WaveSpy: Remote and Through-wall Screen Attack via mmWave Sensing

Zhengxiong Li, Fenglong Ma, Aditya Singh Rathore, Zhuolin Yang, Baicheng Chen, Lu Su, Wenyao Xu*
University at Buffalo, SUNY, Buffalo, New York 14228, USA
{zhengxio, fenglong, asrathor, zhuoliny, baicheng, lusu, wenyaoxu}@buffalo.edu

***Abstract*—Digital screens, such as liquid crystal displays (LCDs), are vulnerable to attacks (e.g., "shoulder surfing") that can bypass security protection services (e.g., firewall) to steal confidential information from intended victims. The conventional practice to mitigate these threats is isolation. An isolated zone, without accessibility, proximity, and line-of-sight, seems to bring personal devices to a truly secure place.**

In this paper, we revisit this historical topic and re-examine the security risk of screen attacks in an isolation scenario mentioned above. Specifically, we identify and validate a new and practical side-channel attack for screen content via liquid crystal nematic state estimation using a low-cost radio-frequency sensor. By leveraging the relationship between the screen content and the states of liquid crystal arrays in displays, we develop WaveSpy, an end-to-end portable through-wall screen attack system. WaveSpy comprises a low-cost, energy-efficient and light-weight millimeter-wave (mmWave) probe which can remotely collect the liquid crystal state response to a set of mmWave stimuli and facilitate screen content inference, even when the victim's screen is placed in an isolated zone. We intensively evaluate the performance and practicality of WaveSpy in screen attacks, including over 100 different types of content on 30 digital screens of modern electronic devices. WaveSpy achieves an accuracy of 99% in screen content type recognition and a success rate of 87.77% in Top-3 sensitive information retrieval under real-world scenarios, respectively. Furthermore, we discuss several potential defense mechanisms to mitigate screen eavesdropping similar to WaveSpy.

I. Introduction

The digital screen is a pivotal output device which delivers intended information to users in modern devices (e.g., smartphones, laptops and access control). Due to the development of computer and networking cybersecurity services in core electronic devices, vulnerable computer accessories in physical worlds become a more effective and critical attack surface in practice, where digital screens are the most sought venue that adversaries can favorably leverage to steal information [1], [2], [3], [4]. Screen attacks can directly gain access to their organizational or personal resources and then pilfer the secrets (e.g., SSN, tax return, financial transactions, confidential data, and private communication), money (e.g., depository safe) and intellectual property (e.g., scientific research reports and blueprint). These leakages could lead to a series of catastrophic results, mainly severely increasing the risk of huge financial and reputation loss for both enterprises and individual [5], [6].

Mitigating risks of screen attacks has a long and rich history in the literature and is a core topic in the computer security community. Shoulder surfing, i.e., looking over the victim's shoulder, is one of the most investigated threats to user's screens [7]. With an increase in the user vigilance, however, adversaries have begun to exploit remote surveillance cameras to either directly or indirectly [8], [1] infer the screen content without line-of-sight assumptions. For example, it has been shown that various emanations from electronic displays, including ultrasound [9], electromagnetism (EM) [2], acoustic [3] and visible lights [10], can be leveraged to compromise the screen security. Therefore, one intuitive suggestion to enhance screen security is that people can place the screen in an enclosed location, e.g., no adversary-proximity/accessibility, no line-of-sight and occluded to the outside. However, is this ideal scenario truly secure against attacks? Our answer is *no*.

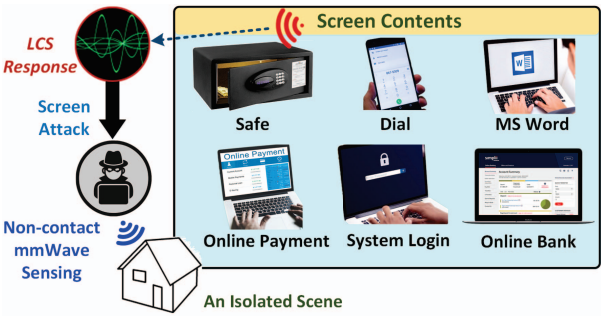


Fig. 1. Examples of different screen contents in the screen attack applications. The WaveSpy system can infer the screen content and underlying sensitive information even in an isolated scene in the real world.

In this work, we discuss a new screen attack approach by exploiting the display mechanism using **liquid crystal** (LC) elements. Screen contents on displays (e.g., LCD) are generated by the states (e.g., shape distributions) of LC arrays behind the display panel [11]. In other words, there is a deterministic dependence between liquid crystal states (LCS) and screen contents. By utilizing this dependent relation, we discover a new and stealthy LC-based side-channel to remotely attack screens in real world. Specifically, we hypothesize that if an adversary can monitor either the state of each liquid crystal or their distribution in a display, it is possible to retrieve screen information by exploiting the LC dependent model. Because this new side-channel attack approach did not assume any traditional passive emanation (e.g., EM or light [2], [10]) to the outside world, conventional wisdom on screen risk mitigation will fail, even in an isolated scene. If this hypothesis holds, there might be a novel screen attack approach which can

Address comments to wenyaoxu@buffalo.edu

© 2020, Zhengxiong Li. Under license to IEEE.
DOI 10.1109/SP40000.2020.00004

change the conventional wisdom on screen risk mitigation and compromise screen security under an isolated scene mentioned above as shown in Figure 1.

There are multiple technical challenges to realize the new attack system. First, how can we obtain the information of a liquid crystal state on the targeted display? There are several recent studies on using radio frequency (RF) signals to characterize objects (i.e., shape, geometric features and material types) [12], [13], [14], however, sensing resolution of a dot pitch (0.2-0.3mm) in an LCD display is still not reached. Second, to achieve a complete screen attack, the RF sampling frequency of liquid crystal states needs to be fast enough given that the modern screen flashes content every 4 to 10 milliseconds [15]. Lastly, it is critical to ensure the stealthiness of such an attack, without creating noticeable disclosure when eavesdropping the screen content using LCS remote sensing.

Our Work: In this paper, we present WaveSpy, a new real-world screen attack system which rests on the concept of a liquid crystal nematic pattern inside the display panel which acts as a passive signal modulator and reflects RF signals, namely *LCS response*, containing the screen information. We first investigate the dependent relation between the reflected RF signal and the content displayed on the digital screens using a portable mmWave probe. Afterward, we develop a RF signal processing scheme, including a deep learning model, to investigate the internal traits in the *LCS response* signal through wavelet analysis, followed by the spectrogram feature augmentation while ensuring minimum time complexity. Subsequently, we conduct an extensive attack evaluation to assess the performance of our model in real-world applications. Eventually, we conclude the study by developing WaveSpy, a remote ($5m$ away, through-wall), low-cost and stealthy screen inference system that precisely acquires the mmWave-based LCS response to facilitate two goals: (1) attack screens in a stealthy and through-wall manner; (2) retrieve the real-time sensitive information without the prior knowledge of their screen.

Summary: Our contribution in this work is as follows:

• *A new side-channel:* We discover a new side-channel to access the screen information from digital screens by exploiting the liquid crystal nematic response effect under the remote mmWave sensing.

• *A new attack system:* We design and implement an end-to-end hierarchal system, WaveSpy, to remotely monitor screen activities and retrieve screen using a mmWave probe and a novel signal processing scheme. WaveSpy can launch a remote screen attack without using traditional emanation, such as EM and light.

• *A new threat analysis:* We intensively evaluate the performance of WaveSpy with 100 different types of content on 30 different digital screens of modern electronic devices. WaveSpy achieves an accuracy of 99% in screen content type recognition and a success rate of 87.77% in Top-3 sensitive information retrieval under real-world scenarios.

• *A new defense exploration:* We discuss the effectiveness and study a set of passive and active countermeasures to prevent the leakage of information against this unprecedented information threat.

II. Attack Overview

A. Attack Scenario

We consider a scenario where a victim, namely Bob, utilizes common electronic devices (e.g., computer, mobile, smartwatch) in daily life. To ensure protection against attackers, Bob enables a password-based mechanism for every online activity including emailing, texting and monetary transactions and even facilitates an initial login screen for his devices. Observing Bob's vigilance, an innovative attacker, hereafter Alice, aims to breach the established security and extract sensitive information without the victim's knowledge.

Scenario #1 (Privacy Invasion): To infer the type of screen content and user activity at a certain time, Alice intends to acquire information about the specific application initiated by Bob, usage statistics and the underlying content in real-time.

Scenario #2 (Security Attack): To compromise the personal security, Alice senses the information presented on the digital screen from a long distance or through-wall and reconstructs the sensitive information (e.g., PIN, password, lock pattern, words or sentences) without alerting Bob or his nearby surroundings, or even Bob in a closed room.

In contrast to prior work, we envision that the following constraints restrict Alice:

• **No Device Proximity:** Bob is alert to traditional shoulder-surfing or channel state information (CSI) attacks in terms that either Alice cannot get close or there is a blockage (e.g., wall) between Alice and Bob.

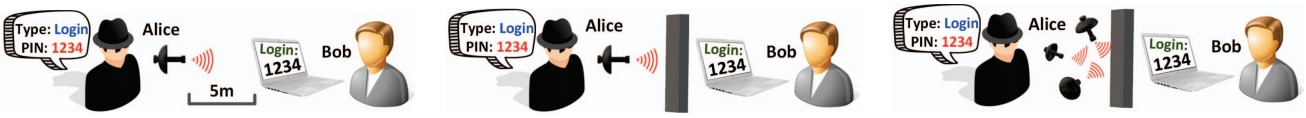
• **No Pre-installed Malware:** Assuming that Bob's electronic device is isolated from the Internet or any other communication channel, Alice is unable to directly compromise the electronic device from malware such as Trojans or malicious web scripts.

