Phantom-CSI Attacks against Wireless Liveness Detection

Qiuye He
qiuye.he@ou.edu
The University of Oklahoma
Norman, OK, USA

Song Fang
songf@ou.edu
The University of Oklahoma
Norman, OK, USA

ABSTRACT

All systems monitoring human behavior in real time are, by their nature, attractive targets for spoofing. For example, misdirecting live-feed security cameras or voice-controllable Internet-of-Things (IoT) systems (e.g., Amazon Alexa and Google Assistant) has immediately intuitive benefits, so there is a consequent need for detecting liveness of the human(s) whose behavior is being monitored. Emerging research lines have focused on analyzing changes in prevalent wireless signals to detect video or voice spoofing attacks, as wireless-based techniques do not require the user to carry any additional device or sensor for liveness detection. Video/voice streaming and coexisting wireless signals convey different aspects of the same overall contextual information related to human activities, and the presence of spoofing attacks on the former breaks this relationship, so the latter performs well as liveness detection to augment the former. However, we recognize and herein evaluate how to spoof the latter as well to defeat this liveness detection. In our attack, an adversary can easily create phantom wireless signals and synchronize them with spoofed video/voice signals, such that the legitimate user can no longer distinguish real from fake human activity. Real-world experimental results on top of software-defined radio platforms validate the possibility of generating fake CSI flows and demonstrate that with the phantom-CSI attack, the true positive rates (TPRs) of wireless liveness detection systems for video and voice decrease from 100% spoofing detection to just 4.4% and 0, respectively.

CCS CONCEPTS

• **Security and privacy → Mobile and wireless security**.

KEYWORDS

liveness detection, spoofing attacks, CSI, human motion

ACM Reference Format:
Qiuye He and Song Fang. 2023. Phantom-CSI Attacks against Wireless Liveness Detection. In *The 26th International Symposium on Research in Attacks, Intrusions and Defenses (RAID '23), October 16–18, 2023, Hong Kong, Hong Kong*. ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3607199.3607245

1 INTRODUCTION

Liveness detection using wireless signals aims to detect whether human activity is real (from a live person present at the point of capture) or fake (from a spoof artifact or lifeless body part) by exploring the correlation between feeds of a sensor capturing human motion and co-existing wireless signals. Wireless liveness detection has proven successful in securing various practical systems [23, 24, 29, 36, 42], such as

- **Video liveness detection:** By launching a video spoofing attack (e.g., [4]), an adversary can hijack the camera feed to replay benign footage while stealing valuables (e.g., contents of a vault) without getting caught. A security guard can detect such attacks by observing mismatches between the live video feeds and the captured wireless signals [29].
- **Voice liveness detection:** Voice controllable systems are especially vulnerable to spoofing attacks (e.g., with pre-recorded voice [12]) due to the inherent broadcast nature of voice transmissions. It can tell whether the voice command is generated by a live user via comparing the features extracted from both voice and wireless signals [36].
- **Human presence detection:** Wireless signals can be utilized to detect human presence by human breathing [32, 57, 64]. Wireless liveness detection can thus associate the detection of breathing with the user presence to combat replay attacks against voice assistants [42].

Human activity usually causes subtle environmental impacts unique to that human activity pattern, which can be observed by analyzing collected nearby wireless signals. As a result, wireless signals can be utilized to detect human activities and thus verify the authenticity of the captured data of another co-existing sensor such as video or microphone.

Mainstream WiFi systems are based on the Orthogonal frequency-division multiplexing (OFDM) technique, which utilizes multiple parallel narrowband subcarriers to encode a packet. Disturbances in wireless signals can be quantified by the *channel state information (CSI)* measurement [17], which describes how the wireless channel impacts the radio signal that propagates through the channel (e.g., amplitude attenuation and phase shift). CSI can be considered as an aptly initialed wireless analog to traditional "Crime Scene Investigation", measuring what has happened on a wireless channel [18]. Specifically, the variation of CSI time series has been widely utilized to identify the motion changes of a target user between a wireless transmitter and receiver pair.

In this work, however, we design a new *phantom-CSI* attack against all existing liveness detection built on the correlation between recorded human activity and co-existing CSI measurements. This attack accompanies traditional spoofing of video or microphone recorders by creating measurable CSI which exhibits corresponding spoofed human activity, bypassing the enforced wireless liveness detection system.

To understand the phantom-CSI attack, we first explain the impact of human activity on wireless signals. Generally, the presence

RAID '23, October 16–18, 2023, Hong Kong, Hong Kong

ACM ISBN 979-8-4007-0765-0/23/10.
https://doi.org/10.1145/3607199.3607245

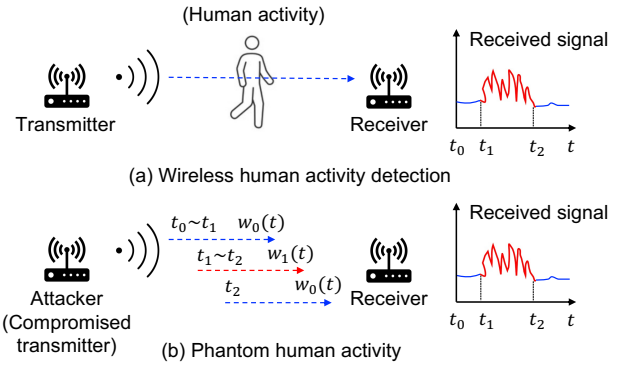


Figure 1: Crafting wireless signal affected by human activity.

of human and related body motion will result in significant changes in both amplitude and phase of the received wireless signals [31]. Accordingly, the received wireless signal (or CSI) at the receiver can thus capture the timing information (e.g., start or end time) and prominent frequency of occurrence of activities [29], and will exhibit a unique pattern corresponding to each activity [36]. For example, the repetitive (rhythmic) patterns of human breathing induce wave-like (sinusoidal-like) periodic change patterns over time in the CSI amplitudes at subcarrier level [32, 57, 64]. To fool a receiver to believe that an event occurs, the attacker needs to create a "virtual channel" that can exhibit a pattern similar to the real wireless channel affected by the event.

Figure 1 presents an example at the OFDM subcarrier level to illustrate how the attacker can build such a channel. Figure 1a shows a real scenario without an attack, where the transmitter sends a wireless signal and a human activity (e.g., walking) occurs between the transmitter and the receiver during the period from time t_1 to t_2. As a result, the received signal at the receiver would reflect the corresponding interference during the activity period $[t_1, t_2]$. Figure 1b shows an attack scenario, where there is no human activity happening between the attacker (i.e., a compromised transmitter) and the receiver, but the attacker aims to make the receiver detect some activities similar to that in Figure 1a. For each transmitted signal at time t, the attacker multiplies it with a corresponding coefficient, i.e., $w_0(t)$ when $t \in [t_0, t_1)$ or $t > t_2$, or $w_1(t)$ when $t \in [t_1, t_2]$, to mimic the distortion effect of the real subchannel in Figure 1a. Consequently, the receiver observes a distinguishable time series in period $[t_1, t_2]$ and incorrectly deduces that it is caused by the activity performed in Figure 1a.

Beyond this example of spoofing human activity in its absence, an attacker may have other goals, such as obscuring a particular human activity or portraying a different fake activity. Performing this general attack requires two technical solutions. First, the phantom motion must be encoded in the form of CSI for the receiver to estimate and map to the intended motion. Accordingly, we design a custom technique to convert an event into manipulated CSI of a wireless channel. Second, the transmitted signal crafted by the adversary is affected by the real wireless channel between herself and the receiver. Thus, the attacker requires a method to cancel the effect of the real channel, so that the receiver only observes the phantom channel corresponding to spoofed activity. We address this challenge by reverse-engineering existing channel estimation algorithms for OFDM systems and pre-coding the original signal.

The discovered attack reveals that an attacker can create fake CSI data corresponding to spoofed voice or video signals. We conduct real-world experimental evaluations on top of Universal Software Radio Peripheral (USRP) X300 platforms. The experimental results show that an attacker camouflaged via our phantom CSI can inject spoofed video and voice to successfully bypass wireless liveness detection systems with a probability of 95.6% and 100%. We summarize our main contributions as follows.

- This paper is the first to point out the vulnerability of wireless liveness detection systems, via phantom-CSI attacks causing wireless signals and spoofed video/voice data to present common yet fake human semantic information.
- We create a technique that can successfully craft fake CSI based on human activities and deliver it to the receiver via a realistic wireless channel.
- We implement real-world prototypes of both existing wireless video/voice liveness detection and the proposed attack techniques, validating the efficacy of the latter against the former.

2 PRELIMINARIES

In this section, we introduce the prevalent algorithm used to estimate CSI for OFDM and the general method used by existing work employing CSI to achieve liveness detection.

2.1 CSI Estimation

As discussed earlier, the occurrence of human activities can induce disturbances in the surrounding wireless signal and thus variation in the observed CSI at the receiver.

The OFDM technique has been widely used in modern wireless communication systems, e.g., 802.11 a/g/n/ac/ad. The *channel frequency responses* measured from all subcarriers form the CSI of OFDM. Let $H(f, t)$ denote the channel frequency response at time t for a particular subcarrier with a frequency f. It is usually estimated by using a pseudo-noise sequence that is publicly known [15, 17, 67]. Specifically, a transmitter sends a pseudo-noise sequence, denoted with $X(f, t)$, over the wireless channel, and the receiver estimates $H(f, t)$ from $X(f, t)$ and the received, distorted copy, denoted with $Y(f, t)$, i.e., $H(f, t) = \frac{Y(f,t)}{X(f,t)}$.

2.2 CSI-aided Liveness Detection

A myriad of recent studies have shown success of using CSI to recognize subtle human movements, including walking [60, 62], falling [40], breathing [32], mouth movements [56], and activities of daily living [44]. Existing CSI-based liveness detection techniques discover that CSI from widely available wireless signals is able to perceive human existence or activities in the place of interest in addition to surveillance cameras [24, 29] or a microphone [36, 42], and thus spoofing attacks can be detected by catching dissimilarities between CSI and video/voice signals.

These techniques normally use four steps to verify live users and detect spoofing attacks, namely, data synchronization, data pre-processing, feature extraction, and consistency checking. The

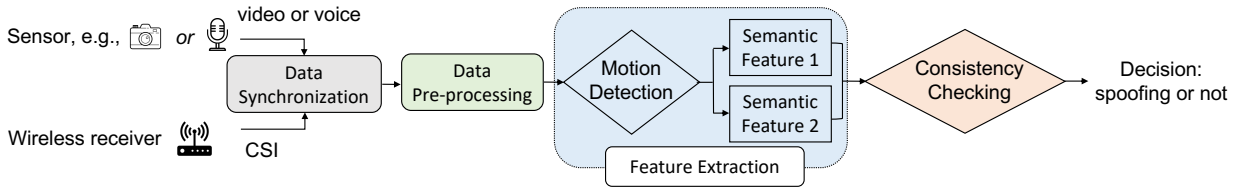


Figure 2: General structure of a wireless liveness detection system.

