future internet

Article

Conservative Interference Injection to Minimize Wi-Fi Sensing Privacy Risks and Bandwidth Loss

Aryan Sharma *, Haoming Wang, Deepak Mishra * and Aruna Seneviratne

School of Electrical Engineering and Telecommunications, University of New South Wales, Sydney, NSW 2052, Australia; haoming.wang@unsw.edu.au (H.W.); a.seneviratne@unsw.edu.au (A.S.)
* Correspondence: aryan.sharma@unsw.edu.au (A.S.); d.mishra@unsw.edu.au (D.M.)

Abstract: With the impending integration of sensing capabilities into new wireless standards such as 6G and 802.11 bf, there is a growing threat to public privacy. Recent studies have revealed that even small-scale activities, like keyboard typing, can be sensed by attackers using Wi-Fi Channel State Information (CSI) as these devices become more common in commercial spaces. This paper aims to model the minimum CSI data rate required to sense activities in the channel and quantifies the detection accuracy of WiFi-based keystroke recognition in relation to the CSI sensing data rate. Our experimental findings using commercial-off-the-shelf hardware suggest that interference can be used as a defence strategy to degrade the CSI data rate and prevent undesirable Wi-Fi sensing attacks. To achieve a reduced data rate, we propose an extension to Bianchi's model of CSMA/CA systems and establish a new mathematical relationship between channel contention and the available CSI. This proposed relationship was empirically verified, and our contention-based defence strategy was experimentally validated. Experiments show that our contention-based defence strategy increases the chances of evading undesired WiFi-based keystroke recognition by around 70%. By leveraging prior work that shows a degradation in CSI quality with lower transmission rates, we show that conservative interference injection can sufficiently reduce sensing accuracy whilst maintaining channel bandwidth.

Keywords: Wi-Fi sensing; privacy; machine learning; IoT

check for updates

Academic Editors: Christos Tryfonopoulos and Nicholas Kolokotronis

Received: 28 November 2024
Revised: 31 December 2024
Accepted: 3 January 2025
Published: 6 January 2025

Citation: Sharma, A.; Wang, H.; Mishra, D.; Seneviratne, A. Conservative Interference Injection to Minimize Wi-Fi Sensing Privacy Risks and Bandwidth Loss. *Future Internet* **2025**, *17*, 20. https://doi.org/10.3390/fi17010020

1. Introduction

1.1. Background

As low-cost Internet of Things (IoT) devices become increasingly pervasive, their coordination into sensor networks has enabled new applications in industrial and home settings. To this end, Wi-Fi is an essential IoT enabler, facilitating communication between these coordinated devices to allow for data sharing and user interfacing. However, more recently, a new paradigm in wireless sensing has demonstrated that the Wi-Fi signals themselves can be used to sense the physical environment [1,2]. It has been shown that artificial intelligence (AI) and machine learning (ML) can be applied to analyse Wi-Fi Channel State Information (CSI) in order to monitor environmental occupancy, movement and the presence of intruders [1]. These applications of Wi-Fi sensing have been extended towards more private human features such as hand movements [2] or keyboard inputs [3,4]. This is creating new security, privacy and trust issues in IoT networks as our cyber information is at risk. To restore the privacy of IoT users, there is growing attention towards the development of defence strategies to prevent unwanted Wi-Fi sensing [5–8]. In this endeavour, the use of adversarial interference has been beneficial for reducing sensing accuracy and restoring user privacy as shown in Figure 1. Here, interference signals are injected (green) in order to impede the sensing signals (red). Unfortunately, the existing literature in this domain uses very high interference rates, which is a significant burden to the user in terms of bandwidth loss. Thus, there is a need for further work on interference-based defence techniques in order to improve their spectral efficiency.

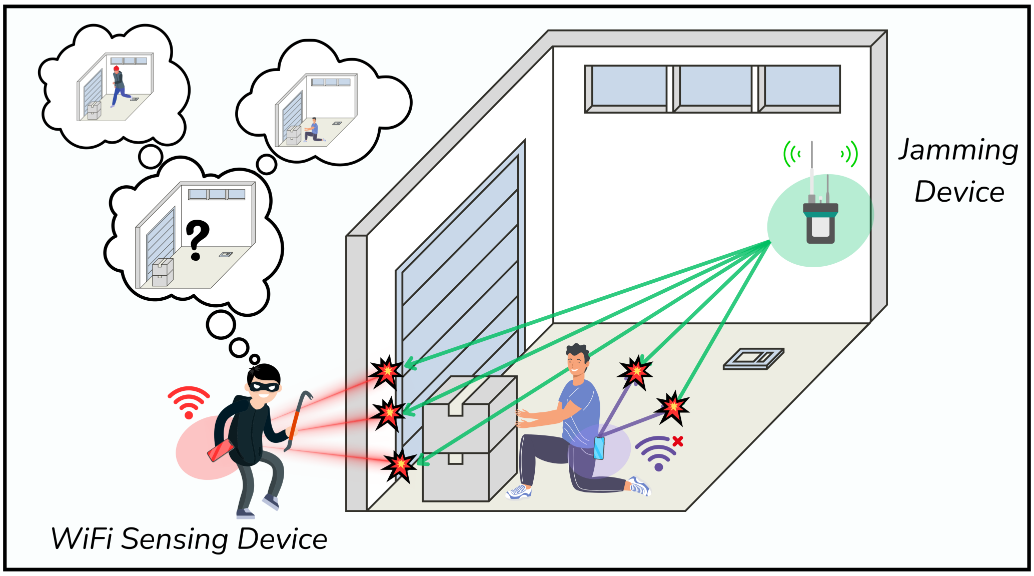

Figure 1. Interference topology.

1.2. Motivation and Contributions

Prior work on Wi-Fi sensing has demonstrated a growing threat to everyday civilians, as their gestures and personal information can be monitored by malicious attackers armed only with commercial Wi-Fi hardware. Hence, we are motivated to develop protection methods to thwart attempts to monitor unwilling patrons. We have identified some prior works that show that a brute forced jamming mechanism can diminish the attacker's threat; however, this approach entails severe degradation of the wireless communications link. This paper first experimentally validates the threat of human gesture recognition attacks with Wi-Fi sensing. We then present novel insights into a contention-based defence mechanism for thwarting said attacks. We make the following contributions:

1. In this study, we replicated keystroke recognition attacks from previous research and discovered new information on the minimum CSI data rates needed to identify and categorize keystrokes through Wi-Fi sensing.
2. After verifying that the accuracy of Wi-Fi sensing drops as a function of the CSI data rate, we extend Bianchi's [9] model of saturated CSMA/CA systems to create a mathematical relationship between the channel contention and the accessible CSI data rate.
3. We conducted an experimental validation of our mathematical model that describes the available CSI data rate in relation to the level of contention. Our findings show that as more devices access the Wi-Fi channel, the CSI data rate decreases, which implies that we may not have sufficient information to detect keystrokes.
4. Using this contention-based drop in the CSI rate, we motivate a defence strategy against Wi-Fi-based keystroke inference attacks. We achieve this by injecting controlled interference through another device that transmits ping packets to reduce the CSI sensing rate below the required threshold for accurate sensing.
5. We present a trade-off study that details the consequences of such contention-based defence strategies on the available Wi-Fi bandwidth for normal users.

The paper is structured as follows. In Section 2, we summarise state-of-the-art Wi-Fi sensing and its defence strategies. In Section 3, we provide a more detailed description of the Wi-Fi sensing system model and the importance of data rates for the system accuracy. We subsequently demonstrate a working test-bed of Wi-Fi-based keystroke detection in Section 4 and experimentally highlight the importance of CSI data rates. In Section 5, we propose a model to exploit Wi-Fi contention protocols to reduce CSI data rates in defensive endeavours. This contention model is then applied in Section 6 to thwart the accuracy of our keystroke detection test-bed. Finally, in Section 7, we provide insights on the trade-off between security improvement and channel bandwidth.