• **No Line-of-sight:** Alice cannot directly visualize the screen content or Bob's physiological attributes (e.g., hand motions, eye movement) during the activity phase from any direction. Considering the alertness of Bob and real-world environments, there are no surveillance cameras that can remotely monitor digital screen contents.

Traditional EM-based, acoustic-based, CSI-based and vision-based screen attacks (e.g., [2], [3], [9], [1]) cannot work under an application scenario with the constraints mentioned above. However, screen security in this scene will not be necessarily guaranteed when we consider that Alice can leverage a tiny and cost-effective mmWave probe to perform real-time surveillance of screen information from an adjacent room and steal the information from the target victim, as shown in Figure 2.

B. Attack Application Study

In addition to the content type recognition, login authentication is one of the most fundamental types of security protection enabled by users in their personal devices. Moreover, this mechanism is increasingly deployed in other cyber-physical



(a) Remote (b) Through-wall (c) Various angles

Fig. 2. Three typical attack scenarios in daily life: (a) Alice infers the screen from a remote location; (b) Alice leverages the penetration properties of mmWave for through-wall inference; (c) Alice has the freedom to choose various sensing distance and angle to maximize the inference accuracy.

technologies such as Internet-of-Things (IoT), electronic depository safe, and smart homes. Figure 3 shows *three* attack applications in this study, including login using the virtual button, physical button and picture password. Based on the user acceptability and device operation, there are three primary categories of login methods:

1) Login Using Virtual Buttons: Presently, the most popular form of human-computer interaction is through the touch-screen via the virtual buttons. For different passwords, including *PIN*, *character password* and *pattern lock*, the user pauses for a brief moment in between subsequent inputs to recognize the user interface (UI) correspondence in the form of color change in the pressed buttons. The typical radius of each button on the screen can be small to $6mm$ (e.g., iPhone 7 Plus).

2) Login Using Physical Button: The non-touch based electronic devices (e.g., desktop monitor, laptop, cell phones and smart locks) require a user to input the password by pressing a physical button on the keyboard. This type of login has two categories. First, the password can appear as an asterisk character on the screen, similar to personal computer login. The second one can be found on some security devices where the password is visible on the screen as typed. The radius of each asterisk on the screen can be as small as $1mm$ (e.g., MacBook Pro). The typical size of each character on the screen can be $10mm$ by $6mm$ (e.g., security intercom system).

3) Login Using Picture Password: In contrast to the previous login methods, picture password offers the merit of unpredictability and superior usability. Rather than pressing the button on a virtual or physical keyboard, it allows a user to create three different gestures in a sequence on the specific position of the selected image and use those gestures as the password. The gesture can be any combination of circles, straight lines, and taps with predefined tolerances during the login process. The typical radius of each tap UI corresponding on the screen can be small to $6mm$ (e.g., Dell U2415).

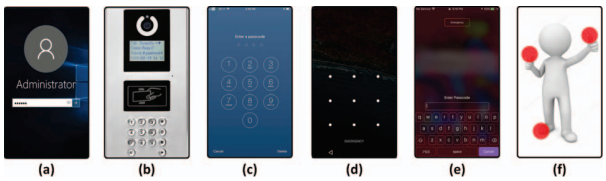


Fig. 3. Six representative attack applications: (a) password length; (b) numeric password; (c) PIN; (d) pattern lock; (e) password; (f) picture password. The attack on each application is extensively evaluated in Section VI-B.

In the remainder of this paper, we present how WaveSpy performs the screen attack on aforementioned login-based authentications. The WaveSpy system also demonstrates the significant promise in reconstructing critical information (e.g., words, sentences) in the digital screen as shown in Section VIII.

III. Liquid Crystal State in Displays: A Closer Look

A. Background and Hypothesis

Presently, the liquid-crystal display (LCD) and organic light emitting diode (OLED) are the mainstream screen technologies adopted in the majority of electronic devices [16]. Our attack approach is applicable to both types of displays because they have the same LCS-based working principles. In the following part, we will review the display architecture and have a closer look at the LCS effects in modern displays.

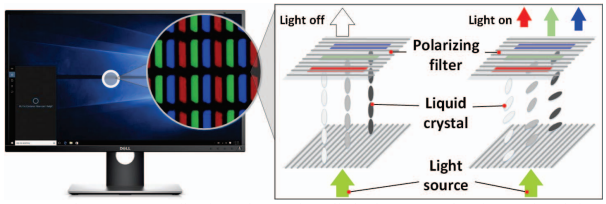


Fig. 4. The content displayed on the digital screen is determined by the arrangement of liquid crystal nematic patterns.

Working Principles of Displays: The LCD panel comprises a thin layer of glass substrate embedded with liquid crystals, while a white fluorescent backlight is positioned behind the screen to produce the images in color or monochrome. Each liquid crystal is aligned between two polarizing filters (parallel and perpendicular) as illustrated in Figure 4. Without the mentioned placement, light passing through the first filter would be blocked by the second (crossed) polarizer. The liquid crystal nematic pattern responds and changes its arrangement based on the voltage applied across the liquid crystal layer in each pixel, thereby altering the polarization of light. Besides, the variations in the liquid crystal nematic patterns lead to varying amounts of light to pass through, constituting different contents on the screen [11]. Note that the liquid crystal nematic pattern remains significantly stable under the probing of RF signals [17].

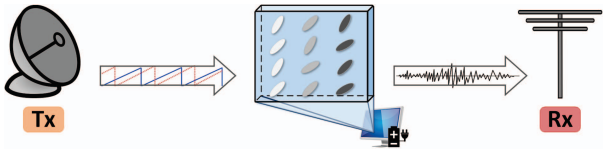


Fig. 5. The liquid crystal nematic patterns on the digital screen incites a *LCS response* under the radio-frequency (RF) beam. Different liquid crystal nematic patterns cause different *LCS responses*.

Liquid Crystal Nonlinear Effects: When a continuous wave with transmitting frequency f_0 from the mmWave probe is projected towards the target, the RF response is modulated with a set of sub-carrier frequencies due to the properties of the target (e.g., liquid crystal pattern, material reflection efficiency). Similarly, given that the screen enters the RF beam field as shown in Figure 5, the liquid crystal nematic patterns are perceived as an array of antennas in the resolution of the mmWave [18], [19]. These antennas act as a passive processor and manipulate the transmit mmWave signals to generate a distortion formulated as:

$$\begin{cases} Z(t) = \phi(\varphi(t), \triangle n, \kappa, \gamma, V_c) \otimes h_f(t), \\ \triangle n = \sqrt{\varepsilon_{\parallel}} - \sqrt{\varepsilon_{\perp}}, \end{cases} \tag{1}$$

where $\varphi(t)$ represents a collection of mmWave subcarriers for the response signals, $\phi(\varphi(t), \cdot)$ is the modulation function of the liquid crystal (LC) patterns, $\triangle n$ is the LC rotational viscosity, κ is LC the elastic constant, γ is the LC rotational viscosity, V_c is the LC threshold voltage, $\otimes$ stands for convolution computing, $h_f(t)$ is the ideal bandpass filter function for the carrier bandwidth, $\varepsilon_{\parallel}$ is the dielectric constant when the electrical field is parallel to the director of the liquid crystal molecules and $\varepsilon_{\perp}$ is the dielectric constant when the electrical field is perpendicular to the director [20], [21]. After the modulated signal radiates from the screen, it is captured by the probe receive (Rx) antenna. Therefore, *LCS response* of the digital screen incorporates profound information of the liquid crystal nematic patterns and holds the potential for monitoring the displayed content type or sensitive information.

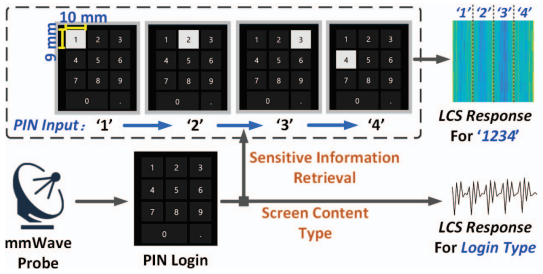


Fig. 6. The *LCS response* illustration for PIN login mechanism with input '1234'. Every numeric input has a distinct *LCS response*, thereby enabling sensitive information retrieval.

Sensing Frequency Estimation: In order to obtain the RF response with exceptional quality and promote the attack performance, it is critical to utilize a proper sensing frequency. Under the most common circumstances, the length of a typical icon along a screen is larger than $l = 3mm$, and the effective dielectric constant of the LC array is close to $\varepsilon = 3.66$ [22]. According to [23], the sensing frequency f_0 can be reckoned as

$$f_0 = \frac{c}{2l\sqrt{\varepsilon}} = \frac{3 \cdot 10^8 m/s}{2 \cdot 0.003m \cdot \sqrt{3.66}} \approx 24GHz, \tag{2}$$

where c is the propagation speed of a radar wave in air [20]. Therefore, we deploy the 24GHz sensing frequency, which can be approximately recognized as mmWave, in our study.

Hypothesis: Owing to the facts that the liquid crystal nematic pattern has a deterministic mapping to each displayed content and mmWave can remotely sense LC patterns, *there exists a unique and measurable connection between the displayed content and associated LCS response obtained under the mmWave beam reflectance of the liquid crystal layers*. Therefore, as shown in Figure 6, it is feasible to develop a portable mmWave probe with advanced signal processing techniques to capture this connection. The attacker can leverage the information for screen content type recognition and sensitive information retrieval, whose problem formulations are further discussed in Sections IV-D & IV-E.