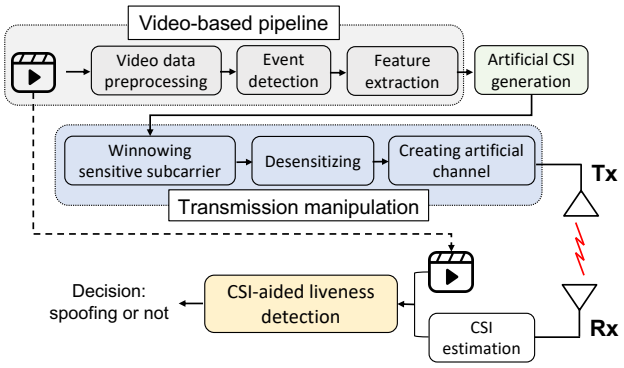


Figure 3: Flow chart of the phantom-CSI attack.

first phase synchronizes signals in both modalities. The following phase pretreats video/voice feeds for activity detection and removes noise from the CSI. Next, specific features are extracted from both CSI and video/voice signals. They are then correlated and exploited for deciding whether a spoofing attack happens in the final phase. Figure 2 illustrates a general flowchart of the CSI-aided liveness detection system.

3 ADVERSARY MODEL

A general wireless liveness detection system utilizes wireless signals as a second-factor authentication for human activity, which is detected via another co-existing sensor. Without loss of generality, we consider a common surveillance scenario, where a camera is used to monitor an open area, and a transmitter and receiver pair is utilized to verify the authenticity of the video captured by the camera. Specifically, the public transmitter constantly transmits the wireless signal; the receiver estimates the CSI based on the received signal. We point out that such a public transmitter can be unreliable and can be exploited for launching the proposed attack. If the detected human activities from wireless signals and the camera match with each other, the video is authentic, otherwise the video spoofing attack is detected.

To demonstrate the impact of our attack, we consider an attacker who can craft a fake video and feed it to the camera (e.g., [4, 20]). This aligns with existing liveness detection studies (e.g., [23, 24, 26, 29]). The attacker aims to make the target system unable to detect the fake video. She may use the public transmitter as an accomplice. Alternatively, if the defender secures the public transmitter, the attacker can set up a hidden transmitter nearby. Similar to other wireless attacks such as GPS spoofing [52], the attacker's transmitter then employs wireless jamming or spoofing techniques [70] to cancel the real signals and let the receiver take the fake signals from the attacker as the real ones. Toward the goal, the malicious transmitter attempts to mislead the receiver by generating phantom CSI that matches the forged video.

4 SYSTEM DESIGN

4.1 Attack Overview

Existing wireless liveness detection systems rely on wireless environmental fluctuations to detect video- or voice-spoofing attacks. Our key idea of the proposed attack is to manipulate the wireless environmental fluctuations so that both the coexisting video/voice and CSI data have a consistent observation of human activities. Wireless liveness detection systems would thus be unaware of the spoofing attacks. Without loss of generality, we assume that the attacker aims to launch video spoofing attacks.

In a typical video spoofing attack, the attacker replaces the live video frames with fake ones (e.g., what are previously recorded) so that she can perform activities in the area monitored by the camera without being recorded. With a stream of video frames, the *data pre-processing* phase first identifies body keypoints in each video frame. Such keypoints input to the *event detection* phase, which determines the ongoing event. After that, the *feature extraction* phase generates semantic features from the processed video data, which are compared with that extracted from the CSI to determine the authenticity of the captured video.

To make the receiver observe fake CSI, whose semantic features are consistent with that extracted from the video data, the attacker first specifies such artificial CSI, and then delivers it to the receiver by manipulating the transmitted signal. Since the transmitted signal has to experience the distortion effect applied by the real wireless channel, the attacker compensates for such distortion effect at the transmitter side. Consequently, the receiver extracts the semantic features of the ongoing event with estimated CSI. Figure 3 depicts the flow chart of the proposed attack.

4.2 Video-based Pipeline

Traditional video-based monitoring system usually involves three steps, data pre-processing, event detection, and feature (i.e., event parameter) extraction.

Data Pre-processing: *OpenPose* is the first open-source real-time video processing tool for 2D pose detection, including tracking body, foot, hand, and facial keypoints [7]. It is also widely used in existing wireless liveness studies (e.g., [24, 29]). We also utilize OpenPose to process video frames, each of which then generates X-Y coordinates of the 18 body keypoints. Figure 4 shows an example of the body keypoints extracted from a video frame using OpenPose.

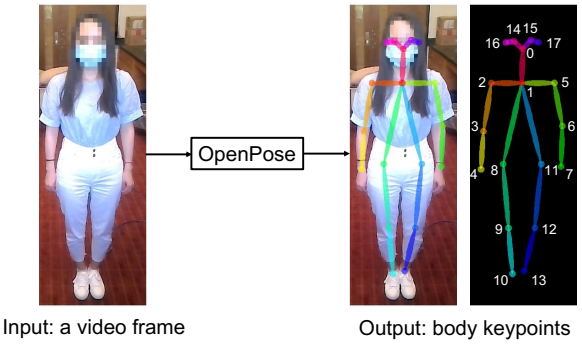


Figure 4: Body keypoints extracted by *OpenPose*.

We see that there are 18 keypoints (labeled with 0-17) of the target person. The displacement of those keypoints over time can then help infer occurrent events (e.g., human activities).

Event Detection: The input of this step is the X-Y coordinates of the 18 body keypoints extracted from each video frame. Let P_m^i denote the i^{th} point in the m^{th} video frame, where $i \in \{1, 2, \cdots, 18\}$ and $m \in \{1, 2, \cdots, M\}$, where M denotes the total amount of video frames. The Euclidean distance of each point between the m^{th} frame and the $(m+1)^{th}$ can be denoted as $L_m^i = \left|P_m^i - P_{m+1}^i\right|$. We then add up all these Euclidean distances and obtain the sum $D_m = \sum_{i=1}^{18} L_m^i$. If D_m is larger than the predefined threshold D_0, we regard that the motion is detected; otherwise, there is no motion detected if $D_m \leq D_0$. We iterate over all neighboring video frames with this scheme, separating dynamic scenes (with motion) from static ones.

Feature Extraction: We need to select a set of distinctive semantic features of motion, so that we can use them to design corresponding phantom-CSI flows. The start time and the end time of motion are often chosen as such features. If the motion occurring in the video is periodic, the motion frequency is also recorded as another. Particularly, to determine the frequency, we apply a metric referred to as motion energy which captures the energy in the different frequency bands of the body keypoints. With the FFT profile of the body keypoints, a single frequency component that exhibits the maximum signal magnitude can be extracted.

4.3 Artificial CSI Generation

The attacker would deliver specified CSI to the receiver, which matches events occurring in the injected fake video. Let $\mathbf{h}_T(t) = [h_{T_1}(t), h_{T_2}(t), \cdots, h_{T_N}(t)]$ denote the target CSI for N subcarriers. Intuitively, we may pre-record the CSI corresponding to the events in the video as $\mathbf{h}_T(t)$. However, this profiling process of collecting CSI is laborsome and may place an extra burden on the attacker. Instead, we propose a method that enables the attacker to generate such artificial CSI.

In general, to craft $\mathbf{h}_T(t)$, there are the following two cases: 1) the video just contains static images and has no human activity in the video; 2) the video contains human activity. For the first case, the target CSI $\mathbf{h}_T(t)$ can be easily crafted, denoting the random noise in the environment. For the latter case, we then need to convert the human activities into $\mathbf{h}_T(t)$.

Different human activities may cause different impacts on the environmental CSI. Specifically, the CSI amplitude on a sensitive subcarrier often shows a strong correlation with human activities. As a non-synchronized transmitter and receiver pair may bring an unknown phase lag [45], the CSI amplitude is often only chosen to characterize the wireless channel for human activity detection. Correspondingly, this paper also focuses on wireless liveness detection using CSI amplitudes.

It is widely observed that periodic movement usually makes the CSI amplitude on a sensitive subcarrier present a sinusoidal-like pattern over time [57]. Let f_a denote the frequency (Hz) of occurred event. We then convert the event into a subcarrier CSI $h_{T_i}(t) = |h_{T_i}(t)| \cdot e^{j\theta(t)} + N_i(t)$, where $|h_{T_i}(t)|$ represents amplitude. We model the CSI envelope on a sensitive subcarrier as a sinusoidal-like wave, i.e.,

$$|h_{T_i}(t)| = a \cdot sin(2\pi f_a t + \beta) + N_i, \text{when } t \in [\tau_s, \tau_e], \quad (1)$$

where a, β, and N_i are the amplitude, initial phase, and additive noise. When $t \notin [\tau_s, \tau_e]$ (i.e., outside of the activity period), there is no need to craft specific CSI and we then have $|h_{T_i}(t)| = 0$. In turn, with such a CSI envelope, the receiver can infer the start and end times of the activity, as well as the event frequency.

4.4 Transmission Manipulation

To invalidate wireless liveness detection, the transmitter (i.e., attacker) needs to make the receiver believe the target CSI $h_{T_i}(t)$ on sensitive subcarriers. To achieve this goal, the following three steps are required towards crafting the transmitted signal.

4.4.1 Winnowing Sensitive Subcarrier. Due to the multipath effect, signals usually arrive at the receiver via different paths, e.g., line-of-sight (LOS) and non-line-of-sight (NLOS). These signals may interfere constructively or destructively, leading the receiver to observe enhanced or weakened signals. This phenomenon may vary for different subcarriers as they have varying wavelengths. Consequently, all subcarriers can be divided into two groups: sensitive and insensitive. Sensitive subcarriers show large amplitudes (or variances), while insensitive subcarriers have imperceptible signal fluctuations. Thus, observations on sensitive subcarriers are utilized to detect human activities.

We utilize a binary decision variable α_i to indicate the subcarrier sensitivity, with 1 denoting sensitive while 0 showing insensitive. Since insensitive subcarriers are not involved in wireless liveness detection decisions, we only exploit sensitive subcarriers for achieving CSI manipulation.

4.4.2 Desensitizing. Since the transmitted signal has to experience the real wireless channel, the transmitter needs to cancel the actual distortion effect of the real channel. We call this process *desensitizing*. Let $h_{r_i}(t)$ denote the real CSI of the i^{th} sensitive subcarrier, and $d_i(t)$ represent the corresponding coefficient of the desensitizing module. $d_i(t)$ would be the inverse of $h_{r_i}(t)$ to eliminate the impact of the real channel on the transmitted signal $x(t)$. We then have $d_i(t) \cdot h_{r_i}(t)$=1, i.e., $d_i(t) = h_{r_i}^{-1}(t)$.

Activity Removal in Dynamic Scenarios: Generally, to obtain the real CSI in environments with human motion, an attacker can utilize a CSI profiling process. Particularly, rhythmic human activities (e.g., breathing) periodically affect the CSI waveforms, and the resultant CSI often presents a sinusoidal-like pattern, which can be then modeled by the attacker, as illustrated in Section 4.3.

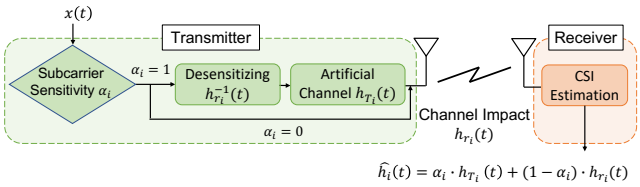


Figure 5: Subcarrier-level CSI wave morphing.