2. State of the Art

2.1. Channel State Information

Wi-Fi sensing is based on the propagation of radio waves through physical environments. As these signals travel from a Transmitter (Tx) to a Receiver (Rx), they encounter physical obstacles such as the air, walls, people and objects [10]. As the signal encounters these obstacles that cause reflection, refraction, diffraction and scattering, it splits into multiple paths that each arrive at the Rx at varying times (i.e., phase shift) with varying power (i.e., amplitude). This phenomenon is otherwise known as the multipath effect, and the overall multipath profile of a channel can be concisely characterised by a complex-valued metric known as Channel State Information (CSI) [1]. Due to the Orthogonal Frequency Division Multiplexing (OFDM) transmission scheme in 802.11n and subsequent Wi-Fi standards, the signal is modulated over a large domain of frequencies known as subcarriers [1]. CSI can be collected independently for each OFDM subcarrier, allowing us to analyse the propagation characteristics of the wireless environment as a function of frequency. In the same manner that varying wavelengths of visible light illuminate different objects, it has been shown that the CSI of different subcarriers can have utility for sensing different physical phenomena [11]. Tools such as Nexmon [12] sample CSI for all OFDM subcarriers on every incoming Wi-Fi packet. Hence, the data rate of CSI systems is bounded by the maximum number of packets they can propagate in the channel. CSI is then used to infer information about the channel.

2.2. Large-Scale Applications

Macroscopic changes in the channel, such as the introduction of new objects, cause large-scale changes to the multipath propagation characteristics of an environment. This is reflected in CSI, which can have large changes in the amplitude level or fluctuations over a captured time series [1,13]. To correlate changes in CSI with the environment, the CSI is first filtered to remove noise. Common filtering approaches include Hampel filters to remove outliers [1] and lowpass filters to smooth the CSI [14]. Following this, features can be calculated to correlate with changes in the physical environment. In applications such as object detection [11,15], the CSI is analysed on a frame-by-frame basis since the sensing subject is not time-varying. Hence, statistical features calculated on the set of CSI amplitudes have been used in conjunction with Support Vector Machines (SVMs) to predict the presence of different people or objects. Beyond this, large-scale changes in a physical environment such as occupancy levels have also been detected using absolute CSI amplitude levels. In addition to this, some large-scale sensing applications have used more complex temporal features calculated over the sampled time series of CSI. This includes feature calculation techniques such as Principle Component Analysis (PCA) [10] and Wavelet Transforms [1,10], which allow temporal variations to be analysed and subsequently fed into supervised machine learning (ML) algorithms to predict behaviours such as falling, walking and waving [16].

2.3. Fine-Grained Gesture Recognition

More recent work has applied advanced signal processing techniques to sense microscopic phenomena in the channel. To detect smaller changes in the channel, such as the movement of an arm [17], prominent works compute advanced features such as power distributions [18] or velocity distributions [17]. These higher-order features are typically applied with more complex learning algorithms such as Convolutional Neural Networks (CNNs) [19] or Long Short-Term Memory (LSTM) classifiers [1]. Of particular interest to us is the application of human keystroke recognition [3]. WiKey and other gesture recognition algorithms use moving window approaches to identify the start and end of some gesture event [3]. The identified windows contain waveforms of CSI amplitudes, which vary based on the gesture (i.e., a key press), and their algorithm subsequently uses Discrete Wavelet Transforms (DWTs) and Dynamic Time Warping (DTW) to distinguish between different waveforms. This approach is similar to generalised human activity recognition, which also seeks to classify based on the frequency composition of CSI [20].

2.4. Protection Methods

As evidenced by the success of prior work in fine-grained gesture recognition, there is a growing need for methods that secure privacy in the presence of Wi-Fi networks. Prior work on thwarting Wi-Fi sensing has introduced jamming signals into the wireless channel [5–7,21], causing a degradation in the sensing accuracy. In [5], interference was introduced on adjacent Wi-Fi channels in order to create competition for channel resources and reduce the CSI available to the sensing system. This subsequently caused a degradation in the sensing accuracy. Then, in [7], it was shown that the interference can be introduced in a time-varying manner in order to bias the measured CSI towards certain activity classes, thereby intentionally misleading the sensing system. This approach requires prior knowledge of the sensing systems ML parameters, which is unlikely. Thus, we focus on untargeted interference in order to degrade sensing accuracy. One aspect of interference injection that is overlooked is its impact on channel bandwidth. To the best of our knowledge, no prior work investigates the trade-off between improved security and loss in bandwidth due to interference injection. Thus, there is scope for more work to understand how the amount of interference can be chosen intelligently to preserve the communications channel for normal users.

3. System Description

In this section, we describe the working principles of Wi-Fi sensing and its dependence on the underlying data rate.

3.1. Wi-Fi Sensing System

As shown in Figure 2, Wi-Fi sensing is predicated on the propagation of signals (red) through a physical environment. As the signal encounters physical entities, its power and phase are altered, causing a disparity in the initial signal and that which is received by Rx. Where the physical entities are moving, such as the finger movements in Figure 2, the artefacts introduced in the wireless signal are also time varying as shown by the spikes in the signal (red) received by Rx. The relationship between the transmitted and received signals can be formally expressed as follows:

$$Y(f,t) = H(f,t)X(f,t) + N(f,t) \tag{1}$$

where the signal Y is received in continuous time t for all OFDM subcarrier frequencies f. This received signal is equal to the transmitted signal X after it incurs the multipath transformation, represented by H (CSI). The Gaussian white noise in CSI is represented by

N. Using this relationship, tools such as Nexmon [12] are able to take 'snapshots' of the CSI data (H) by inspecting the pilot bits of all received packets. The CSI data can be represented as $H[f,k]$ where $t = kT_s$. Here, we measure the CSI at k locations with a separation of T_s along the originally continuous domain t. The frequency domain representation of H is thus the following:

$$H[f,n] = \sum_{k=0}^{N-1} H[f,k]e^{-j\frac{2\pi}{N}nk} \tag{2}$$

where for all subcarriers f, $H[f,n]$ is the Discrete Fourier Transform (DFT) of the sequence $H[f,k]$. Equation (2) shows that the DFT representation of the underlying CSI is inherently dependent on the number of CSI snapshots used to compute the DFT, N, and the discrete domain variable k, which is itself dependent on the sampling period T_s. As described in Section 2, existing techniques in Wi-Fi sensing rely heavily on frequency analysis of CSI to discern gestures and movements of different speeds. Hence, we conclude that Wi-Fi sensing ML algorithms are highly dependent on the CSI sampling rate.

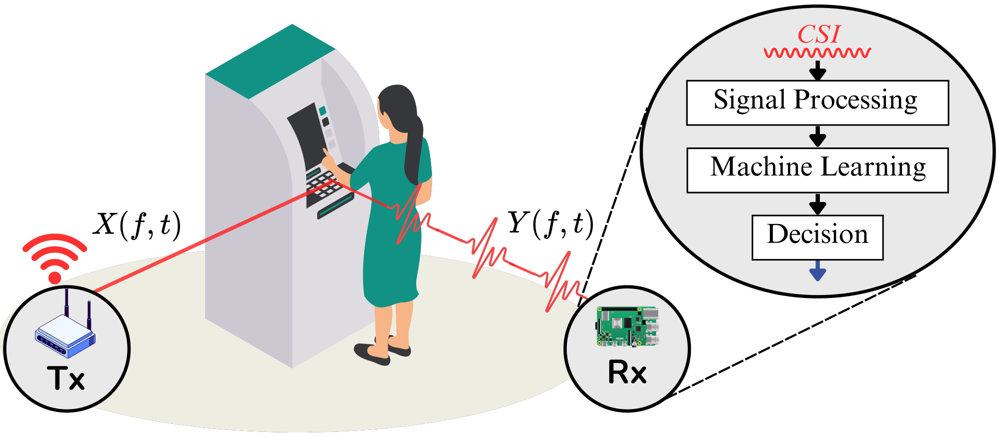

Figure 2. Wi-Fi sensing system overview.

3.2. CSI Sensing Rate and Coherence Time

Beyond the algorithmic dependence on CSI data rates, it is important to note that Wi-Fi sensing systems require a minimum data rate to effectively understand the changing wireless channel. To better understand this, we use the concept of channel coherence time. In order to sense activity in the channel, the CSI data must be captured faster than the rate of change of the channel. The time for which the channel is considered unchanging is known as coherence time, and it can be expressed in 2.4 GHz wireless channels as follows [22]:

$$T_{\text{coherence}} = \frac{9}{16\pi\frac{v}{\lambda}} \tag{3}$$

where v is the speed of movement in the channel (the activity being sensed), and λ is the signal wavelength (0.12 m for 2.4 GHz Wi-Fi). The required CSI data rate is then the reciprocal of the coherence time, and we compute this for several common sensing applications in Table 1.

Table 1. Approximated data rates for various sensing applications.