B. A Preliminary Study of LCS Response: A Side-channel on LCD Display

Proof-of-concept: To validate the above *hypothesis*, we conduct a preliminary experiment using *three* different mainstream off-the-shelf displays from representative device categories (i.e., Dell U2415 monitor, Samsung Galaxy S9, and Apple Watch Series 3) with different user activities. The spectrograms of *LCS responses* with associated content are shown in Figure 7. These devices are stimulated with the mmWave probe with a distance from the devices. The reflected signal profile is explored in the spectral domain. The x-axis represents the modulated frequency; y-axis describes the amplitude in the received signal. Given the vast contrast between the amplitude and the frequency of received *LCS response* marked in the blue circle, the liquid crystal nematic pattern variations have sufficient space to enable content recognition. Furthermore, we observe that the response distributions among different devices are entirely distinct owing to the different device hardware structures and the screen designs.

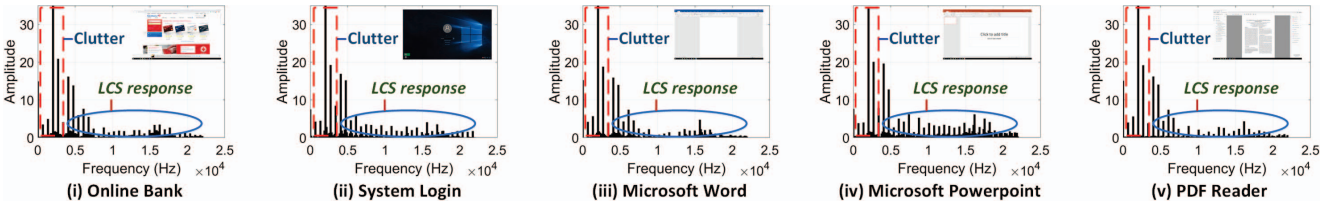
A Study on Wall Effects: In a real-world scenario, it is not uncommon for Bob to access his device in another room with the wall acting as an obstacle between the digital screen and attacker Alice. Therefore, it is crucial to investigate whether the material of the wall will block or interfere with the *LCS response* [24]. We conduct the experiment by positioning a $15cm$ thick wall between the mmWave probe and the digital screen of MacBook Pro. The sensing distance is $80cm$. Figure 8 demonstrates that in the overall signal spectrum, there is minute variation in the amplitude of low-frequency components from the wall and nearby objects. Upon closely analyzing the area within the *LCS response* (marked in the blue circle), there are observable variations in the high-frequency components among the different content displayed on the screen. However, this model is insufficient to precisely identify the liquid crystal nematic pattern as the differences between the *LCS responses* are not significant. Thus, we further develop the WaveSpy system for screen monitoring.

IV. System Framework

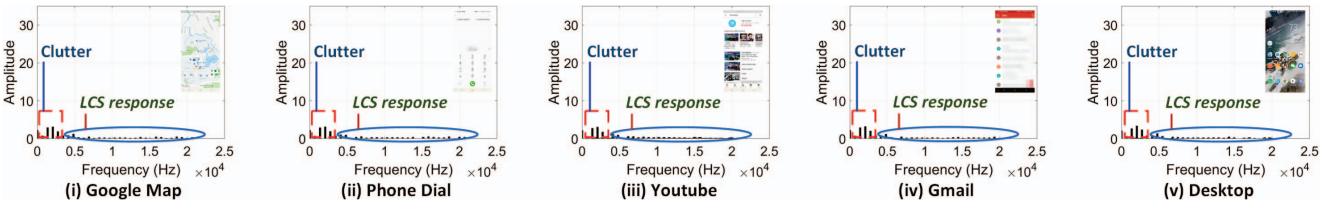
A. WaveSpy: A Through-wall Screen Attack System

We propose a portable, unobtrusive and robust system to facilitate screen activity type recognition and sensitive information reconstruction as shown in Figure 9.

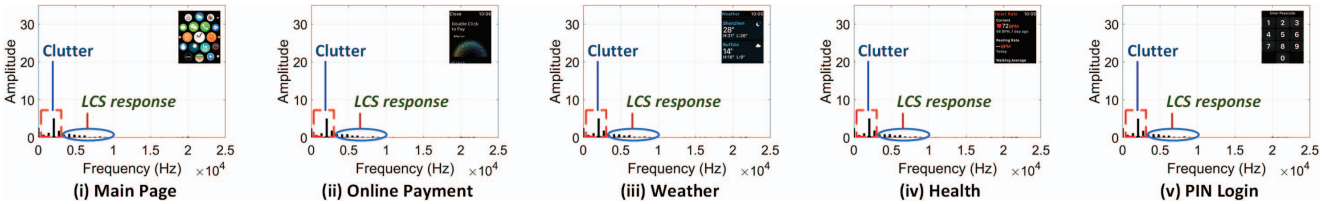
***LCS Response* Stimulation and Modeling:** We introduce the RF hardware in WaveSpy to stimulate and acquire the *LCS*



(a) *LCS responses* on Dell U2415 display with different content types.

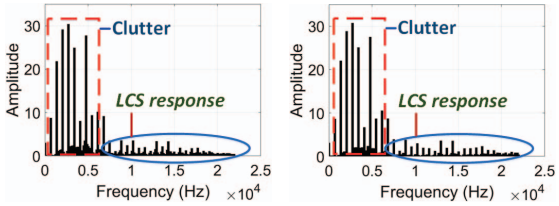


(b) *LCS responses* on Samsung Galaxy S9 with different content types.

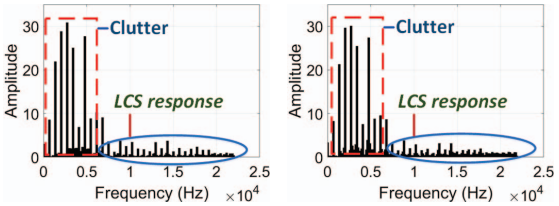


(c) The *LCS responses* on Apple Watch Series 3 with different content types.

Fig. 7. Different screen content present different *LCS responses* (the spectrum in the red circles are distinct in frequency and amplitude) when forced by the same mmWave probe. The screen content on each screen is displayed on the left.



(a) Content Type: Microsoft Word (b) Content Type: Facebook



(c) Content Type: System Login (d) Content Type: Online Bank

Fig. 8. The non-linear response of the wall or surrounding objects is distinct in frequency and amplitude from the *LCS response* of digital screen in MacBook Pro, indicating the feasibility of indirect screen monitoring.

response from electronics. Pulse-Doppler radar that emits a set of periodic powerful pulse signals has been largely used in airborne applications [25], such as the target range and shape detection. However, when a short-time pulse stimulus, which has an infinite frequency band, is applied to illuminate the electronics, the corresponding spectrum response will be overlapped with the stimulus signal and difficult to recognize. Therefore, WaveSpy selects a frequency-modulated continuous-wave (FMCW) radar with a narrow passband filter [26]. The FMCW radar continuously emits periodic narrow-band chirp signals whose frequency varies over time. The non-linear interrelation to these narrow-band stimuli will generate distinct frequency response, and the received signals will carry distinguishable *LCS responses* when the stimuli signals hit the target display. After the manipulated signal is radiated from the display, *LCS responses* will be captured by the RF probe receiver antenna (Rx).

Screen Monitoring: Once the data format obtained from the mmWave probe is demodulated to filter the interference and noise while guaranteeing the preservation of information. A wavelet-based response analysis is employed to extract a set of comprehensive features and formulate a sequence of multi-class deep neural networks based classification algorithm to obtain the content type and the sensitive information displayed on the digital screen.

B. Screen Localization

Searching and localizing the display of interest is the first step in screen attack. In this section, we introduce the screen searching protocol to localize the screen position under the angular coordinates. First, WaveSpy steers the mmWave beams to sweep through all directions in the target areas. Second, considering the display will generate LCS which is significantly different from the background (e.g., LCS response), we utilize LCS-based features (see TABLE I with a threshold) to detect existence of display and estimate the orientation of the target screen. This process is efficient and can be finished within several milliseconds. Adaptive beam

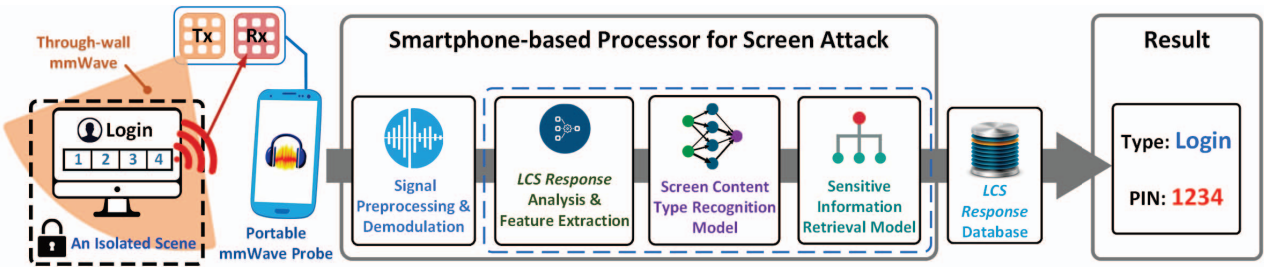


Fig. 9. The system overview for WaveSpy to non-invasively recognize the screen content type and retrieve the security information on the screen. It comprises of a mmWave sensing module in the front-end and a screen monitoring module in the back-end.

