] L. Yu, W. Liu, and R. J. Langley. Robust beamforming methods for multipath signal reception. *Digital Signal Processing*, 20(2):379–390, 2007.

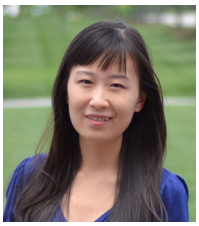
[43] J. Zhang, M. H. Firooz, N. Patwari, and S. K. Kasera. Advancing wireless link signatures for location distinction. In *MobiCom '08: Proceedings of the 14th ACM international conference on Mobile computing and networking*, New York, NY, USA, 2008. ACM.



Tao Wang is currently a third-year Ph.D. student in the Department of Computer Science and Engineering, University of South Florida, Tampa, FL. His research is related to wireless network, mobile security and cyber-physical system security. Currently, his research mostly focuses on securing the wireless communication by exploring the physical-layer features of the wireless channel.

Jian Weng received the M.S. and B.S. degrees in computer science and engineering from South China University of Technology, in 2004 and 2000, respectively, and the Ph.D. degree in computer science and engineering from Shanghai Jiao Tong University, in 2008. From April 2008 to March 2010, he was a postdoc in the School of Information Systems, Singapore Management University. Currently, he is a professor and vice dean with the School of Information Technology, Jinan University. He has published more than 40 papers in cryptography conferences and journals, such as PKC, CT-RSA, ACSAC, SCN, Designs, Codes and Cryptography, Algorithmica, etc. He served as PC cochairs or PC member for more than 10 international conferences, such as ISPEC 2011, RFIDsec 2013 Asia, ISC 2011, IWSEC 2012, etc.

Jay Ligatti received the B.S. (Hons.) degree in computer science and the B.M. degree in music composition from the University of South Carolina, Columbia, SC, USA, in 2001, and the M.S. and Ph.D. degrees in computer science from Princeton University, Princeton, NJ, USA, in 2003 and 2006, respectively.He was a Software Engineer and Security Consultant with Medical Software and Computer Systems, Richmond, VA, USA, in 2000. He was also a Research Intern with Microsoft, Redmond, WA, USA, in 2003, where he co-created and proved soundness of control-flow mechanisms, and a Consultant on software security at CACI, Arlington, VA, USA, in 2012. He is currently an Associate Professor of Computer Science and Engineering with the University of South Florida, Tampa, FL, USA. His current research interests include software security and programming languages, runtime monitoring, enforcement models, policy-specification languages, code-injection attacks, firewalls and packet-classification algorithms, type systems, and tools for building and managing complex security policies.



Yao Liu received the Ph.D. degree in Computer Science from North Carolina State Univ. in 2012. She is now an assistant professor at the Dept. of Computer Science and Engineering, Univ. of South Florida, Tampa, FL. Dr. Liu's 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 interest also lies in the security of cyber-physical systems, especially in smart grid security. Dr. Liu's research work has appeared in premier journals and conferences including ACM Transactions on Information and Systems Security, IEEE Symposium on Security and Privacy (IEEE S&P), ACM Conference on Computer and Communications Security (CCS), and IEEE International Conference on Computer Communications (INFOCOM). She was the recipient of Best Paper Award for the 7th IEEE International Conference on Mobile Ad-hoc and Sensor Systems.