Activity	Speed (m/s)	$T_{\text{coherence}}$ (s)	Req. CSI Rate (Hz)
Walking	1.5	0.016	63
Running	2.7	0.008	129
Typing	4	0.005	189
Occupancy	0.7	0.031	33

In summary, the accuracy of Wi-Fi sensing is influenced by the CSI data rate, which is limited by the coherence time of the activity being sensed. Additionally, the hyperparameters of the ML algorithm are governed by the data rate for training. Therefore, any fluctuations in the rate of CSI capture will negatively impact the system's performance. To prove this assertion and determine the necessary CSI data rate, we will be conducting experiments on Wi-Fi-based keystroke recognition for varying transmission rates in the following section.

4. Keystroke Recognition Using Wi-Fi Sensing

In this section, we implement keystroke recognition using a state-of-the-art algorithm [3], before performing a subsequent experiment to show its reliance on the underlying data rate.

4.1. Keystroke Recognition Model

It has been shown in prior work [3] that a Wi-Fi sensing apparatus can be used to detect which key is pressed on a keyboard by computing features on CSI time series data. This flavour of the attack is particularly concerning since it extends the domain of privacy risks beyond physical attributes towards cyber information that may be entered through a computer keyboard. We implemented the algorithm initially proposed by WiKey as demonstrated by Figure 3 below.

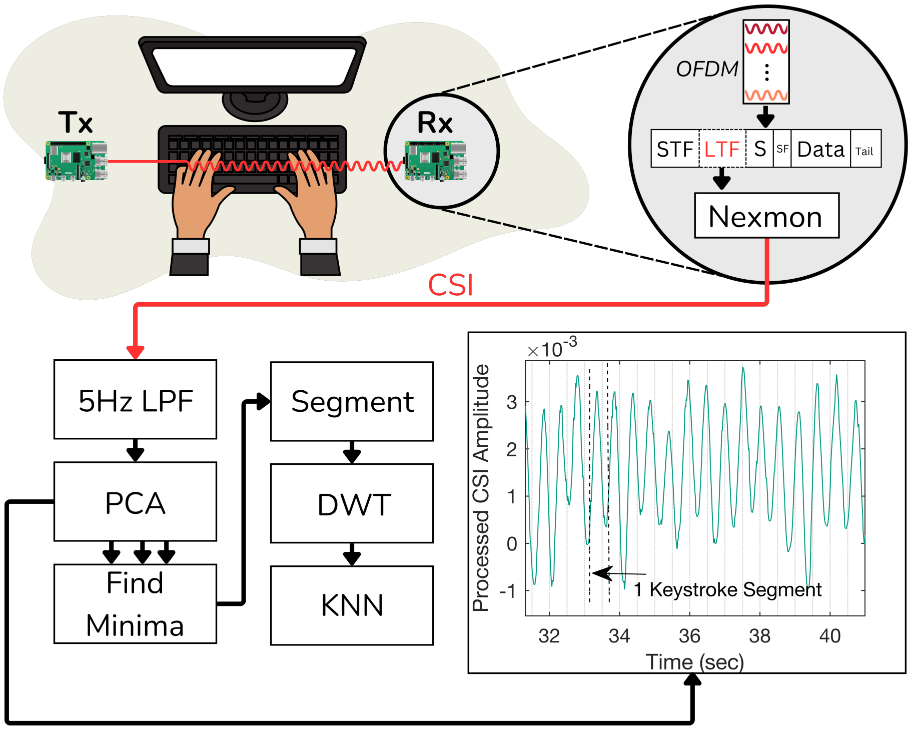

Figure 3. WiFi-based keystroke detection model.

In Figure 3, we illustrate the algorithm for CSI-based keystroke recognition. As before, Tx propagates a Wi-Fi signal (red) which incurs time-varying oscillations due to the typing hand. This time varying Wi-Fi signal is then received by Rx. The CSI time series is first processed with a lowpass filter to remove noise above the cutoff frequency of 5 Hz. Then, the 56 time series (one for each of the 56 OFDM subcarriers) are collectively processed via PCA. The 2nd, 3rd and 4th PCA vectors are used in parallel for all subsequent computations.

Encapsulated within Figure 3, we present a plot of a CSI time series after lowpass filtering and PCA. The CSI data was captured whilst a human repeatedly pressed the 'Q' key on a keyboard. The x-axis represents time, and the y-axis represents the amplitude of the processed CSI. We observe clearly that the CSI amplitude varies in sync with the repeated keystrokes. To extract this keystroke information, we find the local minima in the time series to isolate each of the 'crevices' at the end of a keystroke. The samples between these minima are then considered to be keystrokes, and we perform segmentation of the time series to extract the detected keystrokes. Each segmented keystroke is then subject to the DWT, which characterises the shape of the keystroke segment by a frequency decomposition using Daubechies Wavelet 4. These shape features are then used as inputs to 3 KNN's, which utilise DTW distances to calculate the distance between points and their neighbours. The decisions from the 3 KNN algorithms are then multiplexed in order to make an overall classification of the keystroke.

4.2. Experimental Setup

In a standard office building within the range of commercially operating Wi-Fi networks, we setup our experimental apparatus as depicted below.

In Figure 4, we depict the experimental setup for human keystroke recognition using Wi-Fi CSI. We positioned a Wi-Fi Transmitter (Tx) on one side of a standard keyboard, which broadcasts ping packets at a rate of 1300 Hz. On the other side, we placed a Raspberry Pi 4B Receiver (Rx) equipped with the Nexmon CSI extraction firmware [12]. This is denoted by the red circle in Figure 4. This device is manufactured by Raspberry Pi Ltd. (Cambridge, UK), and was sourced online within Australia via. Element 14. The Nexmon firmware activates monitor mode on the Raspberry Pi 4B Network Interface Card (NIC) and then captures CSI for every received packet. Given that Tx is propagating packets at 1300 Hz, Rx will capture 1300 CSI snapshots per second. All communications occur on channel 2 of the 2.4 GHz Wi-Fi spectrum, with a bandwidth of 20 MHz and CSI sampled for 56 subcarriers. The experimental subject (pictured) pressed the set of keys Q, Z, H, M and P a total of 100 times in the training phase and 50 times in the testing phase. Without loss of generality, we chose this subset of keys since they are evenly spread across the keyboard and allow for simpler analysis compared to a full keyboard.

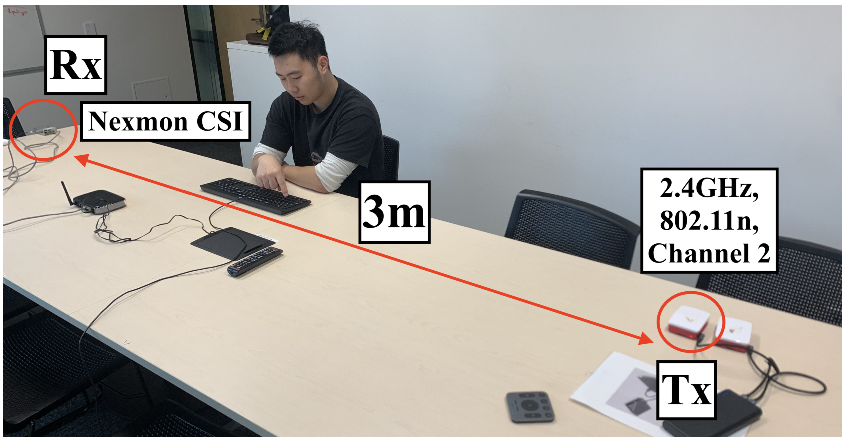

Figure 4. Keystroke recognition experimental setup.

4.3. Results

The experimental data was passed into the processing framework described above, and we obtained the following result.

In Figure 5, we present a confusion matrix of the ML algorithm when evaluated with our experimental data. The rows represent true classes, and the columns represent predicted classes. Hence, the leading diagonal represents the true positive rates for each key, with darker colors corresponding to higher rates. We observe that the classifier has an overall true positive rate of 84.96%. The largest error occurs when classifying the H key being pressed, which is incorrectly predicted as M 19.20% of the time. The misclassification between these two classes can be explained by observing the relative proximity of the M and H keys on a keyboard. Furthermore, the worst performing class overall is the H key, with a true positive rate of 71%. This once more reflects the position of the H key, in between all the other keys. Overall, this result supports those achieved in prior work [3], demonstrating the threat posed to user keystroke information by Wi-Fi sensing.

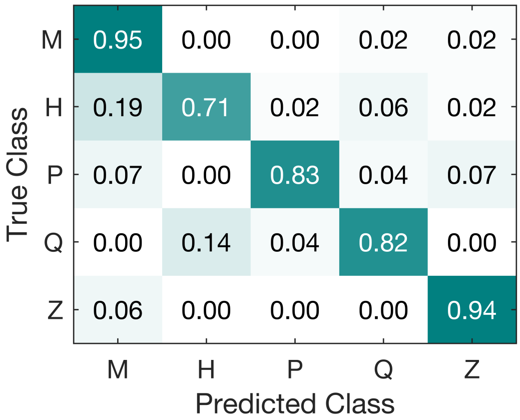

Figure 5. Classification accuracy.