Signal Annihilation in Realistic Settings: To cancel the real channel effect, the attacker needs to know the real CSI via CSI profiling or modeling ahead. In certain cases, the human activity is complex and the real CSI is not available. However, the attack impact still exists. Although the attacker cannot control the CSI obtained at the receiver, she can then utilize a random coefficient of the desensitizing model. This may not successfully cancel the real channel effect, but it can make the target wireless liveness detection system obtain random and incorrect decisions. In the following, we focus on the scenarios where the attacker has knowledge of the real CSI due to the higher manipulability and more misleading nature of such attacks.

4.4.3 Creating Artificial Channel. After canceling the real channel effect, the attacker also needs to create an artificial channel to make the receiver obtain the target CSI, crafted during the phase of event-CSI conversion, as demonstrated in Section 4.3. Let $h_{a_i}(t)$ denote the specified CSI of the artificial i^{th} subchannel, and we thus obtain $h_{a_i}(t) = h_{T_i}(t)$.

Figure 5 illustrates subcarrier-level transmission signal manipulation. We use $x_{a_i}(t)$ to show the actual transmitted signal on the i^{th} subchannel. After the original signal $x(t)$ goes through the two steps of desensitizing and artificial channel, we have $x_{a_i}(t) = (1-\alpha) \cdot x(t) + \alpha \cdot x(t) \cdot h_{r_i}^{-1}(t) \cdot h_{a_i}(t)$. The received signal at the receiver then becomes $y_{a_i}(t) = x_{a_i}(t) \cdot h_{r_i}(t)$ (where we omit the noise term for the sake of simplicity). With $y_{a_i}(t)$ and the publicly known training sequence, the receiver can estimate the subcarrier CSI $\hat{h_i}(t)$, i.e., $y_{a_i}(t) = x(t) \cdot \hat{h_i}(t)$. As a result, we have

$$\hat{h_i}(t) = \alpha \cdot h_{T_i}(t) + (1-\alpha) \cdot h_{r_i}(t). \tag{2}$$

Consequently, for insensitive subcarriers ($\alpha = 0$), we obtain $\hat{h_i}(t) = h_{r_i}(t)$, i.e., no manipulation is applied; while for sensitive subcarriers ($\alpha = 1$), we have $\hat{h_i}(t) = h_{T_i}(t)$, demonstrating that the proposed method is able to make the receiver estimate the specified CSI via creating an artificial channel.

Synchronization for Real CSI Cancellation: CSI patterns (e.g., peaks and valleys in sinusoidal waves) change with human motion, and CSI during the motion period shows a larger variance than those happing out of the period. We can thus utilize human motion and the corresponding CSI feature to achieve synchronization, so that the real channel effect can be compensated.

4.5 CSI-aided Liveness Detection

With both the video and CSI signals, as discussed in Section 2.2, we apply the general wireless liveness detection process in existing studies (e.g., [29]). Particularly, we first synchronize both signals and then process each. The video data processing follows the procedures described in Section 4.2, while the CSI-based monitoring pipeline is an inverse process of event-CSI conversion, including CSI data preprocessing, event detection, and feature extraction. Finally, we cross-check features extracted from the two sources to determine whether a spoofing attack happens.

4.5.1 CSI and Video Data Synchronization. Spoofing detection relies on the concurrent camera and wireless signals, thus it is crucial to synchronize both. The out-of-sync data may result in different semantic features, causing a high false alarming rate when they are used for spoofing detection [24].

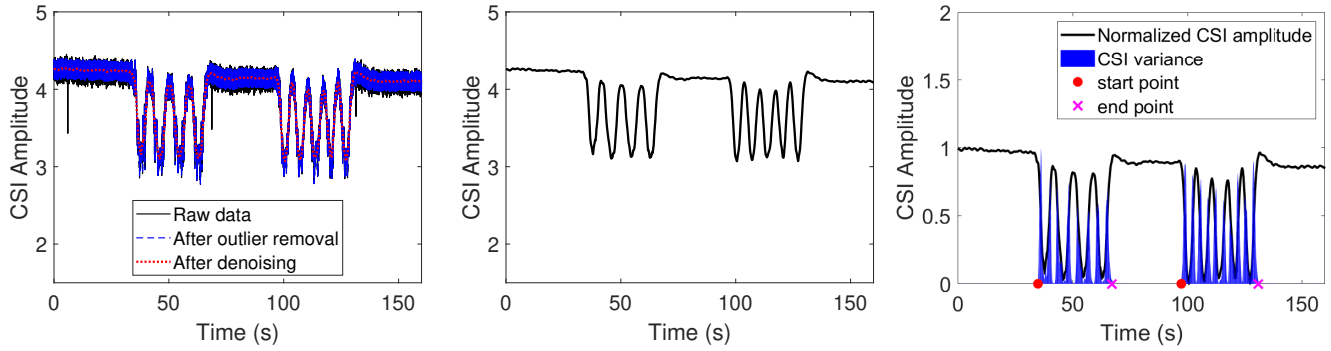
Suppose that f_v denotes the frame per second (FPS) or frame rate of the camera, and Δ_v represents frame interval, i.e., the interval between two consecutive frames. The frame interval is normally constant and mathematically, we have $\Delta_v = 1/f_v$. The common frame rates for video are 24 FPS (standard), 30 FPS (close-second standard), and 60 FPS (for slow motion) [51]. Thus, the corresponding frame intervals are 42 ms, 33 ms, and 17 ms. Meanwhile, let f_w represent the CSI sampling rate at the receiver, which is much larger than f_v. We use N_c to denote the number of CSI measurements that a frame interval corresponds to. Note that if there is no packet loss, N_c is constant and equals $\frac{f_w}{f_v}$. Due to packet loss, unlike video frames, CSI measurements may have variable time intervals between them. As a result, each frame interval corresponds to a varying number of CSI measurements, i.e., N_c varies.

To address the issue, we apply linear interpolation to resample CSI measurements with a constant interval $\Delta_c = \frac{\Delta_v}{N_c}$, so that each video frame corresponds to a fixed amount of resampled CSI measurements.

4.5.2 CSI Data Preprocessing. The imperfect CSI can be caused by environmental noise, radio signal interference, and hardware imperfection. CSI data preprocessing includes (1) outlier removal and noise reduction, making CSI more accurately reflect the impact of human activities; (2) Principle Component Analysis (PCA) [48], reducing dimensionality of feature vectors to facilitate data analysis.

Outlier Removal and Noise Reduction: The collected CSI series may have some abrupt changes that are not caused by human activities, and such abnormal values should be corrected. Hampel filter is generally applied to identify and replace outliers (which differ significantly from other samples) in a given series [11, 41]. It uses a sliding window of configurable width to go over the input data. For each window, the median η and the median absolute deviation (MAD) λ can be calculated. The sample of the input is regarded as an outlier if it lies outside of the range of $[\eta - \gamma \cdot \lambda, \eta + \gamma \cdot \lambda]$, where γ is a pre-determined scalar threshold. In this way, the Hampel filter is able to identify all outliers in the CSI series and then replace them with the corresponding median.

Besides, CSI variations caused by human activities may occur at the low end of the frequency range. We thus utilize the moving average filter [49] to smooth the CSI series. This filter is simple to use and is optimal for retaining a sharp step response [38]. It computes the arithmetic mean of M input points at a time to produce each point of the output stream, where M is the pre-defined number of points. Thus, the high-frequency noise in the raw CSI measurements can be eliminated.



(a) Outlier removal and noise reduction. (b) Waveform after PCA. (c) Variance scan and event extraction.

Figure 6: Procedures of CSI data preprocessing.

Figure 6a shows an example of applying outlier removal and noise reduction, where we effectively reduce outlier peaks and the strong high-frequency noise.

Dimension Reduction: We apply the PCA technique to decrease computational complexity by converting the received CSI into a set of orthogonal components (i.e., the most representative or principal components), which are influenced by human activity. Meanwhile, PCA also facilitates removing the uncorrelated noisy components. Figure 6b shows the CSI waveform after PCA, and we can clearly observe CSI fluctuations that correspond to human activity and smooth waveform, indicating static periods within which there is no human activity.

4.5.3 Event Detection. Generally, when there is no movement in the monitored area, the CSI fluctuation is small and maintains stability in the time domain [65], while human activity would bring distinguishable CSI fluctuations [63]. To segment CSI waveforms corresponding to human activities, we need to determine the start and end points of the CSI time series, which covers as much of the activity-disturbed waveform as possible while minimizing the coverage of the non-activity portion.

We then calculate the moving variance σ^2 of each window $\mathbf{h} = \{h_1, h_2, \cdots, h_J\}$, where J is the pre-defined size of the window and h_j is the j^{th} CSI value in this window. Mathematically, we have $\sigma^2 = \frac{\sum_{j=1}^{J}(h_j-\mu)^2}{J-1}$, where μ is the mean CSI value of the window $\mathbf{h}$. Empirically, the CSI segments during the human motion period show a much larger variance than those happening out of the period. Thus, we are only interested in the CSI segments with a variance larger than a predetermined threshold while ignoring the segments with a variance under this threshold. Later, those segments containing information about human activities will be further processed to extract semantic features about human activities. As shown in Figure 6c, by scanning the CSI variances, we can determine the start and end points for each event (two are detected, occurring during [34.9 s, 67.1 s] and [97.3 s, 130.7 s], respectively).

4.5.4 Feature Extraction. With CSI segments during human activities, a set of distinctive semantic features would be extracted and compared with that obtained from the video streams. The time period of human activities intercepted by CSI waveforms and video frames would usually match. Thus, the start and end times of each CSI segment, corresponding to that of human activity, will be recorded as the features. The frequency of CSI variations denotes the frequency of the event, which the video frames can also generate. Accordingly, we use the inter-peak intervals (i.e., the time period between successive peaks) to compute the frequency of occurred events.

As the first derivative of a peak switches from positive to negative at the peak maximum, it can be used to localize the occurrence time of each peak. However, noise may occasionally bring fake peaks and consequently false zero-crossings. Generally, the event usually cannot occur beyond a certain frequency. This observation enables us to develop a threshold-based fake peak removal algorithm. Specifically, if the calculated interval between the current peak with the previous one is less than $1/f_{max}$ (seconds), where f_{max} (Hz) denotes the maximum possible event frequency, this peak will be labeled as a fake one and thus discarded.

Let p_i denote the number of true peaks detected via an event-associated CSI segment, and $[t_1, t_2, \cdots, t_{p_i-1}]$ denote the corresponding sequence of inter-peak intervals. The event frequency f can be then estimated using the mean inter-peak interval, i.e., $f = \frac{p_i-1}{\sum_{j=1}^{p_i-1} t_j}$.

4.5.5 Consistency Checking. Given two tuples of features $\mathbf{f}^v = [f_1^v, \cdots, f_n^v]$ (from video) and $\mathbf{f}^c = [f_1^c, \cdots, f_n^c]$ (from CSI), where n is the number of extracted features, the multi-feature similarity score S can be calculated by comparing the similarity of each corresponding feature.