TABLE I
LIST OF FEATURES EXTRACTED FROM THE *LCS* RESPONSE.

Category	Feature Names
Temporal Features	Mean Value, Standard Deviation, Skewness [28], Kurtosis [29], Lowest and Highest Value
Spectral Features	Mean Value, Standard Deviation, Kurtosis, Crest Factor [30], Flatness [31]
Others	*MFCC (12)* [32]

training protocols can be adopted to improve the accuracy in screen localization further [27]. Therefore, WaveSpy can pinpoint the screen and prepare for the *LCS response* analysis. Note that we evaluate the WaveSpy performance sensitivity to the probing orientation in Section VII (see Figure 14 in details).

C. The Wavelet Analysis on LCS Response

After removing the direct current (DC) component, modulated LCS signal $s(t)$ becomes a signal with zero-mean and some variance and satisfies the following condition: $\int_{-\infty}^{\infty} s(t)dt = 0$, which indicates $s(t)$ is a waveform. ***P()*** uses $\psi_{a,b}$ and $\phi_{a,b}$, where $\phi_{a,b} = \frac{1}{\sqrt{a}}\phi(\frac{t-b}{a})$ and $\psi_{a,b} = \frac{1}{\sqrt{a}}\psi(\frac{t-b}{a})$, as the mother wavelet function that satisfies the condition of dynamic scaling and shifting, where a and b are the scale and translation parameters accordingly [33]. In order to get signal properties at high frequency, the wavelet-based analysis is achieved as Eq. (3):

$$s(t) = P_0 + P_1 + P_2 + P_3, \tag{3}$$

where $s(t)$ is the LCS response, $P_0 = \frac{1}{C_\phi}\int_{-\infty}^{\infty} F_W(a_0,b)\phi_{a_0,b}\frac{db}{\sqrt{a_0}}$ is the approximation part, $P_1 = \frac{1}{C_\psi}\int_{-\infty}^{\infty} F_W(a_1,b)\psi_{a_1,b}\frac{da}{a_1^2}\frac{db}{\sqrt{a_1}}$ is the Level 1 detail part, $P_2 = \frac{1}{C_\psi}\int_{-\infty}^{\infty} F_W(a_2,b)\psi_{a_2,b}\frac{da}{a_2^2}\frac{db}{\sqrt{a_2}}$ is the Level 1 detail part, $P_3 = \frac{1}{C_\psi}\int_{-\infty}^{\infty} F_W(a_3,b)\psi_{a_3,b}\frac{da}{a_3^2}\frac{db}{\sqrt{a_3}}$ is the Level 3 detail part, $F_W(a_0,b)$, $F_W(a_1,b)$, $F_W(a_2,b)$ and $F_W(a_3,b)$ are the coefficients.

For the inverse transform to exist, we require that the analyzing wavelet satisfies the admissibility condition, given in the following: $C_\phi = 2\pi\int_{-\infty}^{\infty}\frac{|\hat{\phi}(\omega)|^2}{\omega}d\omega < \infty$ and $C_\psi = 2\pi\int_{-\infty}^{\infty}\frac{|\hat{\psi}(\omega)|^2}{\omega}d\omega < \infty$, where $\hat{\phi}(\omega)$ and $\hat{\psi}(\omega)$ are the Fourier transform of $\phi(t)$ and $\psi(t)$ respectively. Also, C_ϕ and C_ψ are constants for corresponding wavelets. Afterwards, the detail parts of the *LCS response* can help us to further achieve screen content type recognition and sensitive information retrieval.

D. Screen Content Type Recognition

Content type recognition can be formulated as a multi-class classification problem. We begin by defining the key terms and then formulate the content type recognition problem.

`Definition 1` *(The LCS Response Set on the Liquid Crystal Nematic Pattern by mmWave)*: For a mmWave sensing process, let ***s*** denote a mmWave response of the liquid crystal nematic pattern that is attained by a certain sample method. ***S*** is defined as the response set, which contains every liquid crystal nematic pattern response. Specifically, we define s_0 as a complete sensing signal that has the entire information about characteristics of the source content on the screen. Therefore,

$$\forall s \in S, \emptyset \subset s \subseteq s_0. \tag{4}$$

Given the *LCS response* signals, it is hard to classify them using similarity and distance-based approaches directly. The reason is that *LCS responses* have a large variation in magnitudes as well as frequencies, which leads to irregularity and asymmetry. Therefore, we present the wavelet-based analysis which is resilient to the scale and magnitude variation.

`Definition 2` *(Feature Extraction from Wavelet-based Analysis)*: The response analysis function can be any function that demodulates the response, reflects the liquid crystal nematic characteristics, obtains the integration of content features, and outputs a feature vector. We use ***P()*** to represent the *LCS response* analysis function.

In this paper, we use wavelet transform (WT) as ***P()***, which is an effective multi-resolution analysis tool for signal decomposition [34], [35]. The ***P()*** approach can overcome the shortcoming of Fourier analysis, which only works in the frequency domain, not in the time domain [36]. $s(t)$ matches the waveform and can be decomposed into many groups of coefficients in different scales with ***P()*** through differently scaled versions, as shown in Section IV-C.

Subsequently, we obtain the approximation and Level 1, 2 and 3 detail parts in Eq. (3) (in Section IV-C). As the above mentioned in Section III, the unique characteristic information is hidden in the high-frequency range (i.e., the detail parts). Intuitively, the signal with more features in the high-frequency signal will contain more distinguishable characteristics of the screen content and thereby achieve a better recognition

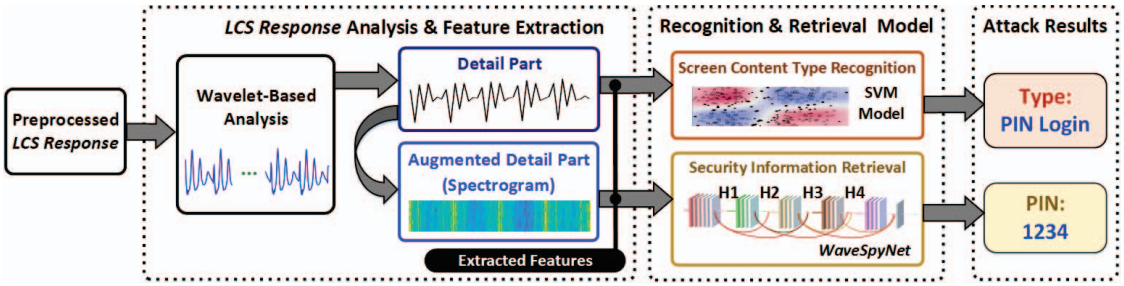


Fig. 10. The flow chart of the screen monitor module, including two parts: (a) *LCS response* analysis & feature extraction, and (b) screen content type recognition & sensitive information retrieval model.

accuracy. However, it also increases the computation overhead. To balance this trade-off, we empirically choose the level 3 detail part (we will investigate the system performance with different $level$ setups in Section VI-A). As a result, we exploit the internal traits in the *LCS response* signal by extracting a 40-dimension feature vector in spectrum domains.

`Definition 3` *(Screen Content Type Classification)*: ***C()*** is the classification function that utilizes several response features to predict the screen content type. The specific implementation of ***C()*** responds to the real-world scenarios and the applied database mentioned in Section IV-D.

Formulation 1 *(User Activity Monitoring)*: The purpose of screen content type recognition is to identify the specific application and user online activity initiated by the mmWave response $\boldsymbol{s}$. We first extract its feature vector using ***P()***, and then recognize the application type with the screen content type classification function ***C()***. β is used to denote the result of the predicted specific application on the screen as follows:

$$\beta = C(P(s)). \tag{5}$$

In the WaveSpy system, we employ universal and easy-to-deploy classifiers, i.e., Support Vector Machine (SVM) and K-Nearest Neighbor (KNN) as the screen content type/user activity classification method $C()$, to identify content type based on the extracted features. Previously, SVM and KNN have been successfully applied in wireless sensing recognition [13] and physical cybersecurity [37], respectively. SVM locates an optimal hyperplane in high-dimensional space to perform the classification. The Gaussian radial basis function is selected as the kernel function to map the original data to a higher dimensional space. However, KNN stores all available cases and classifies new cases based on a similarity measure. We opt to use SVM as the classifier after we compare their performances in Section VI-A.

WaveSpy uses a supervised approach to classify content types, beginning with a training phase followed by testing. During the training of the **Classifier**, n traces of *LCS response* signals from each content type are collected. For m content types in the database (namely, m pre-registered classes), $n \times m$ feature vectors are used to train the classifier altogether. During the testing phase, WaveSpy collects a trace, extracts a feature vector, and inputs to the classifier model. The classifier model generates the probability set of classifying this test trace into each pre-trained class. We output three candidates with the top three possibilities.

E. Sensitive Information Retrieval

1) Sensitive Information Retrieval Method: When the user presses a button on the screen, the pixel-level configuration of this button changes, showing the correspondence UI illumination and allowing the user to confirm the correctness of the input, which causes different *LCS responses* as shown in Figure 6. From a high-level point of view, WaveSpy infers password or sensitive information of the user by collecting and analyzing the *LCS response* sequence received on the mmWave probe. The sequence length is equal to the user's typing duration. When the screen content type is detected as the login interface, this sensitive information retrieval model is then activated to detect PIN passwords.