4.4. Minimum CSI Sensing Rate for Reliable Sensing

The result in the previous section was achieved with a CSI data rate of 1300 Hz. To illustrate the importance of this rate, we perform a subsequent experiment in which we change the transmission frequency of the Wi-Fi sensing device (Tx). We setup the devices as in Figure 4 and asked the participant to press a key 50 times. This process is repeated for transmission frequencies varying between 1300 Hz and 20 Hz. We observe what percentage of the 50 keystrokes the system was able to detect from the captured CSI. This is presented in Figure 6.

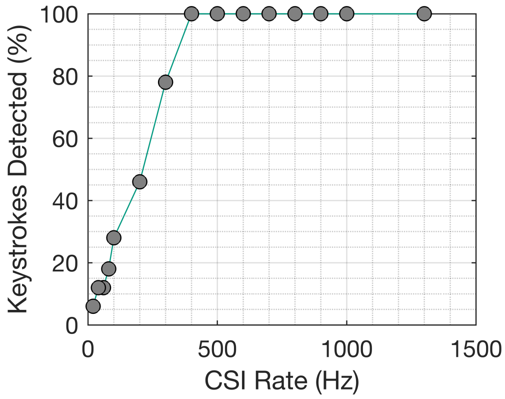

Figure 6. Impact of CSI Rate.

In Figure 6, we present a result that demonstrates the effect of the CSI data rate on the keystroke detection algorithm. The x-axis represents the data rate (Hz) of the sensing system, and the y-axis represents the percentage of 50 keystrokes that were detected from the CSI. Then, each plotted point represents an empirical result. Here, we note that *detection* refers to the ability of the system to extract a keystroke event from the CSI waveform, *not* the classification accuracy.

We observe that CSI data rates above 400 Hz are able to detect 100% of the keystrokes. Below this, we see a clear negative trend in the keystroke detection rate as a function of the

CSI data rate. A key point of interest is 200 Hz, below which only 47% of the keystrokes can be detected by the system. This corroborates our analytical finding in Table 1, where the critical data rate for keystroke recognition was 189 Hz. Given that we are discussing the ability to *detect* keystrokes, these results do not reflect ML performance or the reliance of feature calculation schemes on the sampling rate. Instead, this result demonstrates the importance of dense sampling to reconstruct human movement in sampled CSI time series. This issue is akin to the Nyquist–Shannon sampling problem for sampling any time varying information. Overall, the results in this section demonstrate the importance of the underlying CSI data rate for Wi-Fi sensing systems.

5. Proposed Contention Model for Defence Against Wi-Fi Sensing

The previous section showed empirically that the efficacy of WiFi-based keystroke recognition attacks is predicated on the underlying CSI data rate. In this section, we theorise how the CSI data rate can be reduced by intentionally injecting contention into the Wi-Fi channel. In this way, we formulate a contention-based model for preventing undesirable Wi-Fi sensing applications and safeguarding our privacy.

5.1. CSMA/CA in 802.11 Networks

802.11 networks are designed with Medium Access Control (MAC) protocols such as Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) to handle a large amount of devices attempting to access the wireless channel resources simultaneously [23]. CSMA/CA requires stations to sense the channel to ensure that it is idle prior to attempting any transmission [24]. When a station is ready to send a packet, it listens to the channel for a period known as the Distributed Inter-Frame Space (DIFS) interval [24]. If the channel remains idle throughout the entire DIFS interval, the station backs off for a randomly chosen number of discrete time slots. This random back-off duration is uniformly distributed between $(0, W-1]$, where W is a variable known as the contention window [23–25]. In each slot where the station senses an empty channel, it decreases the back-off counter. Once the back-off counter reaches zero, the station can attempt to transmit again using the wireless medium. If a collision is detected, the station again randomly chooses a back-off interval, this time choosing an integer uniformly distributed between $(0, 2W-1]$, and the process repeats. The upper limit iteratively increases as a function of the number of collisions (m) [25] by $2^m W - 1$. This is limited to a maximum ceiling of $2^{m_{\max}} W - 1$, where $m_{\max}$ is the maximum number of collisions allowed, which varies between Wi-Fi hardware.

5.2. Throughput of CSMA/CA WLANs

The CSMA/CA process described above has been represented using a Markovian model in a prominent work by Bianchi around the time of its advent [9]. Subsequent work [26] has built upon Bianchi's model to express the collision probabilities of 802.11 stations. This can be boiled down to two important expressions:

$$p = 1 - (1-\pi)^{N-1} \tag{4}$$

$$\pi = \frac{2(1-2p)}{(1-2p)(W+1) + pW(1-(2p)^m)} \tag{5}$$

where π is the probability that a station will transmit during a slot, and then p is the probability that there is a collision in that slot. The probability π is considered to be equal for all stations, operating under the assumption that they are all attempting to perpetually transmit packets. This is formally referred to as the saturation condition by Bianchi [9]. These two expressions can be solved simultaneously for π and p as a function of the number of stations (N). We use the following set of parameters for 802.11n networks:

Subsequently, the parameters in Table 2 can be used to solve Equations (4) and (5) simultaneously. We solve for $N = [2, 25]$ devices and illustrate the collision probabilities below.

Table 2. 802.11n CSMA/CA parameters.

Parameter	Value
Slot Time	20 µs
DIFS	50 µs
Short Inter-Frame Space (SIFS)	10 µs
Acknowledgement (ACK) Time	304 µs
T_{Header}	400 µs
T_{Payload}	3.4 ms
Minimum Contention Window (W)	16 slots

In Figure 7, we illustrate the collision probability (p) and the transmission probability (π) as functions of the number of competing stations in 802.11n networks. The x-axis represents the number of 802.11n devices in the network, and the y-axis represents probabilities ranging from 0 to 1. Then, the blue curve represents the probability of a collision occurring in the channel and the red curve represents the probability that any station will choose to transmit at a given time.

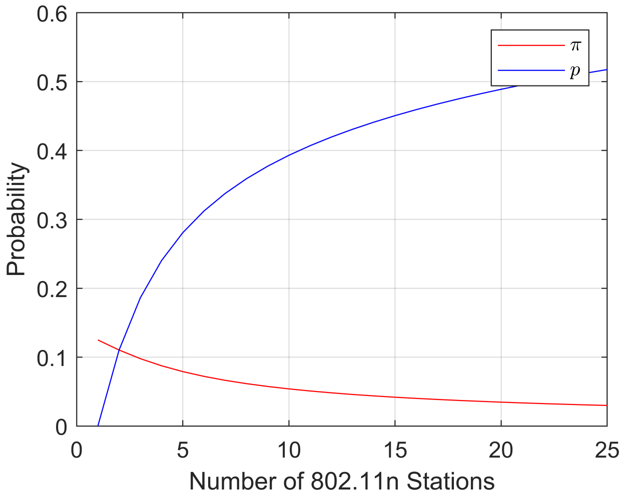

Figure 7. Collision probability of 802.11n networks with increasing users.

We observe that the probability of collisions within the channel grows logarithmically as more users enter the channel. This growth is particularly aggressive when the patronage increases from 2 to 5, with the probability of collision increasing from 0.11 to 0.28 (254% increase). We are subsequently interested in the effect this increased collision probability has on the overall throughput for all devices in the channel. Building on Equations (4) and (5), it has been further shown in prior work [26] that the throughput of a CSMA/CA network can be expressed as follows:

$$\tau = \frac{E[\text{Payload Transmitted in a slot}]}{E[\text{Duration of Slot Time}]}$$

$$\tau = \frac{P_{\text{s}} P_{\text{tr}} L}{P_{\text{s}} P_{\text{tr}} T_{\text{s}} + P_{\text{tr}}(1 - P_{\text{s}}) T_{\text{c}} + (1 - P_{\text{tr}}) T_{\text{id}}} \tag{6}$$

for a packet of length L and a slot time T_{id}. Here, the probability of transmission (P_{tr}) and probability of that transmission being successful (P_{s}) are further defined as follows:

$$P_{\text{tr}} = (1 - (1 - \pi)^N), P_s = \frac{N\pi(1 - \pi)^{N-1}}{1 - (1 - \pi)^N} \tag{7}$$

and the average transmission time (T_{s}) and average time spent in collision (T_{c}) are expressed as follows:

$$T_s = T_{\text{Header}} + T_{\text{Payload}} + \text{SIFS} + \sigma + \text{ACK} + \text{DIFS} + \sigma \tag{8}$$

$$T_c = T_{\text{Header}} + T_{\text{Payload}} + \text{DIFS} + \sigma \tag{9}$$

where σ is the propagation delay. We solve Equation (6) for N = $[1, 10]$ devices, and vary the length of the payload between 700 bits (roughly the size of an ICMP ping packet) to 1900 bits.