If the difference between the two features, each extracted from one of the two sources, is within a predefined threshold, we regard that both sources show the same feature. Mathematically, let s_j denote the single-feature similarity score and it can be obtained through

$$s_j = \begin{cases} 1 & \text{if } |f_j^v - f_j^c| \le D_j \\ 0 & \text{otherwise} \end{cases}, j \in [1, \cdots, n]. \tag{3}$$

D_j is chosen empirically to achieve a high detection accuracy with a low false positive rate. We set the optimal thresholds for both

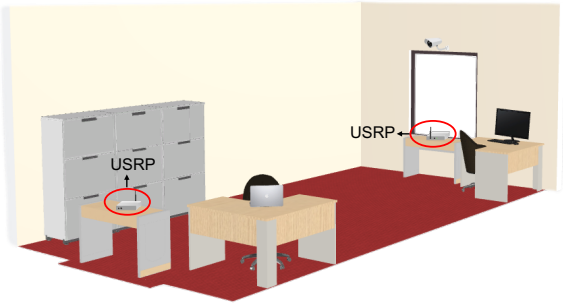


Figure 7: Layout of the experimental environment.

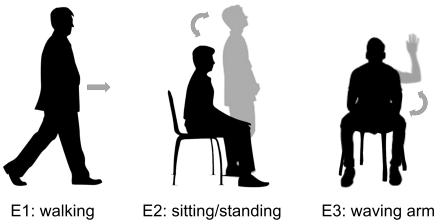


Figure 8: Three daily events.

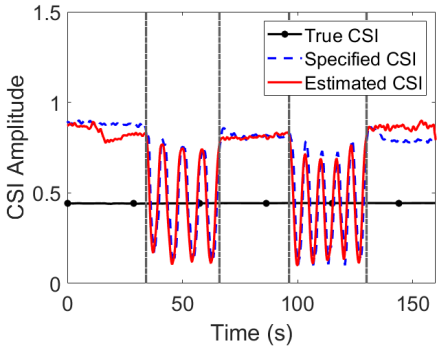
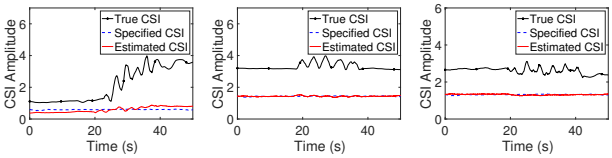


Figure 9: Channel manipulation in a static environment.



(a) Walk. **(b) Sit/stand.** **(c) Wave arms.**

Figure 10: Channel manipulation in a dynamic environment.

the start and end times as 1.5 seconds, and that for the event frequency as 0.08 Hz. As a result, we have $S(i) = \sum_{j=1}^{n} s_j$. If all features extracted from both sources are consistent, i.e., $S(i) = n$, we determine that there is no spoofing attack present; otherwise, the video spoofing attack is detected.

5 EXPERIMENTAL RESULTS

We implement an existing wireless liveness detection (e.g., [29, 30]) and our proposed attack on top of a typical surveillance camera (CODi HD 1080p [10]) and two USRP X300s [14], each equipped with an SBX-120 daughterboard [13].

5.1 Evaluation Setup

We perform the experiment in a laboratory office. For a good field of view, the camera is mounted on a wall 2.2 meters above the floor to monitor human activities in the office. It creates 1280×720 RGB images at 30 frames per second (FPS). Meanwhile, a wireless transmitter and receiver pair is utilized to verify the authenticity of the recorded video. Each node is a USRP X300.

The channel estimation algorithm runs at the receiver to extract the CSI for liveness detection. The attacker launches the phantom-CSI attack by replacing the original real-time video frames with pre-recorded fake ones (e.g., [4, 20]) and simultaneously manipulating the transmitted signal, aiming to make both the recorded video and the measured CSI at the receiver consistently show the same human activities. Figure 7 shows the positions of the camera, the transmitter, and the receiver.

We ask the user to perform the following three daily activities, as shown in Figure 8, including E_1: walking on the floor; E_2: sitting on a chair and then standing on the floor; E_3: moving the arm up and down. We consider two typical attack scenarios based on the goal of the attacker.

- *Fabricating Event*: when no event occurs in the monitored area, the attacker feeds a video with motion to the camera and synchronously makes the CSI detect the same motion.
- *Hiding Event*: when motion appears in the area, the attacker feeds a static shot to the camera and meanwhile makes CSI exhibit no motion.

Metrics: We use the following two evaluation metrics.

- *True Positive Rate*: this is the percentage of actual spoofing incidents that are correctly detected, denoting the accuracy of the spoofing detection.
- *False Positive Rate*: this is the proportion of all negatives (i.e., when no spoofing occurs) that are wrongly categorized as cases with spoofing.

5.2 Effectiveness of Channel Manipulation

In the section, we utilize examples to demonstrate the effectiveness of channel manipulation in different environments, which aims to make the receiver obtain the channel specified by the attacker.

Static Environment: Figure 9 presents the true CSI between the transmitter and the receiver, the CSI specified by the attacker, and the estimated CSI at the receiver in a static environment (with no human activity). We can observe that the estimated CSI is greatly similar to the specified one, while both significantly deviate from the true CSI. The estimated CSI further causes the receiver to believe that there are human activities during the periods from 34.2 s to 66.1 s, and from 96.3 s to 130.0 s. The activity repeats four and five times in the two periods, respectively. When the attacker injects a fake video with such events (e.g., waving arms) into the camera, the system would alert as the true CSI and the video detect inconsistent results without our attack, whereas our attack can successfully bypass the CSI-aided liveness detection system.

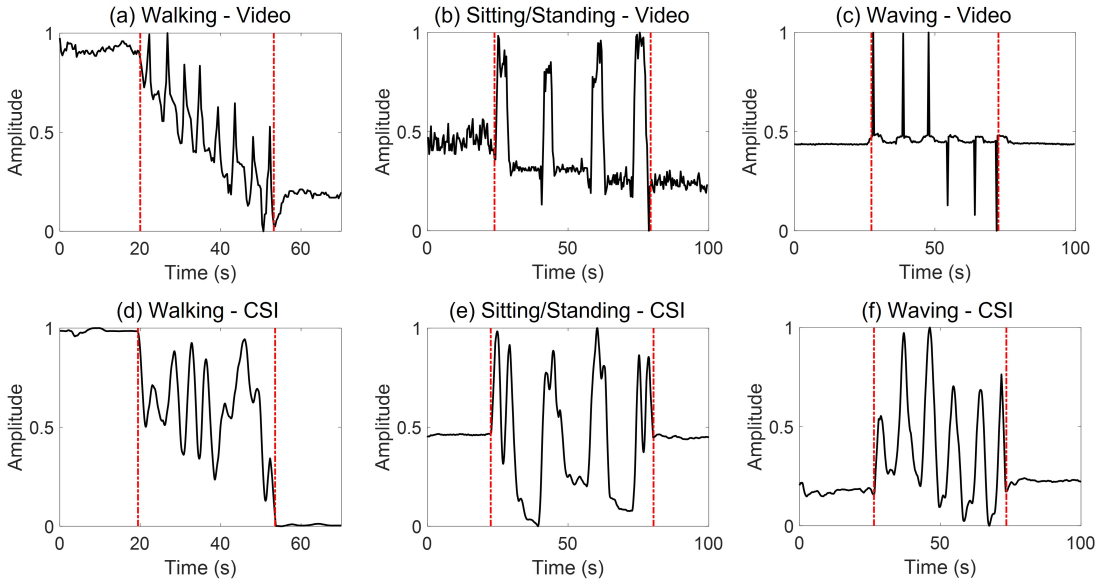


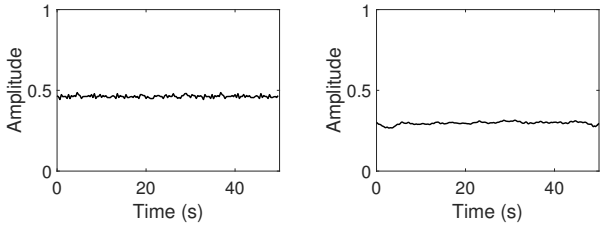
Figure 11: Video and the CSI signals when fabricating events.

Dynamic Environment: Figure 10 presents the true CSI between the transmitter and the receiver, the CSI specified by the attacker, and the estimated CSI at the receiver in an environment with human activities present. Human activities bring fluctuations in the CSI waveforms. Specifically, a walking activity involves significant body movements and location changes. Thus, it causes significant CSI changes over time. However, an in-place activity, i.e., sitting/standing and waving arms, only involves relatively smaller body movements and does not cause significant CSI changes. Also, channel manipulation enables the receiver to obtain an estimated CSI that is almost flat and close to the CSI specified by the attacker, causing the receiver to believe that no event happens. Thus, when the attacker injects a fake static video into the camera and meanwhile human activities occur in the monitored area, the system may alert without our attack due to the inconsistent detection results from the video and CSI, whereas our attack can make the CSI present no event and succeed to defraud the CSI-aided liveness detection system.

5.3 Two Attack Cases

Case I - Fabricating Nonexistent Events: The attacker makes the estimated CSI at the receiver side change with the injected fake video containing scenes of human activities, where the environment is in fact static.

Figure 11 compares the time series of the video and CSI when the fake video contains different activities. As shown in Figures 11a and 11d, with the video signal, the extracted feature tuple (including start time, end time, and frequency) for walking equals $(20.0\ s, 53.2\ s, 0.15\ Hz)$; with the CSI data stream, the corresponding tuple is $(19.5\ s, 53.5\ s, 0.15\ Hz)$. The absolute errors between features from the two sources are thus 0.5 s, 0.3 s, and 0. As the optimal thresholds for start time, end time, and event frequency are 1.5 s, 1.5 s, and 0.08 Hz, the similarity score equals 3. We have similar observations for the cases of sitting/standing (Figures 11b and 11e) and waving arms (Figures 11c and 11f). In all cases, our attack successfully bypasses wireless video liveness detection.



(a) Video stream. **(b) CSI stream.**

Figure 12: Video and CSI signal comparison when hiding events.

Case II - Hiding True Events: The attacker aims to make the CSI disclose no human activities when feeding a fake video containing only static scenes, though the user performs activities in the monitored area.

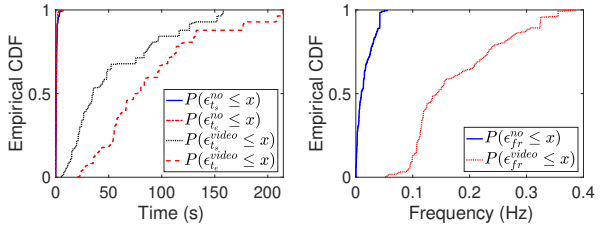
When the spoofed video contains no person, *OpenPose* extracts no keypoints from it and thus shows the empty output. When the spoofed video of a static scene contains a still user, the extracted keypoints have no movement, as shown in Figure 12a. Figure 12b plots the corresponding CSI time series obtained at the receiver side when the user performs events (e.g., walking). From the video and CSI signals, the respective extracted features are consistent. Thus, the wireless liveness detection system generates no alarm of spoofing detection, verifying the success of the proposed attack.

5.4 Overall Attack Impact

We test both static and dynamic environments. Each has two scenarios: (i) the attacker launches a video spoofing only attack; (ii) the attacker launches the proposed attack. For comparison, we

Table 1: Different human activity combinations.

Number of events	Human activity combination
1	E1 only; E2 only; E3 only
2	E1+ E2; E2 + E3; E1 + E3
3	E1+E2+E3



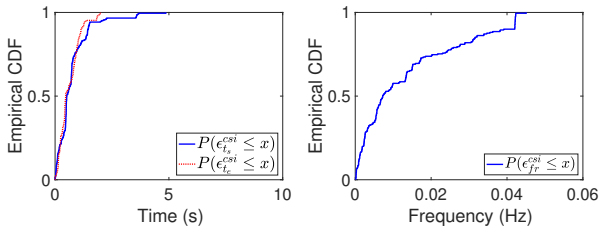
(a) Start/end time. (b) Event frequency.