A traditional approach to address this problem is first to segment the input signals into N pieces, where N is the length of the PIN passwords, and then to classify each segmented piece as a digit. However, it is difficult for us to segment those signals manually. The other possible solution is to extract features for the whole signals first, and then to train a classifier for each PIN digit. However, we observe that the differences among those signals on different screen contents are significantly miniature. In other words, the extracted features of different signals are nearly the same, which leads to the failure of both SVM and KNN.

To tackle this challenge, we employ deep neural networks (DNN) [38] in the WaveSpy System. The advantage of adopting DNN is that it is able to learn better feature representations automatically and further makes the signals distinguishable. Moreover, the DNN-based security inference framework can be easily applied to new scenarios without domain knowledge about the functioned sensors. Thus, we propose a novel end-to-end deep learning based approach, which takes the raw sensing data as the input and computes the most likely sensitive information that the users have entered.

First, sensitive information retrieval can be formulated as a sequence multi-class classification problem. However, the original sequence signal is too large to be considered the input of DNN. For example, the recording for PIN typing usually produces a four-second long audio containing about 176,400 samples in total. Thus, we utilize the technique of Joint Time-

Frequency Analysis [39] to convert the sequence signal into a spectrogram.

Definition 4 *(Feature Augmentation using Spectrogram)*: Let ***W()*** be the function to generate the spectrogram from the input signal, which is defined in Eq. (6) as follows:

$$\begin{cases} X(m,\omega) = \sum_{n=-\infty}^{\infty} x[n]w[n-m]\exp(-j\omega n), \\ W\{x(t)\}(m,\omega) \equiv |X(m,\omega)|^2. \end{cases} \quad (6)$$

Note that in our implementation, instead of using the original signal, we use the level 3 detail part from wavelet decomposition as the input of ***W()*** as shown in Figure 10, which reflects the internal trait in the *LCS response* signal (evaluated in Section VI-A). Finally, the converted spectrograms are the inputs of DNN.

Definition 5 *(Sensitive Information Classification Function)*: ***V()*** is defined as the DNN model that utilizes several response-analysis to predict the sensitive information shown on the screen.

It is worth noting that for each real-world scenario mentioned in Section II-B, we train a customized ***V()***. The details of the DNN model are further illustrated in Section IV-E2.

Formulation 2 *(Sensitive Information Retrieval)*: The final goal of sensitive information retrieval is to reconstruct the sensitive information from the input signal ***s*** using the response analysis function. Since there are N characters in the credential, for each reconstructed character, we train a specified DNN model. Let T_n be the candidate results of the sensitive information on the screen for the ***n***th character:

$$T_n = \mathbf{V}_n(\mathbf{W}(\mathbf{s})). \quad (7)$$

Formulation 3 *(Ranking Sensitive Information Candidates)*: To deal with noisy mmWave signal traces and accommodate the input number allowed by the system, we need to rank the candidate credentials according to their possibilities. $R(n)$ is a function to obtain top ***k*** candidates for predicting the sensitive information shown on the screen:

$$\begin{cases} R(n) = f(T_n), & n = 1 \\ R(n) = f(R(n-1) \circ f(T_n)), & n \geq 2 \end{cases} \quad (8)$$

where the operation $\circ$ represents that all the data in one set are multiplied by all the elements in the other set, and $f(\circ)$ is the function to find the candidates with top ***k*** possibilities among the results. Thus, the algorithm for the screen attack in WaveSpy is established in Algorithm 1.

2) Sequence-to-Credential Model for General Security Information Inference: Though the original sequential signal can be transformed to the spectrogram using the time-frequency analysis technique, the transformed spectrograms are extremely similar as shown in Figure 7 and hard to be distinguished by traditional classification algorithms such as SVM and CNN. To make these spectrograms distinguishable, we design WaveSpyNet, a Densely Connected Convolutional Networks (DenseNet) [40]-based classifier for the sensitive information retrieval, i.e., ***V()***. Besides perfectly guaranteeing the classification performance, WaveSpyNet also alleviates the vanishing-gradient problem, strengthens feature propagation,

Algorithm 1: The Screen Attack by WaveSpy

Input: $s(m)$: m *LCS response* traces from the screen
Output: β: the Screen content type recognition result
R: the sensitive information retrieval result

```
Initialize C, P, R, V, W, β, T;
%Screen content type recognition:
for i ∈ {1, ..., m} do
    β(i) = C(P(s(i)));
    if β(i) ==' Login' then
        %Sensitive information retrieval:
        T(i) = V(W(s(i)));
        return R(T(i));
    return β(i);
```

encourages feature reuse, and substantially reduces the number of parameters, which naturally satisfies the requirements of our problem. Next, the details of WaveSpyNet are introduced.

The WaveSpyNet consists of an initial layer, four dense blocks, three transition layers, and a prediction layer as shown in Figure 10. The initial layer aims to convert the transformed spectrogram $W(s) \in \mathbb{R}^{128\times128}$ into a latent space. The initial layer includes four consecutive operations: a convolution (Conv), followed by a batch normalization (BN), a rectified linear unit (ReLU) and a max pooling. Let $\mathbf{x}_0^1 \in \mathbb{R}^{128\times128}$ represent the output of the initial layer, which is the input of the first dense block.

Each dense block $b \in \{1, \cdots, \mathcal{B}\}$ comprises $\mathcal{L}^b$ layers, and each layer implements a non-linear transformation $H_\ell(\cdot)$, where ℓ indexes the layer. $H_\ell(\cdot)$ is defined as a composite function with three consecutive operations: BN-RELU-Conv. The most greatest advantage of WaveSpyNet is that for the ℓ-th layer ($1 \leq \ell \leq \mathcal{L}^b$ and $\ell \in \mathbb{R}^+$), the input of $H_\ell(\cdot)$ is the direct concatenation of all the previous layers, i.e., $[\mathbf{x}_0^b, \mathbf{x}_1^b, \cdots, \mathbf{x}_{\ell-1}^b]$. The output of the ℓ-th layer is represented by:

$$\mathbf{x}_\ell^b = H_\ell([\mathbf{x}_0^b, \mathbf{x}_1^b, \cdots, \mathbf{x}_{\ell-1}^b]). \quad (9)$$

When the size of filters in convolutional layers changes, the concatenation operation used in Eq. (9) is not viable. Thus, a transition layer is designed to change the size of filters, which is between two consecutive dense blocks as shown in Figure 10. The transition layer consists of a batch normalization layer, a 1×1 convolutional layer followed by a 2×2 average pooling layer. The output of the transition layer is the first input of the next dense block.

The above two operations are repeatedly conducted until arriving at the last dense block. The output of the $\mathcal{B}$-th dense block is the input of the prediction layer. A simple linear function is used to produce a latent vector to represent the original input signal or the transformed spectrogram. Actually, each signal contains N characters. In the implementation, we train a separate WaveSpyNet with the cross-entropy loss, and we choose Adam, a light-weight stochastic function optimizer [41] to fine-tune the WaveSpyNet parameters.

V. Performance Prototype and Evaluation

A. WaveSpy System Implementation and Integration

WaveSpy utilizes an FMCW mmWave probe equipped with a pair of 4×4 antenna arrays. The transmission power is around one Millie Watt. The RF signal is processed using the novel mechanism of the inverse synthetic aperture radar [13]. Besides, the probe can be mounted on the wall or integrated with other portable devices like a laptop or smartphone. Therefore, WaveSpy can launch the attack with a convenient and user-friendly manner in real-world applications.

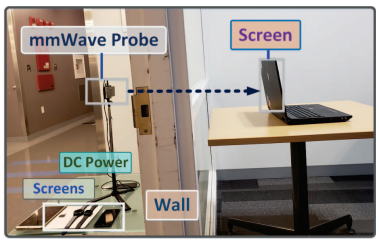


Fig. 11. The setup for the evaluation mainly consists of three parts: a mmWave probe, screen, and wall.

B. Experiment Setup

1) Experiment Preparation and Data Collection: In order to comprehensively evaluate every possible form of the *LCS response*, we employ 30 digital screens and categorize them in six groups; specifically, nine monitors, five laptops, three tablets, six smartphones, four wearable devices, and three other electronic devices. Among these, 21 devices have LCD displays and 9 have OLED displays. All displays are well functioning with no defects. Their sizes vary between 1.5 inches to 70 inches while the usage time ranges from 1 to 11 years. All screens are under the default profile that is set to at the factory. The walls are made of wood and concrete, two mostly used in modern buildings. We recruit ten anonymous participants. It is ensured that every participant follows the host institute internal review board protocol. During the experiment, the mmWave probe is placed $80cm$ from the screen and its initial position is recorded as $0°$ orientation in the level plane. During the experimental phase, we position the screen behind the wall or obstacle (see Figure 11).

In general, we conduct two sets of experiments to enable screen content type recognition and sensitive information retrieval. We repeat each experiment for ten times for every participant. For the *content type recognition*, we prepare and label 100 screen content types from common user activities, e.g., typing on Microsoft (MS) Word, and collect $2s$ of sensing data for the specific content type on each trial. As mentioned earlier, each participant is asked to repeat for 10 times. From the overall dataset, we randomly extract 100 traces for each content type (totaling 100 x 100 = 10,000 traces) with respect to an individual location. Unless specified, we randomly choose 7,000 out of 10,000 traces from each device as the training set and the remaining for testing.