In Figure 8, we illustrate the analytical throughput for 802.11n networks. The x-axis represents the number of Wi-Fi stations in proximity, and the y-axis represents the throughput (bits/sec) of a station. We illustrate 5 curves corresponding to the devices propagating packets of varying lengths.

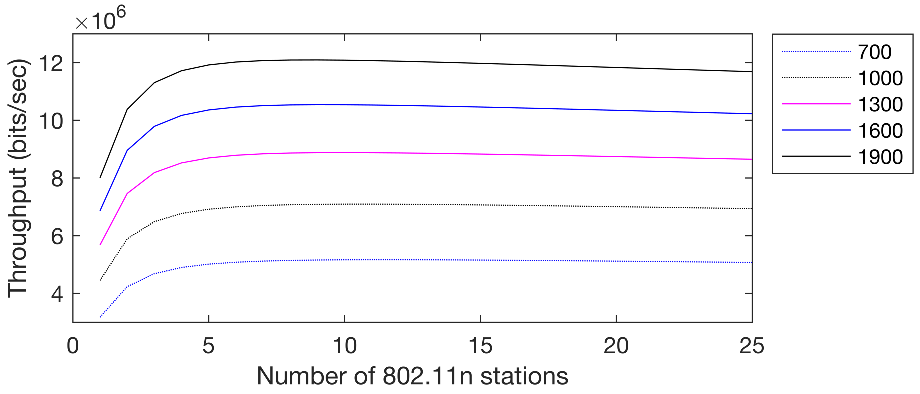

Figure 8. Throughput of 802.11n networks with increasing users and different packet lengths.

From the vertical spread of the various curves, we note that the throughput varies as a function of the packet size. This is symptomatic of the greater channel utilisation due to larger packet sizes since the stations spend less time in the inter-frame periods such as DIFS and SIFS. We further note that the decay in throughput appears to be independent of the size of the packets being transmitted. Overall, the total channel throughput decreases slightly as the number of stations increases, presumably due to the increased overhead of the distributed access protocol. This result represents the throughput in bits per second. We recall from Section 3 that CSI for Wi-Fi sensing is captured on a per-packet basis. Hence, we propose a new expression for the available CSI data rate as the quotient of throughput with the length of the transmitted packets, divided by the total number of stations as in Equation (10).

$$R_{\text{CSI}} = \frac{1}{N}\frac{\tau}{L} \tag{10}$$

We compute R_{CSI} for the throughputs derived earlier and plot the results below.

In Figure 9, we illustrate the CSI data rates for an 802.11n network with an increasing number of stations in the channel. The x-axis represents the number of stations transmitting in the channel, and the y-axis represents the effective CSI data rate of the channel. Each curve represents a simulation with packets of varying lengths. We note that a packet length of 700 bits corresponds to ICMP ping packets, which are commonly used for Wi-Fi sensing applications.

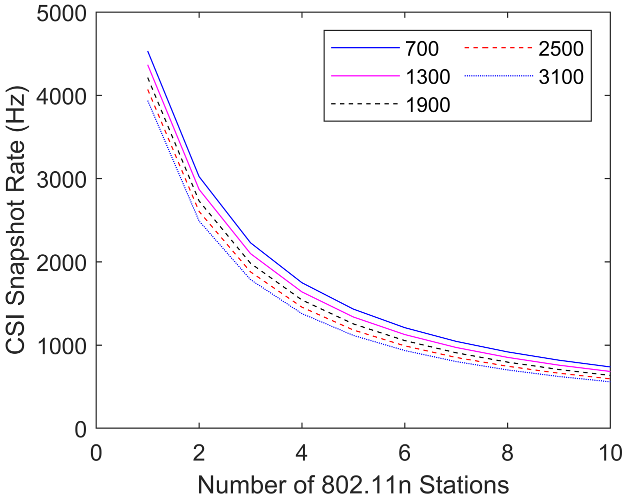

Figure 9. CSI snapshot rate under varying packet lengths.

We observe firstly that the CSI data rate reduces as the number of 802.11n users increases. This is initially a sharp decrease by a factor of 0.5 when the usage increases from one to two users, and the rate of decrease slows down subsequently. We further note that the CSI data rate is maximised for the smallest packet lengths, with the solid blue curve representing the highest CSI data rate for ping packets. This result reflects the smaller time spent occupying the channel per packet, and hence, more packets can use the channel resulting in a higher CSI data rate. Overall, the analytical results in this section support the notion that interference from other devices can reduce the effective CSI data rate by virtue of the 802.11n medium access control.

5.3. Empirical Validation

To validate our proposed theoretical model, we conduct an experiment to demonstrate how an increase in channel users causes a reduction in the available CSI data rate.

5.3.1. Experimental Setup

The experimental setup consists of a CSI sensing apparatus surrounded by interference devices as illustrated in Figure 10.

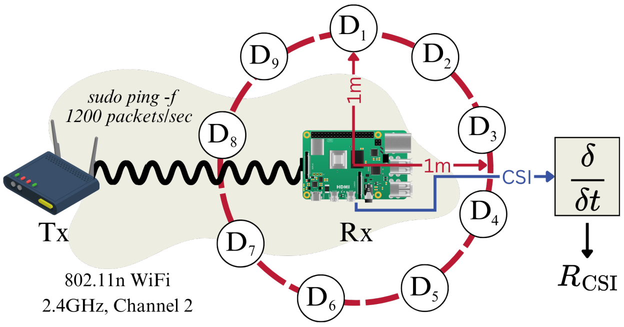

Figure 10. Experimental setup for validating analytical model.

As shown by the red arrows, the CSI sensing apparatus is surrounded by 10 interference devices at 1 m distance from Rx. In each trial, we collect CSI by transmitting Wi-Fi signals from Tx to Rx, as shown by the black wavy line, for a duration of 60 s, and we compute the average gradient ($\frac{d_{\mathrm{CSI}}}{d_t}$) of the time series to estimate the average CSI data rate. We first conduct a trial with none of the interference devices (D_n) operating, and we conduct 9 subsequent trials with each interference device iteratively switched on. When switched on, the interference devices propagate ping packets at the saturation rate using the `sudo ping -f` command.

5.3.2. Results

The results of the experiment are illustrated below.

In Figure 11, we present a plot of the CSI data rate that Rx was able to achieve with varying levels of contention in the channel. The x-axis represents the number of 802.11n stations sending ping packets, and the y-axis represents the average CSI data rate that Rx was able to achieve over the duration of 60 s. The blue line represents the analytical result from earlier, and the red line is the experimentally derived result.

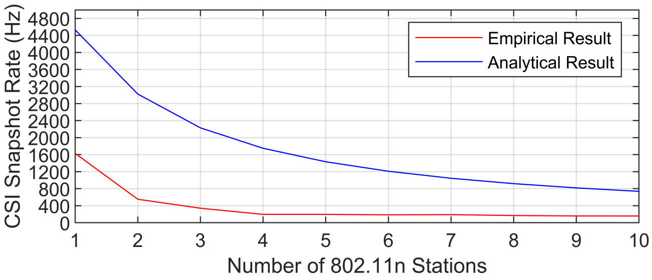

Figure 11. CSI snapshot rate with varying number of stations.

We observe a striking similarity in the shape of the empirical result with the analytical result. The CSI snapshot rate is initially 1200 Hz, and it decreases to less than 400 Hz with the addition of only a single interference device. Introducing subsequent devices causes further degradation in the achievable CSI snapshot rate to less than 200 Hz. Both curves have similar shapes, with the key difference being a vertical translation. This difference can be understood when considering the real-world limitations of 802.11 communications. Although the analytical result implies that a CSI data rate of 4400 Hz is achievable with a channel bit-rate of 72 Mbps, the actual bandwidth is limited by the computation power of the wireless devices and their Network Interface Cards (NICs). Overall, this result supports the theoretical model for contention-based defence against undesired Wi-Fi sensing attacks.

6. Application of Defence Model Against Keystroke Attacks

The previous section demonstrated analytically and empirically that channel contention can reduce the available CSI data rate. We utilise these ideas to develop a defence mechanism to thwart the efficacy of Wi-Fi-based keystroke recognition.