Figure 13: CDF of the extracted features in a normal situation and when a video spoofing only attack happens.

also test the performance of the wireless liveness detection system when there is no any attack. The above three scenarios are referred to as "video", "csi", and "no", respectively. We consider the number of actual or spoofed events ranging from 1 to 3, and test 7 different combinations of the three daily events (E_1, E_2, and E_3), as shown in Table 1, where "$E_i + E_j$" ($i, j \in \{1, 2, 3\}$) denotes that events E_i and E_j occur sequentially. For every combination under each case, we perform 10 trials. Thus, in total, we perform $(2 \times 2 \times 7 + 7 + 1) \times 10 = 360$ attempts.

Event Feature Matching: Let $\epsilon_{t_s}^{sce}$, $\epsilon_{t_e}^{sce}$, and ϵ_{fr}^{sce} denote the measured absolute estimation errors for start time, end time, and event frequency, in scenario sce ($sce \in no, video, csi$). We show the empirical cumulative distribution functions (CDFs) of $\epsilon_{t_s}^{no}$, $\epsilon_{t_e}^{no}$, $\epsilon_{t_s}^{video}$, and $\epsilon_{t_e}^{video}$ in Figure 13a. Also, Figure 13b shows the CDFs of ϵ_{fr}^{no} and ϵ_{fr}^{video}. We see that the absolute errors for all three features are always small with no attack. Specifically, $\epsilon_{t_s}^{no}$ and $\epsilon_{t_e}^{no}$ are less than 2.0 s with probabilities 92.9% and 98.6%, respectively; ϵ_{fr}^{no} is always less than 0.045 Hz. Such results clearly show that without any attacks, the co-existing video and CSI data are highly consistent, i.e., the false positive rate of wireless liveness detection is low. On the other hand, for a video spoofing only attack, the features extracted from the two sources show an apparent mismatch. We observe that $\epsilon_{t_s}^{video}$ and $\epsilon_{t_e}^{video}$ are larger than 7.8 s and 23.8 s with probability 97.6%, respectively. Also, ϵ_{fr}^{video} ranges from 0.05 to 0.39 Hz, and is larger than 0.07 Hz with probability 97.6%. These results convincingly demonstrate that the wireless liveness detection system can effectively detect video spoofing only attacks.

Figure 14 presents CDFs of $\epsilon_{t_s}^{csi}$, $\epsilon_{t_e}^{csi}$, and ϵ_{fr}^{csi}. We observe that the absolute estimation errors for all three features become consistently small. Particularly, $\epsilon_{t_s}^{csi}$ and $\epsilon_{t_e}^{csi}$ are less than 1.5 s with probabilities 93.8% and 95.3%; ϵ_{fr}^{csi} is less than 0.042 Hz with probability 98.6%. These results show that our attack can successfully synchronize the CSI and video signals observed at the receiver. With



(a) Start/end time. (b) Event frequency.

Figure 14: CDF of the extracted features with our attack.

Table 2: Wireless video liveness detection vs. feature count.

Count	Two		Three	
Case	*video spoofing only attack*	*our attack*	*video spoofing only attack*	*our attack*
TPR	1	3.1%	1	4.4%
FPR	4.4%	4.4%	13.3%	13.3%

Table 3: Impact of different event types.

	E_1		E_2		E_3	
Case	*Video**	*our attack*	*Video*	*our attack*	*Video*	*our attack*
TPR	1	5.0%	1	6.0%	1	4.3%
FPR	10.0%	10.0%	10.0%	10.0%	7.1%	7.1%

**Video*: *video spoofing only attack.*

consistent CSI and video data streams, the wireless liveness detection system would fail to send out an alarm when video spoofing attacks happen.

Impact of Feature Count: By comparing extracted features from both sources, it can determine whether the recorded video is spoofed or not. Table 2 presents TPRs and FPRs of the liveness detection system when the video spoofing only attack happens and when our attack initiates. We see that if using two features (start and end time), the overall TPR can be up to 1 when there is a video spoofing only attack, while it is decreased to as small as 3.1% when the proposed attack is launched. This implies that the CSI-aided liveness detection system can reliably detect traditional video spoofing attacks, but becomes ineffective with our attack (with just 9.1% accuracy). Besides, we observe that the proposed attack rarely has an impact on FPR, which maintains a relatively low value. Moreover, when using three features (start time, end time, and event frequency) for event detection, we have similar observations. Specifically, compared with the video spoofing only attack, the TPR of our attack is slightly increased but still below 4.5%, again indicating the attack effectiveness against the wireless liveness detection scheme.

Impact of Event Type: For different types of events in the spoofed video, we construct respective phantom CSI to launch our attack. As shown in Table 3, the TPR of the liveness detection system is always 100% under video spoofing only attacks regardless of event type, while it drops dramatically to 5.0%. 6.0%, and 4.3% for E_1, E_2, and E_3, respectively. Also, the FPRs across all event

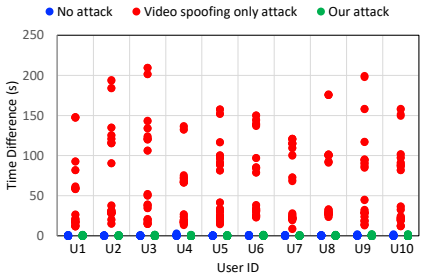


Figure 15: Event start time discrepancies.

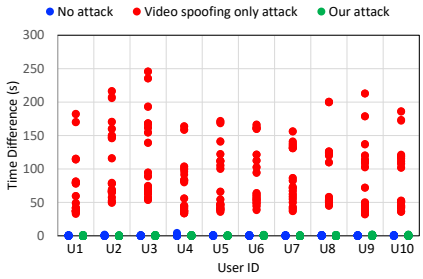


Figure 16: Event end time discrepancies.

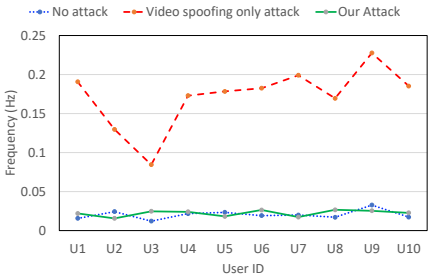


Figure 17: Mean frequency discrepancies.

Table 4: The list of voice commands we test.

ID	Command	Word #
C1	Please call 911	3
C2	Please play music	3
C3	Please open the door	4
C4	Please turn on the TV	5
C5	Please open the notification center	5

types under both scenarios are no larger than 10%. These results demonstrate our attack is robust against event type.

5.5 User Study

We recruited 10 volunteers (aged 18-35 years old; 5 self-identified as females and the rest as males).[1] Every participant was asked to perform each motion event in Table 1 twice in a normal scenario (i.e., without any attacks). We also recorded the corresponding videos and replayed them in the other two cases, i.e., the video spoofing only attack and the proposed attack. For each case, we test the performance of wireless video liveness detection for $(3+4+1)\times 2 = 16$ trials per participant.

Figures 15, 16, and 17 illustrate respective feature differences. We see that all feature differences are consistently low with no attack. Specifically, for the start/end time, the feature difference is less than 1.5 s while it is less than 0.03 for the frequency. With the video spoofing only attack, each feature discrepancy of all users increases greatly, which becomes an effective indicator of the existence of video spoofing. However, when the proposed attack is launched, all feature differences become consistently small again, similar to that in the scenario of no attack. These results convincingly demonstrate that an attacker can effectively bypass the wireless video liveness detection system with spoofed videos by launching the phantom-CSI attack.

6 ATTACK AGAINST WIRELESS VOICE LIVENESS DETECTION

Voice assistants, such as Amazon Alexa and Google Assistant, have been embedded in a slew of digital devices (e.g., smartphones and smart TVs). Due to the open nature of voice assistants' input channels, a malicious attacker could easily record people's use of voice commands [1, 16], and even build a model to synthesize a victim's voice [37]. The attacker plays pre-recorded or synthesized voice commands, which may spoof voice assistants, causing these devices to perform operations against the desires of their owners [5, 75]. Wireless voice liveness detection cross-checks the consistency between simultaneously obtained audio and wireless signals. Specifically, we preprocess audio signals using the spectral subtraction technique [6] to remove the background noise, where the average noise spectrum is first estimated and then subtracted from the noisy speech spectrum. By extracting semantic features (e.g., start time, end time, and word count) from the audio and wireless signals, spoofing attacks via pre-recorded or synthesized voice can be then detected [35, 36, 42, 53, 76]. Our attack can generate fake CSI and make it synchronized with the voice signal played by a speaker.

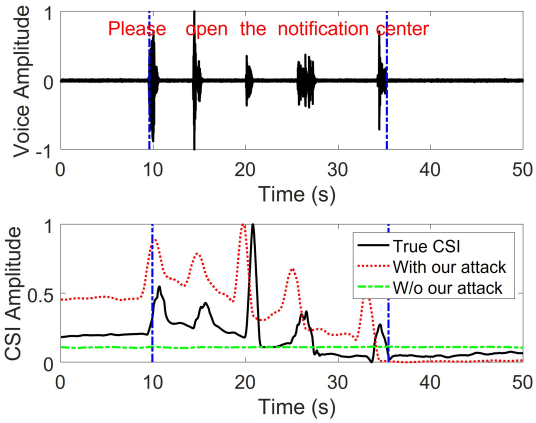


Figure 18: An example of a wireless-based voice liveness detection.

6.1 Implementation Setup

We implement wireless voice liveness detection and our attack in real-world environments. We utilize USRP X300 as a transceiver to collect CSI, and a microphone to collect voice signals. The transmitter and the receiver are placed at opposite positions relative to the target speaker. We randomly select 5 commands (C1-C5) from a list of the best Siri voice commands for a variety of daily tasks [8], as shown in Table 4. The evaluation metrics are the same with that for assessing the attack against wireless video liveness detection.

[1] Our study has been approved by our institution's IRB.

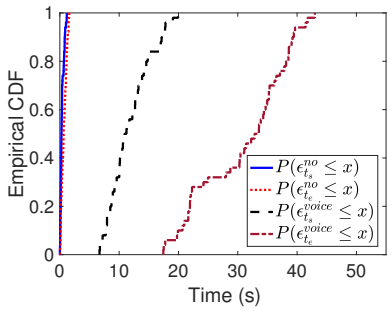


Figure 19: CDFs of start/end time for normal and voice spoofing attack only cases.

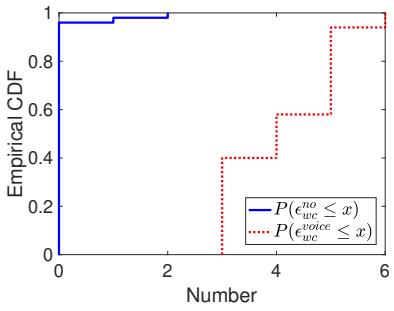


Figure 20: CDFs of word count for normal and voice spoofing attack only cases.

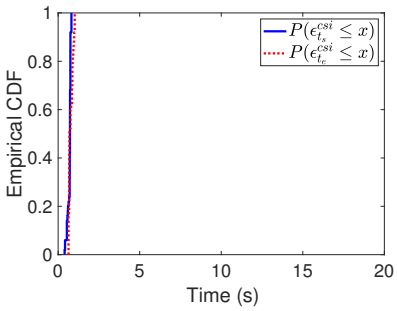


Figure 21: CDFs of start/end time when the proposed attack is launched.