For the *sensitive information retrieval*, the participants were asked to input a diverse set of sensitive information, including a PIN on the numeric keyboard. The official default interfaces are utilized here (e.g., system login), where only a certain region (system default size) is changed along with the input while other areas stay unaltered. For example, in *S2A: Password Length*, the font of a character is 10pt; in *S2E: Password*, the size of a virtual button is 70×50 pixels. For every piece of sensitive information listed above on each device, we collect more than 21,000 traces beforehand to train a DNN model. Notedly, the other 1,000 traces of data are utilized for the testing set.

2) Metrics: We employ **Top-k** ($k = 1$, 2 and 3) inference accuracy as the primary performance metric, which implies the candidates with top k possibilities. Specifically, the system generates a set of ranked candidates (i.e., PINs, lock patterns, or letters) for each trial. We claim that a trial succeeds if the true input appears in the Top-k candidates. Top-k inference accuracy is defined as the percentage of successful trials. Furthermore, to evaluate the picture password, we utilize *Distance Estimation Error (mm)* to measure the estimation error of the tapped position on the screen.

VI. Evaluation I: A Control Study

In this section, we perform a control study to validate the legitimacy of our proposed system design under the ideal environmental condition.

A. The Performance of Screen Content Type Recognition

The performance of WaveSpy depends on the design of recognition approaches. To investigate the sensitivity of classification model and verify the capability our selected features, we perform a multi-level detail part (mentioned in Section IV-D) analysis, denoted as L1, L2 and L3, towards two mostly used classification configurations, i.e., SVM and KNN. The data are acquired from the database, hereafter Data Collection, built using our sensing system.

With respect to Top-3 inference, SVM achieves an accuracy of 90.71%, 94.13%, and 99.13% for L1, L2 and L3 schemes respectively. Correspondingly, KNN achieves 78.19%, 87.89%, and 93.98% for the three schemes as shown in Figure 12. The satisfactory performance on both classifiers indicates the effectiveness of our feature vector (see **Definition 2**) in reflecting the unique and salient characteristics of *LCS responses*, while the performance of SVM is superior compared to KNN for this application. It is worth mentioning that during the acquisition of traces, the content on the screen is not static because of several factors in the screen corner (e.g., UI animation, advertisements or updated news), which also increases the difficulty of this task. Against the original belief that this may severely interfere with recognition performance, WaveSpy maintains high inference accuracy, implying that the general layout (or template) of the application is static and unique.

B. The Performance of Sensitive Information Retrieval

To maximize the efficiency of WaveSpy in retrieving the sensitive information, the attacker may know the security mechanism employed by the victim on his electronic device

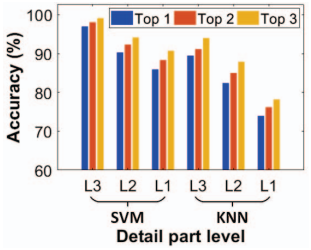


Fig. 12. The overall performance for screen content type recognition (Scenario 1) with three different detail parts and two common classifiers.

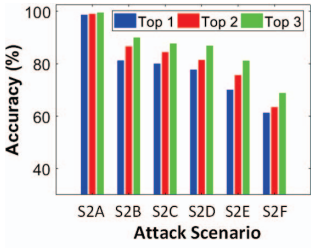


Fig. 13. The overall performance of the sensitive information retrieval in six types of login information (S2A∼F described in Section VI-B).

prior to performing an attack. However, due to the increasing growth of smart devices supporting multiple login mechanisms, it would be ideal for the attacker if WaveSpy system can precisely retrieve the victim's input regardless of its length or type. While ensuring that all the credentials were only known by the participants, we investigate the realism of our attack model for three login methods and illustrate the results in Figure 13.

1) Overall Performance of Login Attack on Physical Button: This login usually has two aspects sorted by the information type.

S2A: Password Length: Each participant was asked to input the password on the system login of MacBook Pro. Notably, the resulting text on the screen is only shown as an asterisk character. Thus, this attack aims to evaluate the performance of WaveSpy in detecting the password length. The length of the password is within a typical range of 1 to 16 [9]. Our results demonstrate that WaveSpy can precisely infer the password length with an average Top-1, Top-2, Top-3 accuracy up to 98.73%, 99.09%, and 99.56%, respectively, leading to a drastic reduction in the recognition period for the key information. Moreover, we also can recover keystroke timings, that contains substantial information about the password being typed, by continually recording the change of the typed password length and the accuracy for the keystroke timing inference is 99.96%.

S2B: Numeric Password: We instruct the participants to press the respective key on the numeric keyboard (i.e., 0-9) of a security intercom system, where the resulting 4-digit password is shown on the screen. In this attack, every password was input ten times. As observed in Figure 13, the average Top-1, Top-2 and Top-3 inference accuracy for the numeric password reaches up to 81.27%, 86.86%, and 90.03%, which significantly reduces the numeric password entropy, further discussed in Section X.

2) Overall Performance of Login Attack on Virtual Button: The information from virtual button can be represented in three subtypes as follows.

S2C: PIN: A four-digit PIN was fed by each participant for ten times to the PIN keyboard of iPhone 7 Plus. The average inference accuracy of Top-1, Top-2 and Top-3 is up to 80.09%, 84.49%, and 87.77%, respectively. A typical mispredicted example is the PIN '1258' is wrongly considered as '1268'. The reason is that the '6' button is near to '5', causing the similar *LCS response*. A similar phenomenon can be observed in S2D and S2E. Upon careful analysis, we examine that the performance of this attack is inferior compared to the numeric password (S2B), due to the smaller display area (see Section II-B) of the digital screen, which influences the characteristics of the received *LCS response*.

S2D: Pattern Lock: Each participant was required to draw 10 lock patterns on the pattern-lock keyboard of Nexus 5. The length of the lock pattern ranges from 1 to 6 units. For this attack, the average Top-1, Top-2 and Top-3 inference accuracy reaches to 77.81%, 81.49%, and 86.93%, respectively. The inference accuracy is slightly lower compared to previous four-digit PINs, as the UI correspondence of pattern locks changes little, increasing the challenge for WaveSpy to retrieve the sensitive information.

S2E: Password: A password generally comprises 26 letters and ten single-digit numbers. The participants were required to type on the alphabetical keyboard of MSI GL62. The length of the input varies from 1 to 8 characters. WaveSpy can infer passwords with the average Top-1, Top-2, Top-3 accuracy up to 70.12%, 75.72%, and 81.19%, respectively. In contrast to the PIN (S2C) and pattern lock (S2D), the password comprises numerous combinations of letters and numbers while having a longer character length, which affects the system performance. However, the observed accuracy is still within an acceptable range considering that the attacker can utilize other learning techniques to guess the misclassified characters.

3) Overall Performance of Login Attack on Picture Password: For the attack on **S2F: picture password**, every participant clicked the specific locations on the digital screen of Dell U2415 using a cursor. The Top-1, Top-2 and Top-3 accuracy is 61.31%, 63.49%, and 68.86%, respectively. For more than 40% retrieval taps, the distance estimation error is less than $5mm$ (1.9% of the screen side length), which is within the UI correspondence area. Lower performance is observed due to the miniature radius of UI correspondence (i.e., $6mm$) and a high tolerance of the password mechanism, which provides the users more freedom in selecting the specific location on the screen as an input.

In conclusion, our results demonstrate the effectiveness of WaveSpy to facilitate screen content type recognition and sensitive information retrieval under ideal conditions. We further explore the system performance against varying sensing parameters and real-world scenarios in the remaining sections.

VII. Evaluation II: Robustness Investigation

A. Impact of Sensing Distance and Device Orientation

In practical scenarios, the attacker should be able to keep a certain distance or an angle to avoid being discovered. Such a convenient practice, however, will lead to the changing distance and orientation between the screen and the mmWave probe. Therefore, it is important to investigate whether these aspects will affect system performance. Specifically, we measure the different device orientations (from 0° to 40°) at different distances (from $20cm$ to $180cm$). Following Section

V-B, we recollect the training and testing set. Three participants select 100 screen content types at a random sequence shown on the Dell U2415. The results are shown in Figure 14. The average Top-3 inference accuracy remains high when the sensing distance varies within $180cm$ (above 99.5%). As for the orientation, although the reflected signal slightly changes due to the different probe angles for each content type, the inter-type distinguishability among 100 screen content types is significant such that each device can be correctly recognized. Thereby, WaveSpy can facilitate portable and convenient screen attack in real practice.

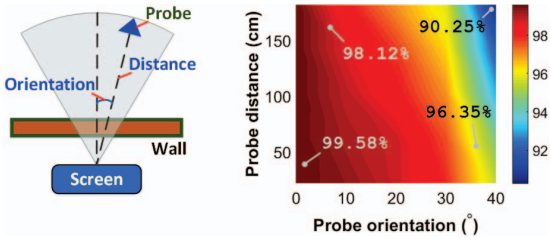


Fig. 14. The attack accuracy according to sensing distance (from $20cm$ to $180cm$) and device orientation (from $0°$ to $40°$) keeps over 90.25%.

B. Impact of Display Resolution

The display resolution is a crucial consideration in a real application, which is related to the screen type. Specifically, we recruit five different screens with four different display resolutions between 800×600 (VGA) to 3840×2160 (4K) pixels. For each resolution setting, we evaluate the screen content type inference following the preparation in Section V-B and re-prepare the training and testing sets. Figure 15 manifests that their performance of average Top-3 accuracy can achieve up to 99.52%. Besides, the identification results all remain above 99.4%. Hence, WaveSpy can maintain a high success rate in attacking screens under different display resolution setups.