6.1. Experimental Setup

We utilise the same experimental apparatus as before, with the addition of an interference device as depicted in Figure 12.

The interference device is a Macbook Pro laptop on the same communications channel (Channel 2 of the 2.4 GHz Wi-Fi spectrum). It is configured to propagate ping packets at varying rates between 1300 Hz and 0 Hz, whilst the keyboard sensing apparatus operates with Tx propagating ping packets at 1300 Hz. As demonstrated by the ellipses in Figure 12, the proximity of these devices causes the range of their communications to overlap, resulting in competition for control over the channel resources. As illustrated by the wavy red signal in Figure 12, movement of fingers on the keyboard causes time varying changes in the Wi-Fi signal which are classified by the keystroke sensing system. To investigate how the varying interference packet rate influences the accuracy of the keystroke sensing system, we once more collected CSI data during 100 keystroke events for each of the letters Q, Z, H, P and M and used the model in Section 4 to train a sensing system. The system was then evaluated with a dataset of CSI collected during 50 keystrokes for each of the same keys. We collected 6 such sets of testing data, under varying rates of contention from the laptop.

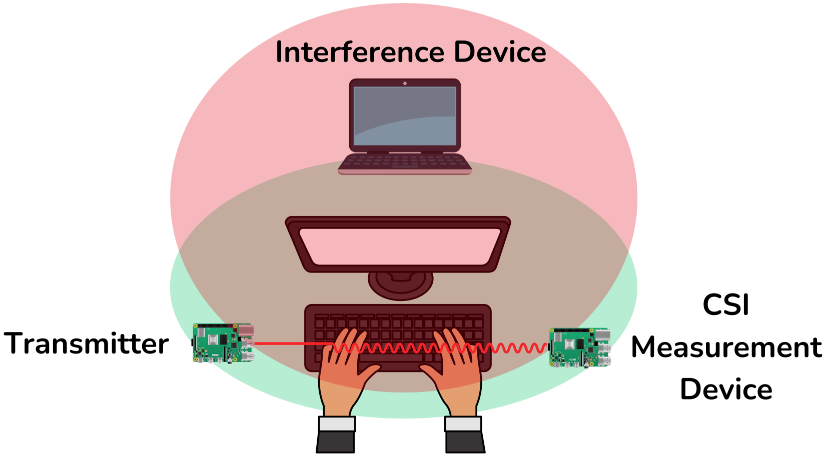

Figure 12. Setup for interference experiment.

6.2. Results

As the laptop begins to transmit ping packets, the captured CSI waveforms degrade, as demonstrated by Figures 13–17.

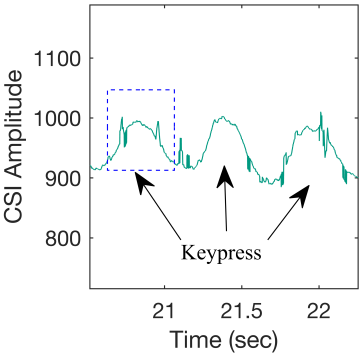

Figure 13. CSI time series with no interference.

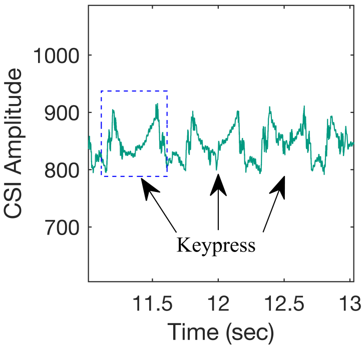

Figure 14. CSI time series with 50 Hz interference.

We observe firstly from Figure 13 that the keypress events are clearly visible in the sampled CSI as time-varying ripples, in line with our expectations as discussed in Section 3 and in line with our earlier results in Section 4. In each Figure, a single keystroke is outlined by the dotted box, and the arrows point towards the other observed keystrokes. Then, in Figures 14–17, there is an incremental degradation in the quality of the CSI and the clarity of the keystroke events. In Figures 14 and 15, we are still able to discern three clear keypress events; however, in Figures 16 and 17, this is no longer possible. This gradual degradation of the CSI time series further supports our idea that maximum interference rates are not necessary, and lower interference can be applied to sufficiently degrade sensing. To understand the impact of the degraded CSI on the sensing accuracy, we proceed with evaluating the sensing output of the keystroke recognition threat model using this dataset.

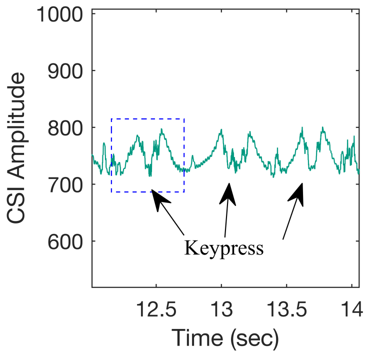

Figure 15. CSI time series with 250 Hz interference.

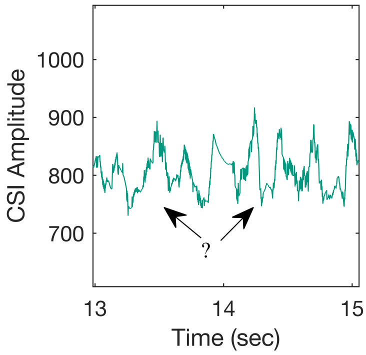

Figure 16. CSI time series with 500 Hz interference.

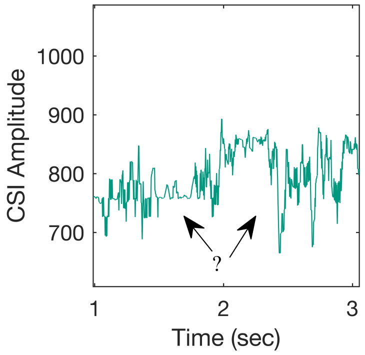

Figure 17. CSI time series with 1 kHz interference.

In Figure 18, we present the sensing accuracy against the interference packet rate. The x-axis represents the packet rate from the interfering laptop. The green left y-axis then represents the accuracy of the keystroke recognition system as it operates in the presence of said interference, and the blue right y-axis represents the CSI data rate under each level of interference. For each interference rate, the blue point thus denotes the achieved CSI data rate (Hz) and the green diamond denotes the achieved sensing accuracy. There is a clear negative correlation between the interference frequency and the sensing accuracy. With an initial interference rate of 50 Hz, the sensing accuracy drops from 85% to 70%. Beyond this, the sensing accuracy crosses below 50% at an interference rate of 500 Hz, corresponding to a CSI data rate of 700 Hz. This corroborates our observations from

Figures 13–17, where the keystrokes could no longer be clearly identified with interference rates of 500 Hz. Beyond this, the accuracy appears to decay in a decreasing manner asymptotically. The poor performance of the sensing system with a CSI Data Rate of 700 Hz due to the interference injected at 500 Hz is an interesting result. We recall the earlier result in Figure 6, where a CSI sensing rate of 700 Hz was able to yield a keystroke detection rate of 100%. Once more we reiterate that detection refers to the ability to pick up the keystroke event in the CSI time series, *not* the classification accuracy. Despite being able to pick up all the keystrokes, the sensing classification accuracy is below 50%. This result corroborates the ideas proposed in Section 3, where we postulated that the dependence of Wi-Fi sensing systems on frequency-based feature engineering techniques creates a susceptibility to changes in the CSI data rate. Even though all the keystrokes are picked up within CSI, the change in CSI sampling rate distorts the DWT features and hence reduces the accuracy of the sensing system. Overall, this experiment clearly shows that minimal interference can still be effectively used to defend against Wi-Fi sensing. To further visualise the improvement in privacy, we define the *Privacy Preserving Probability (PPP)* metric as the percentage probability of safeguarding a key from being detected by Wi-Fi sensing. This is calculated as the complement of the true positive detection rate for each key. We plot the PPP under several rates of defensive interference in Figure 19. The x-axis represents the five keys that the Wi-Fi sensing system is attempting to detect. Then, the height of each column represents the PPP with different levels of defensive interference. We observe that the PPP generally increases as a function of defensive interference. Generally, the PPP is higher for all keys when comparing 500 Hz interference with 0 Hz interference, and higher again when comparing 1500 Hz interference. This is congruent with our results in Figures 13–17, where increasing amounts of interference resulted in more degraded CSI waveforms. With 1500 Hz of interference, we can safeguard all keys with an overall PPP of 85%, an improvement of 70% over the baseline. Compared with prior work, the result with 500 Hz of interference demonstrates that lesser amounts of interference can considerably improve user privacy. One aspect of this result which is unclear is how to choose an appropriate level of interference. At the outset of this paper, we discussed prior work that belligerently injects dense packet streams to maximally thwart CSI. In lieu of this, our goal is to be more conservative in our application of interference to sufficiently impede sensing accuracy without overconsuming channel resources.