6.2 Case Study

We compare the following cases: (1) *Normal Case*: the user speaks command C5 in Table 4; (2) *Voice Replay Only*: a speaker plays C5; (3) *Our Attack*. Figure 18 plots corresponding voice and CSI signals.

Normal Case: From the voice signal, the speaking interval is [9.6 s, 35.2 s] and there are 5 separate segments full of fluctuations, corresponding to 5 words. Meanwhile, the fluctuations of the CSI time series (referred to as "true CSI" in Figure 18) happen with the occurrence of the command; accordingly, we get the speaking interval [9.9 s, 35.3 s] and the word count 5 (as the sharp and rise pattern appears 5 times, each caused by speaking a word). Thus, the errors between corresponding features extracted from the voice and CSI signals are all small, indicating that both signals are consistent.

Voice Replay Only: When an attacker launches a voice spoofing only attack (with no mouth motion), the voice signal that the microphone captures maintains almost unchanged. However, the CSI waveform (referred to "W/o our attack' in Figure 18) becomes flat, demonstrating that the CSI would detect no event. The inconsistency of event detection via voice and CSI data facilitates the detection of the voice spoofing attack.

Our Attack: The waveform of the estimated CSI is highly similar to the true one. The correspondingly extracted features are 9.0 s, 34.4 s, and 5. By comparing them with the features extracted from the voice signal, we obtain the absolute errors as 0.6 s, 0.8 s, and 0, each of which is smaller than the respective threshold, indicating the failure of the liveness detection.

6.3 Overall Performance

For each command in Table 4, we perform the proposed attack 10 times. We synchronize the CSI and spoofed voice signals each time to bypass the wireless-based liveness detection system. For comparison, we also record the performance of the normal case with no attack, and the voice spoofing only attack. We refer to the above three scenarios as "*csi*", "*no*", and "*voice*", respectively.

Speaking Activity Detection: Let $\epsilon_{t_s}^{sce}$, $\epsilon_{t_e}^{sce}$, and ϵ_{wc}^{sce} denote the absolute estimation errors of start time, end time, and word count in scenario *sce*, where $sce \in \{no, voice, csi\}$. Figure 19 shows CDFs of $\epsilon_{t_s}^{no}$, $\epsilon_{t_e}^{no}$, $\epsilon_{t_s}^{voice}$ and $\epsilon_{t_e}^{voice}$. We see that $\epsilon_{t_s}^{no}$ is always less than 1.2 s and $\epsilon_{t_e}^{no}$ is less than 1.5 s with probability 98.0%, while $\epsilon_{t_s}^{voice}$ and $\epsilon_{t_e}^{no}$ are apparently larger. Meanwhile, ϵ_{wc}^{no} equals 0 with probability of 96.0%, whereas ϵ_{wc}^{voice} ranges from 3 to 6, as shown in Figure 20. These results convincingly imply that the wireless liveness detection system can effectively recognize voice spoofing attacks via feature differences. Figure 21 presents CDFs of $\epsilon_{t_s}^{csi}$ and $\epsilon_{t_e}^{csi}$. We see that $\epsilon_{t_s}^{csi}$ and $\epsilon_{t_e}^{csi}$ are always less than 0.8 s and 1.1 s, respectively. Also, ϵ_{wc}^{csi} is always 0. Evidently, with our attack, the extracted features from both voice and CSI signals are highly consistent, leading to the failure of the liveness detection system.

Table 5: Wireless voice liveness detection vs. feature count.

	Two			Three		
Case	no	voice	csi	no	voice	csi
TPR	N/A	1	0	N/A	1	0
FPR	6.0%	6.0%	6.0%	8.0%	8.0%	8.0%

Table 6: Wireless voice liveness detection vs. word count.

	3		4		5	
Case	voice	csi	voice	csi	voice	csi
TPR	1	0	1	0	1	0
FPR	10.0%	10.0%	10.0%	10.0%	5.0%	5.0%

Impact of Feature Count: Table 5 compares TPR and FPR for different cases when utilizing two features (start and end time) or three features (start time, end time, and word count) to detect spoofing attacks. We observe that regardless of the feature count, the wireless voice liveness detection system can achieve a TPR of 100% to recognize voice spoofing only attacks, while the TPR plummets to 0 with the proposed attack, implying that a voice replay attack is no longer to be correctly recognized. Meanwhile, we see that the FPR maintains small and consistent in different cases, demonstrating that our attack does not raise extra false alarms.

Impact of Number of Spoken Words: Aligned with existing work [36, 58, 66, 74], we also investigate the impact of count of spoken words. As show in Table 6, for word count ranging from 3 to 5, the FPR of the liveness detection system is always 100% without considering our attack, while it drops 0 under our attack. This verifies the robustness of our attack against word count. Also, the FPRs across all word counts for two cases are no larger than 10%, and the small fluctuation in FPR appears due to the minute changes in the environment.

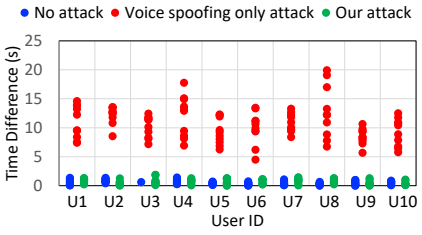


Figure 22: Speaking start time differences.

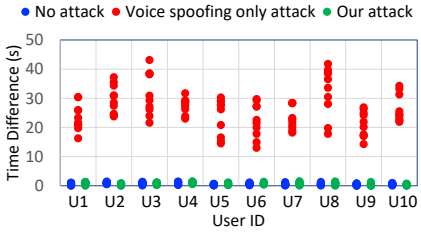


Figure 23: Speaking end time differences.

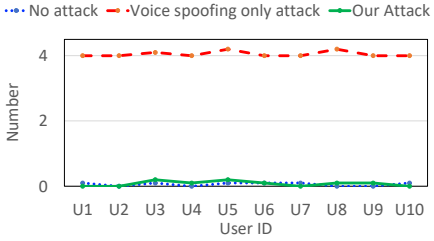


Figure 24: Mean word count differences.

6.4 User Study

The 10 volunteers (as described in Section 5.5) were asked to speak each command in Table 4 twice in a normal scenario. We also recorded the voices and replayed them in the other two cases with the voice spoofing only attack and our proposed attack, respectively. Figures 22, 23, and 24 illustrate respective feature discrepancies. We have the following observations. With no attack, the differences in both start time and end time are consistently low (less than 1.5 s) across all users. Also, the mean difference in word count for each user is always small (less than 0.1). However, for a voice spoofing only attack, the discrepancies in all features for all users jump sharply. These results convincingly show that the wireless liveness detection system can robustly detect voice spoofing only attacks. With our attack, however, those feature discrepancies decrease to small values, similar to that in the scenario of no attack, indicating that spoofed voice can successfully bypass the wireless voice liveness detection system.

7 DISCUSSIONS

7.1 Limitations

Cross-modality Sensing: Currently, the proposed attack targets compromising wireless video/voice liveness detection systems. Thus, except for generating fake CSI time series, it should also perform a video spoofing or voice replay attack simultaneously. In general, phantom-CSI can be utilized alone to confuse any CSI-based applications, such as keystroke recognition techniques [2, 15] or vital signs inference methods [25, 32].

Complex Human Activities: Our work currently just considers three popular daily activities (i.e., walking, sitting/standing, and waving arms), while a person may perform more complex activities (e.g., playing games). It may thus become difficult to construct phantom CSI associated with these activities. Accordingly, we expect that if the adversary could pre-collect CSI traces from such activities, she can feed them to the wireless liveness detection system to launch the proposed attack.

Scenarios Where Real CSI is Unknown: The proposed work may fail to make the receiver obtain the specific CSI in scenarios where real CSI is unavailable or cannot be correctly predicted. Machine learning-based approaches have demonstrated success in achieving accurate CSI prediction (e.g., [33, 69]). They thus can be added to our technique to improve the attack effectiveness, and we leave such integration to our future work.

Channels with Noise and Interference: Normally, if the real channel has noise and interference, existing wireless liveness detection may not work, and thus in this case, it is unnecessary to explore the feasibility of the proposed attack. The directional antenna can be adopted to eliminate CSI noises and other interferences.

7.2 Countermeasures

The proposed attack needs to compromise the transmitter and cancel the real channel effect before injecting phantom CSI to mislead the target system. Intuitively, to defend against such attacks, we can *utilize a trustful transmitter or a protected frequency* (on which the attacker is not allowed to inject signals). Such methods, however, would incur extra costs. Alternatively, we can also directly stop the attacker from obtaining the true wireless channel information by leveraging *friendly jamming*. Specifically, an ally jamming sends out intentional radio interference signals, i.e., jamming signals, to the wireless channel to prevent the attacker from measuring the real CSI, while the receiver itself can eliminate the impact of interference signals to guarantee that the wireless liveness detection system still works when the proposed attack is not launched. Similarly, this defense brings additional overhand for jamming hardware.

To validate the liveness detection result, another viable defense strategy is to integrate extra sensors. For example, the work [50] uses thermal infrared (IR) images to detect live signals; motion sensors can be employed to detect the presence of humans from the radiation of their body heat [19]; by exploiting the circular microphone array of the smart speaker, voice spoofing attacks can be thwarted [34]. However, these extra sensors are not always available, and the deployment of additional infrastructure requires authentication of the new sensor data that may potentially introduce a new attack surface [29].

8 RELATED WORK

In this section, we review two domains of prior works that are tightly related to the proposed phantom-CSI attack.

Wireless Human Activity Detection: Due to the pervasive, low-cost, and non-intrusive sensing nature, wireless human activity sensing has drawn increasing attention [31]. The received signal strength (RSS) or channel state information (CSI) obtained at the receiver may vary with environmental human activity. RSS represents the average power in a received wireless signal over the whole power bandwidth. Different from RSS, which uses synthetic values, CSI offers fine-grained channel information by decomposing the entire channel measurement into subcarriers and obtains better human activity detection performance than other metrics

(e.g., received signal strength) [21]. CSI contains both subcarrier-level amplitude and phase information. Extensive research efforts show that CSI amplitudes can capture various human activities, such as walking [60, 62, 71], breathing [32], gestures [55], and keystrokes [3, 15, 68]. Also, the work [61] exploits CSI phase difference data to monitor vital signs. Moreover, CSI amplitude and phase information can be employed together to achieve human activity detection [42, 43, 47, 72]. For example, the study [72] points out that human respiration cannot be detectable in all the locations when CSI amplitude or phase is used individually, and then proposes to use both phase and amplitude that are complementary to remove blind spots (where respiration detection experiences poor performance). Another study [42] presents that compared with using CSI amplitude alone, leveraging CSI amplitude along with CSI phase improves the accuracy of breathing rate estimation.

Liveness Detection: With the rapid advance in speech synthesis and video editing methods, it becomes increasingly popular to replay tampered voices/videos [27, 28, 59]. Specifically, in an audio replay attack, a recording of a target speaker's voice is replayed to a voice recognition system in place of genuine speech [28]; in a video spoofing attack, an attack can play back a clip of footage to cover up a crime [27]. With such spoofing techniques, attackers may bypass voice authentication or video monitoring, and even stealthily inject illegal voice commands or conduct malicious activities. To deal with these spoofing attacks, liveness detection is widely applied to differentiate the alive and present data (originating from live users) from forged data that are pre-recorded, concatenated, or synthesized by the attacker. Liveness detection against those spoofing attacks mainly includes the following three categories.