C. Impact of Screen Model

Due to the fact that many attacks rely on the screen model, we simulate a scenario where the attacker lacks this prior knowledge. In this section, we evaluate the attack performance under the impact of the screen model to verify the training data generalization. To address this concern, we employ four devices for testing, including iPhone 6, iPhone 6s, Pixel 2 and MacBook Pro. We repeat the experiment of the screen content type inference (as described in Section V-B). Importantly, we still use the previous training data from iPhone 6. Notably, there are two iPhone 6 here, one for training, one for testing. As shown in Figure 16, the testing results illustrate the inference accuracy. We observe that the average Top-3 accuracy on iPhone 6 and iPhone 6s are the highest, 99.18% and 97.03% respectively, while others are both below 10%. The reason is that the tested iPhone 6 and iPhone 6s have an equal or similar hardware structure with the training device, which are entirely different from others. Moreover, the comparable accuracy on iPhone 6s testing indicates that our trained classifier does not have the over-fitting issue and can adapt to various usage scenarios. To sum up, results indicate that WaveSpy can work across different screens with the same or similar hardware structures (see a further discussion in Section X).

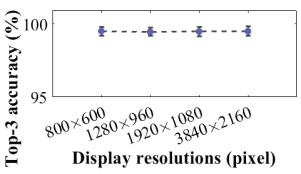


Fig. 15. Inference performance under different display resolutions.

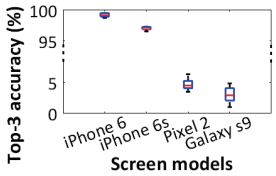


Fig. 16. Inference performance under different screen models.

D. Impact of Cover Material

We consider the scenario where the user hides the screen in other materials to evade attacks. Particularly, we collect five different daily-accessible materials (i.e., brick, glass, plastic, wood, curtain, cardboard). We place the screen behind each of them and evaluate the screen content type inference accuracy for all nine monitors. The performance is reported in Figure 17, where we can see that the overall accuracy for each is above 98%. Certain materials slightly affect the performance to some extent. This is because WaveSpy utilizes a high-frequency signal and therefore has a small wavelength and limited penetration ability. As a result, it is prone to the scattering reflection upon some specific materials. Generally, WaveSpy still provides reliable performance in screen content type recognition.

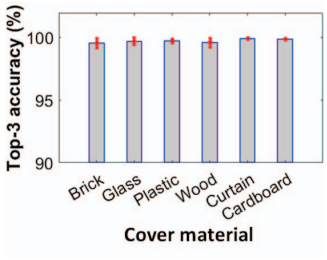


Fig. 17. Evaluation to determine the influence of cover material on the screen content type recognition.

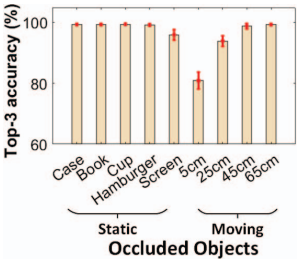


Fig. 18. The system performance for screen content type recognition under the influence of different surrounding objects.

E. Impact of Occluded Objects

In this experiment, we investigate the influence of the *static and moving* surrounding objects that affect the mmWave signals on the inference accuracy of screen content type. For *static objects*, we select device case, book, cup, hamburger, and an extra screen. For *moving objects*, we perform this experiment with two participants, i.e., a participant selects the screen content while another participant is moving with the same normal walking speed as the surrounding object at different distances away from the screen.

For *static objects*, the performance is reported in Figure 18, where we can see that the overall accuracy for each is above 98%, implying these surrounding objects have a limited effect on the performance. For *moving objects*, we can observe that the surrounding moving objects obviously

affect the inference accuracy, but, the effect decreases as the distance increases. When the distance between the object and the screen exceeds $45cm$, the influence of surrounding objects becomes negligible. This is because the mmWave wave has a high directionality and controlled sensing angle, decaying exponentially with respect to distance from the screen to the surrounding objects. This experimental result demonstrates that it is not easy to disrupt WaveSpy using surrounding objects.

F. Impact of Open World Scenarios

In real practice, the attacker may also aim to extract text from the screen. By referring to a recently publish work on screen attack [3], we conduct an experiment on Dell U2415 under the open-world setting to verify whether we can extract content from the screen. We collect 30 paragraphs and each paragraph contains at least 60 words. In trace, each character lasts 0.5s, typed on the virtual keyboard. Following Section V-B, we recollect the training and testing set. The results present the average Top-3 accuracy for word inference is 81.3%. For example, the word "implicitly" is incorrectly recognized as "inplicitly". Similar analysis is further discussed in Section VIII. This performance can further improve by coordinating with the dictionary [1]. This experimental result demonstrates that WaveSpy can perform the screen attack under different open world setups.

VIII. Evaluation III: Real-world Screen Attacks

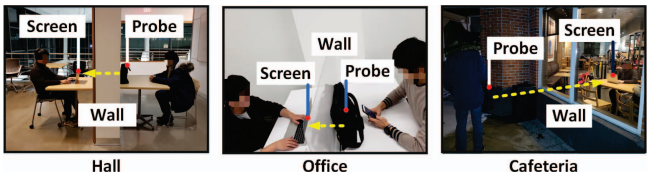


Fig. 19. The carry on attacks are conducted in three locations, i.e., hall, office and cafeteria on the Macbook Pro. The probe is hidden in a normal handbag arousing no suspicion to victim and nearby surrounding.

Experimental Setup: Due to the portability and low cost of the setup, eavesdroppers can access a target screen used in public spaces. Therefore, we conduct a real-world screen attack. The studied sites involve three common locations in daily life (i.e., offices, hall and cafeteria) as shown in Figure 19. For each site, the participants were instructed to use the digital screen placed behind the wall or obstacle. The content type and the sensitive information are displayed on the screen at a default font (i.e., 9-12pt). It is important to note that these sites are different from the environments described in Section III-B where we collected the prior data and characterized the *LCS response*.

Evaluation Results: Table II shows case studies for four attack trials on the screen content type recognition and sensitive information retrieval, including their corresponding ground truths. We can see that, in MS Word types, it was wrongly recognized as MS Visio at the cafeteria location. The reason is that these two types have a considerably similar layout, confusing the classifier. The password mistake happens at the cafeteria, where recognizes the 'c' into 'v'. It is because these two characters have adjacent locations on the screen, leading to similar *LCS responses*. Although in the sentence retrieval, the results are not as good as a PIN, it still shows a huge potential for the sensitive sentence or content eavesdropping. Besides, we also conduct a sustained attack on the screen content type recognition, in an attempt to acquire the user activities usage statistics. The usage statistics analysis for three hours at the office location with Top-1 accuracy is shown in Figure 20. Note that our WaveSpy can be applied to monitor the user activities for a long time with high inference accuracy 96.2%. Though some performances appear lower than those of the above performance, we can improve them by adjusting the characteristics of the antenna according to the screen position. In contrast, if eavesdroppers identify the screen to be attacked in advance, they can optimize their setup according to profiling results. Moreover, an optimized antenna makes the maximum stealing distance much longer. Thus, the result suggests that our system provides reliable performance in real practice.

TABLE II
Error Examples of the Real-world Attack at Three Different Locations Against the Ground Truth.

Attack Scenario	Attack Results on Different Locations			Ground Truth
	Hall	Office	Cafeteria	
#1	MS Word	MS Word	MS Visio	MS Word
#2	a1b2c3	a1b2c3	a1b2v3	a1b2c3
#2	Good Night	Good Night	Good Nitht	Good Night
#2	Have A mice Day	Have A Nict Day	Habe A Nocw Day	Have A Nice Day

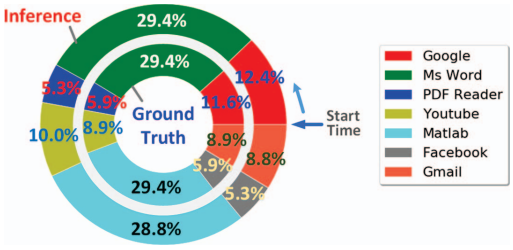


Fig. 20. Usage statistic analysis of on-screen content type recognition for 3 hours at an office location. The inner loop indicates the ground truth while the outer loop demonstrates the usage statistics inferred from WaveSpy.

IX. Countermeasures

Creating a large isolation zone (e.g., over ten thousands of square feet) is effective to defend most of the screen attacks, including WaveSpy. However, it is not practical (e.g., cost and usability) in real-world scenarios. In this section, we will discuss two sets of practical countermeasures against the WaveSpy attack. The first countermeasure set is cost-effective, altering either hardware or user behavior to mitigate the security risk. The second countermeasure set is zero-cost,

a purely software-based solution with no hardware or user cooperation requirement.

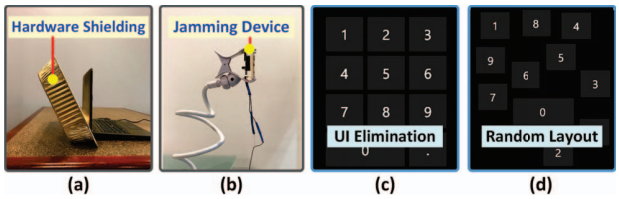
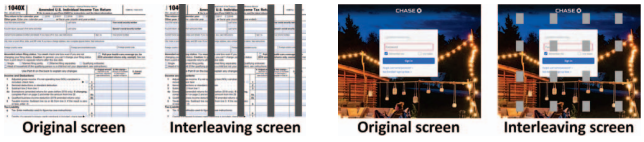


Fig. 21. Examples of countermeasure solutions: (a) conductive hardware shielding; (b) side-channel inference with a jamming device; (c) corresponding UI elimination towards button touch; (d) randomized keyboard layout.