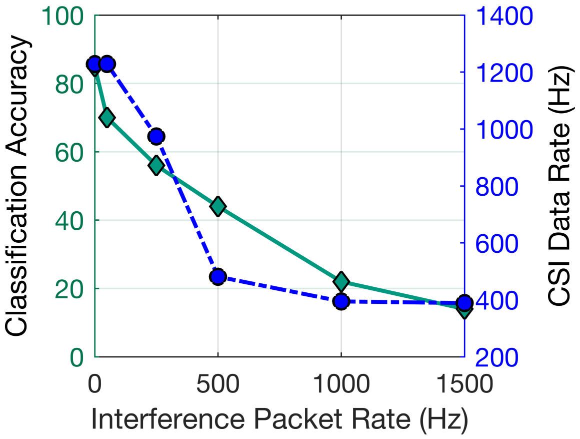

Figure 18. Effect of interference on Wi-Fi sensing accuracy.

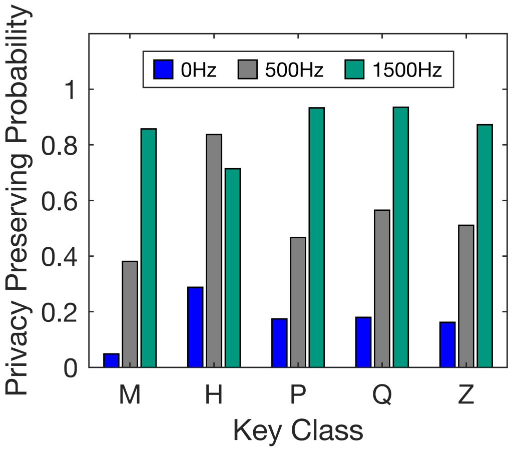

Figure 19. Security rate for each key as interference increases.

7. Tradeoff Between Privacy and Utility

The previous section showed that increasing amounts of interference increasingly degrade Wi-Fi sensing accuracy. This poses a question for IoT users who intend to use interference to fend off attackers, as they need to determine the optimal amount of interference that degrades attack accuracy while still allowing normal IoT communications. This section accordingly presents experiments to determine how the level of interference affects the usable Wi-Fi bandwidth (BW).

7.1. Experimental Setup

To evaluate the effect of interference defence on Wi-Fi bandwidth, we introduce a server–client pair to the experimental environment. Utilising the same setup illustrated in Figure 12, we position a Transmitter (Tx) in the channel, which propagates ping packets at a rate of 1000 Hz to a CSI measurement device. Meanwhile, an interference device is positioned adjacent to Tx. The independent variable is the rate of interference packet propagation, which we vary between 0 Hz and 1000 Hz. Concurrently, the server-client pair on the same Wi-Fi channel (2.4 GHz, Channel 2) transfers a 200 MB file over the wireless link. The dependent variable is the time taken for this file to transfer from the client to the server, which we subsequently convert to a bandwidth in Mbps.

7.2. Empirical Results

In this section, we compare the communications throughput under each interference level.

Figures 20 and 21 demonstrate the effect of interference on channel throughput. In Figure 20, the Wi-Fi throughput is plotted against the interference rates due to injected ping packets. A clear negative correlation between interference rate and Wi-Fi throughput is observed, as the linear fit has an RMSE of 0.4318 and R^2 of 0.952. Overall, an interference rate of 1000 Hz causes a large 25% reduction in throughput for the client–server pair. This once more motivates the use of more conservative interference injection to minimise throughput loss. The linearity of this trend is an interesting result, and can be compared against Figure 11. In Figure 11, the achieved CSI rate, which is proportional to throughput as per Equation (10), approaches an asymptote with the introduction of more devices. Due to hardware constraints on the Raspberry Pi 4B, we were unable to investigate whether the linear trend in Figure 20 approaches an asymptote with further increases in the transmission rate. We motivate future work to investigate this, when commodity devices are able to propagate more dense packet streams. To visualise the trade-off between the security and channel utility, Figure 21 plots the *PPP gain* against the measured throughput loss.

Here, the *PPP gain* is the percentage increase in the PPP over the baseline scenario with no interference. We observe a clear trade-off between the PPP gain and the available throughput. To achieve a PPP gain of 60%, we sacrificed nearly 25% of the available Wi-Fi throughput. Initially, we observe that a 20% gain in the PPP can be achieved with only 4% loss of throughput. This corresponds to the case of interference at a rate of 200 Hz. Referring back to Figure 18, this conservative interference rate is sufficient to reduce the keystroke sensing accuracy below 60%. The trade-off becomes more severe beyond this, as shown by the increasing slope of the curve in Figure 21. A further 20% gain in the PPP (from 20% to 40%) entails a much larger throughput loss of 11%. These results demonstrate the effectiveness of conservative interference injection in significantly mitigating sensing threats, while having minimal impact on Wi-Fi throughput.

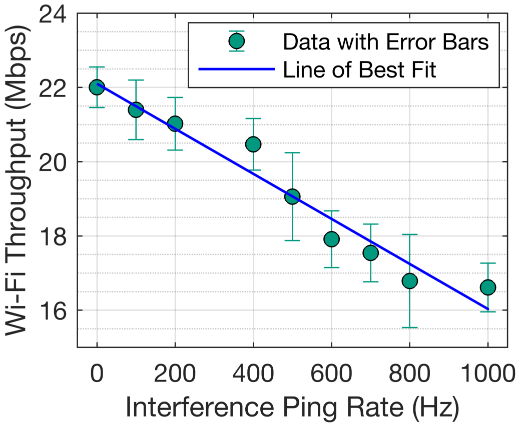

Figure 20. BW vs. interference.

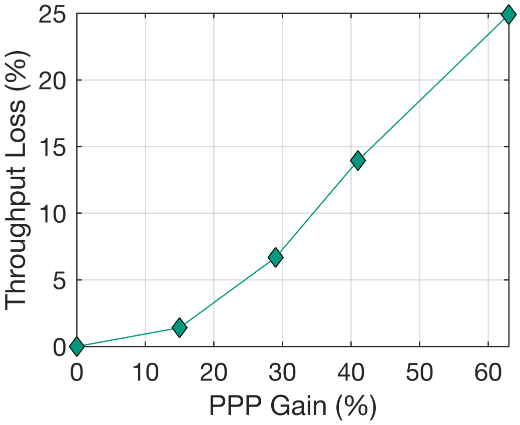

Figure 21. Privacy gains vs. BW.

8. Concluding Remarks

This paper introduces a conservative interference injection strategy to counteract privacy and security threats from Wi-Fi sensing. Through a survey of prior work in Wi-Fi sensing as well as the construction of our own testbed, we demonstrate that fine-grained human activities are under threat of being monitored with high accuracy. By experimental analysis using commodity devices, we demonstrate that this high accuracy of Wi-Fi sensing is highly dependent on the CSI data rate. We subsequently showed analytically and experimentally that this CSI data rate can be intentionally disrupted by other devices in the wireless channel. This principle is the basis of prior work in the state-of-the-art that utilises wireless interference to reduce the CSI rate and diminish sensing accuracy, thereby protecting user privacy. This paper further investigates this, revealing a positive correlation between the degree of interference introduced and the degradation in sensing accuracy. This establishes a trade-off between achieving increased privacy and maintaining acceptable channel throughput for other Wi-Fi users. Compared to state-of-the-art approaches, we can thwart sensing applications with comparable efficacy at a fraction of the cost to channel bandwidth by injecting interference conservatively. Although our approach is applied specifically to keystroke recognition, the results are generalizable to other Wi-Fi-based gesture recognition systems, where interference can degrade the resolution of environmental movement. We suggest that future work should investigate the suitability of this conservative interference injection scheme in commercial settings under interference from additional devices. In such environments, the consumption of channel resources by other devices will create uncertainty around our ability to finely control the CSI rate of a sensing system. Furthermore, the increased environmental noise and poor Signal-to-Noise Ratio (SNR) of our interference devices may impede our ability to generate strong contention in the channel. Finally, whilst the methods and results in this paper are generalisable across Wi-Fi sensing applications, we motivate future work to apply it against coarse-grained sensing applications such as movement recognition. Future work should investigate whether these coarse-grained sensing applications, with relatively simple ML decision algorithms, are as sensitive to the CSI data rate and adversarial interference.