Intrinsic feature-based: Non-live representations often miss some intrinsic features in the corresponding live source. For example, a smartphone's loudspeaker usually presents strongly attenuated frequency responses in the low part of the spectrum [54], but it often has a high false acceptance rate to use this observation for liveness detection. Also, [74] uses the unique time-difference-of-arrival (TDoA) dynamic (i.e., the TDoA changes in a sequence of phoneme sounds to the phone's two microphones) for liveness detection, as it does not exist under replay attacks. Nevertheless, this method is not applicable to a device with only one microphone.

Another sensor-assisted: Liveness detection can also be achieved by combining a microphone/camera with other co-existing sensors [9, 22, 46, 73]. For example, [22] correlates sounds and breathing-induced chest motion (obtained via a gyroscope) to build a liveness detection system; [46] uses earbuds to measure the air pressure in the ear canal for voice liveness detection. These two methods, however, require the user to wear a chest-mounted gyroscope or earbuds. [73] leverages a speaker to emit inaudible signals, and exerts a microphone to record the reverberant signals to distinguish bone-conducted vibrations from air-conducted voices for liveness detection. Unfortunately, not all loudspeakers can emit ultrasound, which limits its practicality.

Wireless-based: There are emerging research efforts (e.g., [24, 29, 35, 36, 39, 42, 53, 76]) performing liveness detection leveraging wireless sensing due to its non-invasive and device-free nature, as well as the ubiquitous deployment of wireless infrastructures. In particular, [39] uses the ratio of the energy in motion affected bands (35-60 Hz) over the entire mmWave radar spectrogram as an indicator for liveness; [24, 29] develops techniques to detect video replay or forgery attacks using CSI extracted from wireless signals near the camera spot; [35] utilizes CSI to capture mouth motions, which can help distinguish authentic voice command from a spoofed one; [42] exploits the synchronized changes in voice and breathing to detect voice replay attacks. Our attack can make the CSI convey the same event semantic information with the spoofed video or voice signals, compromising those wireless liveness detection systems.

9 CONCLUSION

We have identified a new attack against liveness detection systems that use CSI to authenticate environmental human activities. Our *phantom-CSI* attack can manipulate CSI to exhibit the same semantic information as that measured by a co-existing camera or microphone, allowing spoofed video or voice signals to bypass the CSI-based liveness detection system. Our attack implementation on USRPs running GNURadio validates the effectiveness and robustness of the proposed attack, with experimental results showing that the proposed attack drastically lowers the true positive rates (TPRs) of the wireless liveness detection system from 100% to just 4.4% and 0% for detecting spoofed video and voice, respectively.

ACKNOWLEDGMENTS

We would like to thank all anonymous reviewers for their insightful comments. This work was supported in part by NSF under Grants No.1948547 and No.2155181.

REFERENCES

[1] Muhammad Ejaz Ahmed, Il-Youp Kwak, Jun Ho Huh, Iljoo Kim, Taekkyung Oh, and Hyoungshick Kim. 2020. Void: A fast and light voice liveness detection system. In *29th USENIX Security Symposium (USENIX Security 20)*. 2685–2702.

[2] Kamran Ali, Alex X Liu, Wei Wang, and Muhammad Shahzad. 2015. Keystroke recognition using WiFi signals. In *Proceedings of the 21st annual international conference on mobile computing and networking*. 90–102.

[3] Kamran Ali, Alex X Liu, Wei Wang, and Muhammad Shahzad. 2017. Recognizing keystrokes using WiFi devices. *IEEE Journal on Selected Areas in Communications* 35, 5 (2017), 1175–1190.

[4] Zach Banks and Eric Van Albert. 2015. Looping Surveillance Cameras through Live Editing of Network Streams. https://infocondb.org/con/def-con/def-con-23/looping-surveillance-cameras-through-live-editing-of-network-streams.

[5] Logan Blue, Luis Vargas, and Patrick Traynor. 2018. Hello, is it me you're looking for? Differentiating between human and electronic speakers for voice interface security. In *Proceedings of the 11th ACM Conference on Security & Privacy in Wireless and Mobile Networks*. 123–133.

[6] Steven Boll. 1979. Suppression of acoustic noise in speech using spectral subtraction. *IEEE Transactions on acoustics, speech, and signal processing* 27, 2 (1979), 113–120.

[7] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. 2019. OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields. *IEEE transactions on pattern analysis and machine intelligence* 43, 1 (2019), 172–186.

[8] Edgar Cervantes. 2021. The best Siri commands for productivity, information, laughter, and more. https://www.androidauthority.com/best-siri-commands-1094484/.

[9] Shaxun Chen, Amit Pande, and Prasant Mohapatra. 2014. Sensor-assisted facial recognition: an enhanced biometric authentication system for smartphones. In *Proceedings of the 12th annual international conference on Mobile systems, applications, and services*. 109–122.

[10] CODi. 2021. FALCO HD 1080P Auto Focus Webcam. https://www.codiworldwide.com/mobile-accessories/falco-hd-1080p-webcam/.

[11] Laurie Davies and Ursula Gather. 1993. The identification of multiple outliers. *J. Amer. Statist. Assoc.* 88, 423 (1993), 782–792.

[12] Wenrui Diao, Xiangyu Liu, Zhe Zhou, and Kehuan Zhang. 2014. Your voice assistant is mine: How to abuse speakers to steal information and control your phone.

In *Proceedings of the 4th ACM Workshop on Security and Privacy in Smartphones & Mobile Devices*. 63–74.
[13] Ettus Research. 2021. SBX 400-4400 MHz Rx/Tx (120 MHz, X Series only). https://www.ettus.com/all-products/sbx120/.
[14] Ettus Research. 2021. USRP X300. https://www.ettus.com/all-products/x300-kit/.
[15] Song Fang, Ian Markwood, Yao Liu, Shangqing Zhao, Zhuo Lu, and Haojin Zhu. 2018. No Training Hurdles: Fast Training-Agnostic Attacks to Infer Your Typing. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security* (Toronto, Canada) *(CCS '18)*. ACM, New York, NY, USA, 1747–1760.
[16] Huan Feng, Kassem Fawaz, and Kang G. Shin. 2017. Continuous Authentication for Voice Assistants. In *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking* (Snowbird, Utah, USA) *(MobiCom '17)*. Association for Computing Machinery, New York, NY, USA, 343–355.
[17] Andrea Goldsmith. 2005. *Wireless Communications*. Cambridge University Press, New York, NY, USA.
[18] Francesco Gringoli, Matthias Schulz, Jakob Link, and Matthias Hollick. 2019. Free your CSI: A channel state information extraction platform for modern Wi-Fi chipsets. In *Proceedings of the 13th International Workshop on Wireless Network Testbeds, Experimental Evaluation & Characterization*. 21–28.
[19] Yan He, Qiuye He, Song Fang, and Yao Liu. 2021. MotionCompass: Pinpointing Wireless Camera via Motion-Activated Traffic. In *Proceedings of the 19th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys)*. Association for Computing Machinery, New York, NY, USA, 215–227.
[20] Craig Heffners. 2013. Exploiting Network Surveillance Cameras Like a Hollywood Hacker. https://www.youtube.com/watch?v=B8DjTcANBx0.
[21] Peter Hillyard, Anh Luong, Alemayehu Solomon Abrar, Neal Patwari, Krishna Sundar, Robert Farney, Jason Burch, Christina Porucznik, and Sarah Hatch Pollard. 2018. Experience: Cross-technology radio respiratory monitoring performance study. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*. 487–496.
[22] Chenyu Huang, Huangxun Chen, Lin Yang, and Qian Zhang. 2018. BreathLive: Liveness detection for heart sound authentication with deep breathing. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2, 1 (2018), 1–25.
[23] Yong Huang, Xiang Li, Wei Wang, Tao Jiang, and Qian Zhang. 2021. Forgery Attack Detection in Surveillance Video Streams Using Wi-Fi Channel State Information. *IEEE Transactions on Wireless Communications* 21, 6 (2021), 4340–4349.
[24] Yong Huang, Xiang Li, Wei Wang, Tao Jiang, and Qian Zhang. 2021. Towards Cross-Modal Forgery Detection and Localization on Live Surveillance Videos. In *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM '21)*.
[25] Weijia Jia, Hongjian Peng, Na Ruan, Zhiqing Tang, and Wei Zhao. 2020. WiFind: Driver Fatigue Detection with Fine-Grained Wi-Fi Signal Features. *IEEE Transactions on Big Data* 6, 2 (2020), 269–282.
[26] Jesse S Jin, Changsheng Xu, Min Xu, Dai-Kyung Hyun, Min-Jeong Lee, Seung-Jin Ryu, Hae-Yeoun Lee, and Heung-Kyu Lee. 2013. Forgery detection for surveillance video. In *The Era of Interactive Media*. Springer, 25–36.
[27] Naor Kalbo, Yisroel Mirsky, Asaf Shabtai, and Yuval Elovici. 2020. The security of IP-based video surveillance systems. *Sensors* 20, 17 (2020), 4806.
[28] Tomi Kinnunen, Md Sahidullah, Héctor Delgado, Massimiliano Todisco, Nicholas Evans, Junichi Yamagishi, and Kong Aik Lee. 2017. The ASVspoof 2017 challenge: Assessing the limits of replay spoofing attack detection. (2017).
[29] Nitya Lakshmanan, Inkyu Bang, Min Suk Kang, Jun Han, and Jong Taek Lee. 2019. SurFi: detecting surveillance camera looping attacks with Wi-Fi channel state information. In *Proceedings of the 12th Conference on Security and Privacy in Wireless and Mobile Networks (WiSec '19)*.
[30] Nitya Lakshmanan, Inkyu Bang, Min Suk Kang, Jun Han, and Jong Taek Lee. 2019. SurFi: Detecting Surveillance Camera Looping Attacks with Wi-Fi Channel State Information (Extended Version). *arXiv preprint arXiv:1904.01350* (2019).
[31] J. Liu, H. Liu, Y. Chen, Y. Wang, and C. Wang. 2020. Wireless Sensing for Human Activity: A Survey. *IEEE Communications Surveys & Tutorials* 22, 3 (2020), 1629–1645.
[32] Jian Liu, Yan Wang, Yingying Chen, Jie Yang, Xu Chen, and Jerry Cheng. 2015. Tracking Vital Signs During Sleep Leveraging Off-the-Shelf WiFi. In *Proceedings of the 16th ACM International Symposium on Mobile Ad Hoc Networking and Computing* (Hangzhou, China) *(MobiHoc '15)*. Association for Computing Machinery, New York, NY, USA, 267–276.
[33] Changqing Luo, Jinlong Ji, Qianlong Wang, Xuhui Chen, and Pan Li. 2020. Channel State Information Prediction for 5G Wireless Communications: A Deep Learning Approach. *IEEE Transactions on Network Science and Engineering* 7, 1 (2020), 227–236.
[34] Yan Meng, Jiachun Li, Matthew Pillari, Arjun Deopujari, Liam Brennan, Hafsah Shamsie, Haojin Zhu, and Yuan Tian. 2022. Your microphone array retains your identity: A robust voice liveness detection system for smart speaker. In *USENIX Security*.
[35] Yan Meng, Zichang Wang, Wei Zhang, Peilin Wu, Haojin Zhu, Xiaohui Liang, and Yao Liu. 2018. WiVo: Enhancing the Security of Voice Control System via Wireless Signal in IoT Environment *(Mobihoc '18)*. Association for Computing Machinery, New York, NY, USA, 81–90.
[36] Yan Meng, Haojin Zhu, Jinlei Li, Jin Li, and Yao Liu. 2020. Liveness detection for voice user interface via wireless signals in IoT environment. *IEEE Transactions on Dependable and Secure Computing* (2020).
[37] Dibya Mukhopadhyay, Maliheh Shirvanian, and Nitesh Saxena. 2015. All your voices are belong to us: Stealing voices to fool humans and machines. In *European Symposium on Research in Computer Security*. Springer, 599–621.
[38] Eduardo F. Nakamura and Antonio A. F. Loureiro. 2008. Information Fusion in Wireless Sensor Networks. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data* (Vancouver, Canada) *(SIGMOD '08)*. Association for Computing Machinery, New York, NY, USA, 1365–1372.
[39] Muhammed Zahid Ozturk, Chenshu Wu, Beibei Wang, and KJ Liu. 2021. RadioMic: Sound Sensing via mmWave Signals. *arXiv preprint arXiv:2108.03164* (2021).
[40] Sameera Palipana, David Rojas, Piyush Agrawal, and Dirk Pesch. 2018. FallDeFi: Ubiquitous Fall Detection Using Commodity Wi-Fi Devices. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 1, 4, Article 155 (Jan. 2018), 25 pages.
[41] R.K. Pearson. 2002. Outliers in process modeling and identification. *IEEE Transactions on Control Systems Technology* 10, 1 (2002), 55–63.
[42] Swadhin Pradhan, Wei Sun, Ghufran Baig, and Lili Qiu. 2019. Combating replay attacks against voice assistants. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 3, 3 (2019), 1–26.
[43] Kun Qian, Chenshu Wu, Zheng Yang, Yunhao Liu, Fugui He, and Tianzhang Xing. 2018. Enabling contactless detection of moving humans with dynamic speeds using CSI. *ACM Transactions on Embedded Computing Systems (TECS)* 17, 2 (2018), 1–18.
[44] Muhammad Salman, Nguyen Dao, Uichin Lee, and Youngtae Noh. 2022. CSI:DeSpy: Enabling Effortless Spy Camera Detection via Passive Sensing of User Activities and Bitrate Variations. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 6, 2, Article 72 (jul 2022), 27 pages.
[45] Souvik Sen, Božidar Radunovic, Romit Roy Choudhury, and Tom Minka. 2012. You Are Facing the Mona Lisa: Spot Localization Using PHY Layer Information. In *Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services* (Low Wood Bay, Lake District, UK) *(MobiSys '12)*. Association for Computing Machinery, New York, NY, USA, 183–196.
[46] Jiacheng Shang and Jie Wu. 2022. Voice Liveness Detection for Voice Assistants through Ear Canal Pressure Monitoring. *IEEE Transactions on Network Science and Engineering* (2022).
[47] Cong Shi, Jian Liu, Hongbo Liu, and Yingying Chen. 2017. Smart user authentication through actuation of daily activities leveraging WiFi-enabled IoT. In *Proceedings of the 18th ACM International Symposium on Mobile Ad Hoc Networking and Computing*. 1–10.
[48] Jonathon Shlens. 2014. A tutorial on principal component analysis. *arXiv preprint arXiv:1404.1100* (2014).
[49] Steven W Smith. 1999. The scientist and engineer's guide to digital signal processing, Second Edition. (1999).
[50] Lin Sun, WaiBin Huang, and MingHui Wu. 2011. TIR/VIS correlation for liveness detection in face recognition. In *International Conference on Computer Analysis of Images and Patterns*. Springer, 114–121.
[51] Benjamin Tag, Junichi Shimizu, Chi Zhang, Kai Kunze, Naohisa Ohta, and Kazunori Sugiura. 2016. In the Eye of the Beholder: The Impact of Frame Rate on Human Eye Blink. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI EA '16)*. Association for Computing Machinery, New York, NY, USA, 2321–2327.
[52] Nils Ole Tippenhauer, Christina Pöpper, Kasper Bonne Rasmussen, and Srdjan Capkun. 2011. On the Requirements for Successful GPS Spoofing Attacks. In *Proceedings of the 18th ACM Conference on Computer and Communications Security* (Chicago, Illinois, USA) *(CCS '11)*. Association for Computing Machinery, New York, NY, USA, 75–86.
[53] Bang Tran, Shenhui Pan, Xiaohui Liang, and Honggang Zhang. 2021. Exploiting Physical Presence Sensing to Secure Voice Assistant Systems. In *ICC 2021 - IEEE International Conference on Communications*. 1–6.
[54] Jesus Villalba and Eduardo Lleida. 2011. Preventing replay attacks on speaker verification systems. In *2011 Carnahan Conference on Security Technology*. IEEE, 1–8.
[55] Aditya Virmani and Muhammad Shahzad. 2017. Position and orientation agnostic gesture recognition using wifi. In *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services*. 252–264.
[56] Guanhua Wang, Yongpan Zou, Zimu Zhou, Kaishun Wu, and Lionel M. Ni. 2014. We Can Hear You with Wi-Fi!. In *Proceedings of the 20th Annual International Conference on Mobile Computing and Networking* (Maui, Hawaii, USA) *(MobiCom '14)*. Association for Computing Machinery, New York, NY, USA, 593–604.
[57] Hao Wang, Daqing Zhang, Junyi Ma, Yasha Wang, Yuxiang Wang, Dan Wu, Tao Gu, and Bing Xie. 2016. Human Respiration Detection with Commodity Wifi Devices: Do User Location and Body Orientation Matter?. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing* (Heidelberg, Germany) *(UbiComp '16)*. Association for Computing Machinery, New York, NY, USA, 25–36.

[58] Qian Wang, Xiu Lin, Man Zhou, Yanjiao Chen, Cong Wang, Qi Li, and Xiangyang Luo. 2019. VoicePop: A Pop Noise based Anti-spoofing System for Voice Authentication on Smartphones. In *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*. 2062–2070.

[59] Shu Wang, Jiahao Cao, Xu He, Kun Sun, and Qi Li. 2020. When the differences in frequency domain are compensated: Understanding and defeating modulated replay attacks on automatic speech recognition. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*. 1103–1119.

[60] Wei Wang, Alex X. Liu, and Muhammad Shahzad. 2016. Gait Recognition Using Wifi Signals. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing* (Heidelberg, Germany) *(UbiComp '16)*. Association for Computing Machinery, New York, NY, USA, 363–373.

[61] Xuyu Wang, Chao Yang, and Shiwen Mao. 2020. On CSI-based vital sign monitoring using commodity WiFi. *ACM Transactions on Computing for Healthcare* 1, 3 (2020), 1–27.

[62] Yan Wang, Jian Liu, Yingying Chen, Marco Gruteser, Jie Yang, and Hongbo Liu. 2014. E-Eyes: Device-Free Location-Oriented Activity Identification Using Fine-Grained WiFi Signatures. In *Proceedings of the 20th Annual International Conference on Mobile Computing and Networking* (Maui, Hawaii, USA) *(MobiCom '14)*. Association for Computing Machinery, New York, NY, USA, 617–628.

[63] Bo Wei, Wen Hu, Mingrui Yang, and Chun Tung Chou. 2019. From real to complex: Enhancing radio-based activity recognition using complex-valued CSI. *ACM Transactions on Sensor Networks (TOSN)* 15, 3 (2019), 1–32.

[64] C. Wu, Z. Yang, Z. Zhou, X. Liu, Y. Liu, and J. Cao. 2015. Non-Invasive Detection of Moving and Stationary Human With WiFi. *IEEE Journal on Selected Areas in Communications* 33, 11 (2015), 2329–2342.

[65] Kaishun Wu, Jiang Xiao, Youwen Yi, Dihu Chen, Xiaonan Luo, and Lionel M. Ni. 2013. CSI-Based Indoor Localization. *IEEE Transactions on Parallel and Distributed Systems* 24, 7 (2013), 1300–1309.

[66] Libing Wu, Jingxiao Yang, Man Zhou, Yanjiao Chen, and Qian Wang. 2020. LVID: A Multimodal Biometrics Authentication System on Smartphones. *IEEE Transactions on Information Forensics and Security* 15 (2020), 1572–1585.

[67] Edwin Yang, Song Fang, Ian Markwood, Yao Liu, Shangqing Zhao, Zhuo Lu, and Haojin Zhu. 2022. Wireless Training-Free Keystroke Inference Attack and Defense. *IEEE/ACM Transactions on Networking* 30, 4 (2022), 1733–1748.

[68] Edwin Yang, Qiuye He, and Song Fang. 2022. WINK: Wireless Inference of Numerical Keystrokes via Zero-Training Spatiotemporal Analysis. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security* (Los Angeles, CA, USA) *(CCS '22)*. Association for Computing Machinery, New York, NY, USA, 3033–3047.

[69] Jide Yuan, Hien Quoc Ngo, and Michail Matthaiou. 2020. Machine Learning-Based Channel Prediction in Massive MIMO With Channel Aging. *IEEE Transactions on Wireless Communications* 19, 5 (2020), 2960–2973.

[70] Mustafa Harun Yılmaz and Hüseyin Arslan. 2015. A survey: Spoofing attacks in physical layer security. In *2015 IEEE 40th Local Computer Networks Conference Workshops (LCN Workshops)*. 812–817.

[71] Yunze Zeng, Parth H. Pathak, and Prasant Mohapatra. 2016. WiWho: WiFi-Based Person Identification in Smart Spaces. In *2016 15th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. 1–12.

[72] Youwei Zeng, Dan Wu, Ruiyang Gao, Tao Gu, and Daqing Zhang. 2018. FullBreathe: Full Human Respiration Detection Exploiting Complementarity of CSI Phase and Amplitude of WiFi Signals. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 2, 3, Article 148 (sep 2018), 19 pages.

[73] Linghan Zhang, Sheng Tan, Zi Wang, Yili Ren, Zhi Wang, and Jie Yang. 2020. VibLive: A Continuous Liveness Detection for Secure Voice User Interface in IoT Environment. In *Annual Computer Security Applications Conference*. 884–896.

[74] Linghan Zhang, Sheng Tan, Jie Yang, and Yingying Chen. 2016. Voicelive: A phoneme localization based liveness detection for voice authentication on smartphones. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. 1080–1091.

[75] Zhaohe (John) Zhang, Edwin Yang, and Song Fang. 2021. CommanderGabble: A Universal Attack Against ASR Systems Leveraging Fast Speech. In *Annual Computer Security Applications Conference* (Virtual Event, USA) *(ACSAC '21)*. Association for Computing Machinery, New York, NY, USA, 720–731.

[76] Cui Zhao, Zhenjiang Li, Han Ding, Wei Xi, Ge Wang, and Jizhong Zhao. 2021. Anti-Spoofing Voice Commands: A Generic Wireless Assisted Design. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 5, 3, Article 139 (sep 2021), 22 pages.