Author Contributions: Conceptualization, All; methodology, All; software, A.S. (Aryan Sharma) and H.W.; validation, A.S. (Aryan Sharma) and D.M.; formal analysis, All; investigation, All; resources, All; data curation, A.S. (Aryan Sharma) and H.W.; writing—original draft preparation, A.S. (Aryan Sharma) and H.W. and D.M.; writing—review and editing, D.M. and A.S. (Aruna Seneviratne); visualization, All; supervision, D.M. and A.S. (Aruna Seneviratne); project administration, D.M. and A.S. (Aruna Seneviratne); funding acquisition, D.M. and A.S. (Aruna Seneviratne). All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in parts by the Australian Research Council's Discovery Early Career Researcher Award (DECRA) under Grant number DE230101391 and by the Australian National Intelligence and Security Discovery Research Grants (NISDRG) under Grant number NI240100147. Aryan Sharma's participation is supported by the Cyber Security Cooperative Research Centre Limited, whose activities are partially funded by the Australian Government's Cooperative Research Centres Programme.

Data Availability Statement: The data that support the findings of this study are available on request from the corresponding author. The data are not publicly available due to privacy or ethical restrictions.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Chen, C.; Zhou, G.; Lin, Y. Cross-Domain WiFi Sensing with Channel State Information: A Survey. *ACM Comput. Surv.* **2023**, *55*, 1–37. [CrossRef]
2. Yang, M.; Zhu, H.; Zhu, R.; Wu, F.; Yin, L.; Yang, Y. WiTransformer: A Novel Robust Gesture Recognition Sensing Model with WiFi. *Sensors* **2023**, *23*, 2612. [CrossRef] [PubMed]
3. Ali, K.; Liu, A.X.; Wang, W.; Shahzad, M. Recognizing Keystrokes Using WiFi Devices. *IEEE J. Sel. Areas Commun.* **2017**, *35*, 1175–1190. [CrossRef]
4. Li, M.; Meng, Y.; Liu, J.; Zhu, H.; Liang, X.; Liu, Y.; Ruan, N. When CSI Meets Public WiFi: Inferring Your Mobile Phone Password via WiFi Signals. In Proceedings of the ACM SIGSAC, Vienna, Austria, 24–28 October 2016; pp. 1068–1079. [CrossRef]
5. Zhang, J.; Li, M.; Tang, Z.; Gong, X.; Wang, W.; Fang, D.; Wang, Z. Defeat Your Enemy Hiding behind Public WiFi: WiGuard Can Protect Your Sensitive Information from CSI-Based Attack. *Appl. Sci.* **2018**, *8*, 515. [CrossRef]
6. Zhu, Y.; Xiao, Z.; Chen, Y.; Li, Z.; Liu, M.; Zhao, B.Y.; Zheng, H. Et Tu Alexa? When Commodity WiFi Devices Turn into Adversarial Motion Sensors. In Proceedings of the Network and Distributed Systems Security (NDSS) Symposium 2020, San Diego, CA, USA, 23–26 February 2020. [CrossRef]

7. Liu, J.; He, Y.; Xiao, C.; Han, J.; Ren, K. Time to Think the Security of WiFi-Based Behavior Recognition Systems. *IEEE Trans. Dependable Secur. Comput.* **2024**, *21*, 449–462. [CrossRef]
8. Huang, P.; Zhang, X.; Yu, S.; Guo, L. IS-WARS: Intelligent and Stealthy Adversarial Attack to Wi-Fi-Based Human Activity Recognition Systems. *IEEE Trans. Dependable Secur. Comput.* **2022**, *19*, 3899–3912. [CrossRef]
9. Bianchi, G. Performance analysis of the IEEE 802.11 distributed coordination function. *IEEE J. Sel. Areas Commun.* **2000**, *18*, 535–547. [CrossRef]
10. Wang, Z.; Huang, Z.; Zhang, C.; Dou, W.; Guo, Y.; Chen, D. CSI-based human sensing using model-based approaches: A survey. *J. Comput. Des. Eng.* **2021**, *8*, 510–523. [CrossRef]
11. Sharma, A.; Mishra, D.; Zia, T.; Seneviratne, A. A Novel Approach to Channel Profiling Using the Frequency Selectiveness of WiFi CSI Samples. In Proceedings of the 2020 IEEE Global Communications Conference (IEEE GLOBECOM), Taipei, Taiwan, 7–11 December 2020; pp. 1–6. [CrossRef]
12. Gringoli, F.; Schulz, M.; Link, J.; Hollick, M. 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, Los Cabos, Mexico, 25 October 2019; pp. 21–28. [CrossRef]
13. Ma, Y.; Zhou, G.; Wang, S. WiFi Sensing with Channel State Information: A Survey. *ACM Comput. Surv.* **2019**, *52*, 1–36. [CrossRef]
14. Hernandez, S.M.; Bulut, E. WiFi Sensing on the Edge: Signal Processing Techniques and Challenges for Real-World Systems. *IEEE Commun. Surv. Tutor.* **2023**, *25*, 46–76. [CrossRef]
15. Wang, F.; Han, J.; Lin, F.; Ren, K. WiPIN: Operation-Free Passive Person Identification Using Wi-Fi Signals. In Proceedings of the 2019 IEEE Global Communications Conference (GLOBECOM), Big Island, HI, USA, 9–13 December 2019; pp. 1–6. [CrossRef]
16. Ge, Y.; Taha, A.; Shah, S.A.; Dashtipour, K.; Zhu, S.; Cooper, J.; Abbasi, Q.H.; Imran, M.A. Contactless WiFi Sensing and Monitoring for Future Healthcare - Emerging Trends, Challenges, and Opportunities. *IEEE Rev. Biomed. Eng.* **2023**, *16*, 171–191. [CrossRef] [PubMed]
17. Zhang, Y.; Zheng, Y.; Qian, K.; Zhang, G.; Liu, Y.; Wu, C.; Yang, Z. Widar3.0: Zero-Effort Cross-Domain Gesture Recognition With Wi-Fi. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 8671–8688. [CrossRef] [PubMed]
18. Wang, W.; Liu, A.X.; Shahzad, M.; Ling, K.; Lu, S. Device-Free Human Activity Recognition Using Commercial WiFi Devices. *IEEE J. Sel. Areas Commun.* **2017**, *35*, 1118–1131. [CrossRef]
19. Sharma, A.; Jiang, W.; Mishra, D.; Jha, S.; Seneviratne, A. Optimised CNN for Human Counting Using Spectrograms of Probabilistic WiFi CSI. In Proceedings of the GLOBECOM 2022—2022 IEEE Global Communications Conference, Rio de Janeiro, Brazil, 4–8 December 2022; pp. 1–6. [CrossRef]
20. Wang, Z.; Jiang, K.; Hou, Y.; Dou, W.; Zhang, C.; Huang, Z.; Guo, Y. A Survey on Human Behavior Recognition Using Channel State Information. *IEEE Access* **2019**, *7*, 155986–156024. [CrossRef]
21. Li, J.; Mishra, D.; Seneviratne, A. CSI-Based NTC Using Ambient WiFi: Opportunities and Challenges. In Proceedings of the 2020 IEEE Globecom Workshops (GC Wkshps), Taipei, Taiwan, 7–11 December 2020; pp. 1–6. [CrossRef]
22. Xiong, J.; Jamieson, K. SecureArray: Improving Wifi Security with Fine-Grained Physical-Layer Information. In Proceedings of the 19th Annual International Conference on Mobile Computing & Networking, Miami, FL, USA, 30 September–4 October 2013; ACM: New York, NY, USA, 2013; pp. 441–452. [CrossRef]
23. Tay, Y.C.; Chua, K.C. A Capacity Analysis for the IEEE 802.11 MAC Protocol. *Wirel. Netw.* **2001**, *7*, 159–171. [CrossRef]
24. Ziouva, E.; Antonakopoulos, T. CSMA/CA performance under high traffic conditions: Throughput and delay analysis. *Comput. Commun.* **2002**, *25*, 313–321. [CrossRef]
25. Chen, Y.; Agrawal, D.P. Effect of Contention Window on the performance of IEEE 802.11 WLANs. In Proceedings of the 3rd Annual Mediterranean Ad Hoc Networking Workshop, Bodrum, Turkey, 27–30 June 2004; Citeseer: University Park, PA, USA, 2004; pp. 27–30.
26. Manshaei, M.H.; Hubaux, J.P. Performance Analysis of the IEEE 802.11 Distributed Coordination Function: Bianchi Model. Mobile Networks. 2007. Available online: http://www.manshaei.org/files/C1-80211-Perf-Bianchi.pdf (accessed on 15 July 2023).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.