Cost-effective Approaches: To counter the attack, we introduce a cost-effective solution set including four protection strategies in Figure 21. In general, we explore the mmWave signal transmission drawback, and thus a shielding technique is proposed to isolate electrical devices from the "outside world" as shown in Figure 21(a). However, if the shield covers the full display surface, the usability of the screen drops significantly. Besides, deploying the shield needs extra human labor and increases the cost. Another possible way to avoid the attack from the mmWave is to make use of the receiving channel limitations. We employ a wireless jamming device that continuously transmits noise signals to block the probe receiving channel as shown in Figure 21(b). Yet, in real practice, such an approach is hard to achieve, since the jamming device needs to know the attack frequency in advance. In addition, a straightforward countermeasure is to focus on the *LCS response* inhibition. We automatically prevent the usage of the UI reminder when inputting sensitive information, i.e., making no change on the screen, as shown in Figure 21(c). Also, another sophisticated defense exploits the same principle, which is to randomize the layouts of the keyboard grid as shown in Figure 21(d). However, both countermeasures can dramatically decrease the user experience.



(a) Tax return form (b) Credit card login

Fig. 22. Two examples of the countermeasures with the interleaving screen.

Zero-cost Approach: Defense approaches above require either additional hardware or user behavior changes. In this part, we investigate a novel defense approach with zero cost, namely high-frequency interleaving screen. This approach exploits the display mechanism and leverages the RF probe sensing limitation. As indicated in Section VI, the least *LCS response* duration for attacking lasts from 40 to $100ms$, equal to 10-$25Hz$, while at the same time, the refresh rate on modern display is usual higher than $60Hz$. Since screen refresh rate is higher than probe sensing frame rate, we can scribble multiple frames (e.g., adding full-screen flicker marks) within the frame periods to deter attacking, while preserving viewing experience by taking advantage of human eyes flicker fusion effects [42], [43]. Two examples of flicker marks are illustrated in Figure 22. We recollect 20 screen content types with the flicker makers as the testing data, combined with the training data in Section V-B. The results with the flicker mark demonstrate the average Top-3 accuracy of 3.7%, which confirms the feasibility of this protection.

X. Discussion

System Limitations: WaveEye realizes a new remote screen monitoring, and there are several system limitations. First, as discussed in Section VII-C, although we can generalize the training data among the same screen type, it is hard to build a general model among different screen types. The reason is that hardware structures and display designs of different screens are different, which causes completely distinct *LCS responses* (see Figure 7). It is impossible to build the general model and attack the screen by discovering the collaborative internal features among these responses. Besides, we notice, currently, no solution can achieve the real remote image visualization (see Section XI). Among all of the alternatives, we believe the solution based on RF technology is the most promising. We propose to prepare a customized model for each pixel value on the screen. For example, if the resolution is 800×600, we can prepare 540,000 models totally and predict the screen image after integrating the simultaneously predicted value of each pixel. This solution is inevitably limited by the current hardware probe technology. To conduct such pixel-level monitoring, a significantly higher resolution is required, and consequently it is urgent to develop a RF probe utilizing a higher carrier frequency (e.g., Terahertz technology, larger than 100 GHz [44]. Nevertheless, this technology continues to mature at present [45]. Besides, to support this attack, the amount of sensing data is huge, and thus we need an ultra-high speed and larger operation bandwidth analog-to-digital convert (ADC) (e.g., larger than 50 GS/s operation speed), while such ADC technology is still under the research phase [46]. As aforementioned, we believe it could be soon achieved in the coming future.

Sensitive Information Retrieval Analysis: We further analyze our system from two aspects: time complexity and entropy reduction. The strength of a password is a function of length, complexity, and unpredictability, which depends on the resistance to brute-force guessing attacks. Entropy is the typical measure of password strength. For a password X, its entropy is defined as $H(X) = -\sum_{i=1}^{n} P(x_i) \cdot \log_2 P(x_i)$, where $x_i (i \in \{1, 2, ..., n\})$ is one of n possible values of the password X, and $P(x_i)$ represents the probability that $X = x_i$ holds. Considering a keyboard housing t characters, a random password with length N has t^N possible values with $\mathcal{O}(a^n)$ $(a > 1)$ time complexity and $N \cdot \log_2 t$ bits of entropy. Compared with brute-force guessing attacks, our unobtrusive sensitive information retrieval attack greatly decreases both the time complexity and password entropy. As we evaluated in Section VI-B, the time complexity decreases to $\mathcal{O}(n) \sim \mathcal{O}(a^{0.2n})$, and the entropy drops to $0.2N \cdot \log_2 10 \sim 0.3N \cdot \log_2 36$ bits according to different applications.

The Stealthiness of WaveSpy: The stealthiness of WaveSpy is a critical factor in analyzing the feasibility of the attack in practice. The designed probe must guarantee to be undiscovered. To achieve this goal, the designed mmWave probe uses the invisible mmWave signals to attack, which is $11.8cm$ $(4.65in) \times 4.5cm$ $(1.77in) \times 1.5cm$ $(0.59in)$ with the weight only $45.4g$ and can be fueled by a common portable power bank. The output signal can be connected to the audio interface of a smartphone or a tablet for signal processing. Notably, the trace for N-digit password only needs the data storage of NKB around.

XI. Related Work

Electronic Device Emanations: The electronic device functioning of various state-of-the-art sensors leaks critical information that can be acquired to infer application usage and screen activity. Previously, researchers have explored the threat of keystroke inference attacks based on observing the motion [47] or multiple sensor data in the device [48], [49], [50], [51], [52] and tablet backside motion patterns through vision-based monitoring [53], [54], [55]. Given the adversary has access to the target electronic device, the smudges on the screen can be investigated to construct critical information about recent user activity [56]. However, these attacking solutions cannot work in our attack model without line-of-sight. The intensity of light emitted from the cathode-ray tube (CRT) displays can be analyzed to reconstruct the text information shown on the display; however, it is only feasible in dark environments without the interference from other lighting sources [10]. Furthermore, the digital screens leak electromagnetic (EM) or other emanations that can be exploited by an adversary to steal the information displayed on the screen or from a login [2], [57], [9], [58], [3]. However, this type of attack highly depends on the power supply of the screen. Along with the power management development, there is a visible trend that the low-cost technology will be widely deployed in most screens, and thus the scaling of these emanations decrease dramatically making emanation-based attacks fail. In addition, in EM strategy, the attacker must be extremely close to the victim screen and acoustic-based solutions require no occlusion or obstacle, which hardly work under the setting of this paper. Besides, it is worth mentioning that although some EM-based attacks have tried to visualize the results by combining the predicted results with the pre-capture screen image [2], the feasibility of the remote image visualization rests on the assumption that the attacker gets the pre-capture screen image of the victim, which is not the real image reconstruction. As aforementioned, these attacking strategies cannot work under the setting in this study.

Compromising Reflections: The sensitive information displayed on the digital screens to the user cannot be extracted from only be device side-channels, but also the screen's optical emanations on nearby objects. A novel screen-based attack was presented which exploits the comprising reflections on the objects (e.g., eyeglasses, teapots) that are in proximity to the screen posing a significant threat to the privacy of the information displayed on the screen [59]. Even the diffused reflections from a wall or shirt can be employed from the reconstruction of the projected image using a digital camera [60]. Another form of compromising reflections can be obtained by tracking the diverging positions of victim's fingers during typing while they are reflected from proximity objects or even obtainable from long-distance view [61], [8], [62], [1]. All the work above is ineffective in our attack model.

Remote mmWave Sensing: In the last decade, mmWave radars have been extensively employed in both research and practice to detect the target's inherent motion (e.g., cardiorespiratory and gesture sensing [63], [64], [65]) for vital signs monitoring and user authentication. Studies have demonstrated the feasibility of remotely detecting the hand motions and physiological features, such as heart rate and breathing patterns [66], [67]. However, given that the underlying characteristics of the mentioned applications rely on Doppler motion, they cannot be directly applied to sense through the target or other obstacles (e.g., packages and luggage). While some researchers [68], [69], [70] explore the propagation of the mmWave through-wall and through-objects, the systems are still inapplicable for a target with specific mmWave-absorption characteristics. To the best of our knowledge, the proposed WaveSpy is the first non-contact mmWave sensing application that aims to exploit the *LCS response* to achieve the screen attack through the occlusion.

XII. Conclusion

In this paper, we first identified and validated a new and yet practical side-channel to infer contents on digital screen via the liquid crystal nematic state sensing in isolation scenarios. We started from basic functioning mechanism and LC nonlinear effect in digital screens on the personal device and analyzed the *LCS response*. Then, we designed a portable, low-cost, and energy-efficient 24GHz mmWave probe and proposed a novel end-to-end deep learning-based hierarchal module to recognize the screen content type and retrieve the sensitive information on digital screens. Furthermore, extensive experiments indicated that the proposed WaveSpy achieves more than 99% inference accuracy through-wall within $5m$ distance with a centimeter level screen solution. The Top-3 sensitive information retrieval rate of the proposed WaveSpy is up to 87.77%. Various levels of evaluation proved the robustness, reliability, and efficiency of our proposed WaveSpy. Finally, we recommend that privacy-sensitive systems should pay considerable attention to this new side-channel and increase the screen security (e.g., flicker mark).

Acknowledgment

We thank our shepherd, Dr. Matthew Hicks, and all anonymous reviewers for their insightful comments on this paper. We also appreciate the suggestion and hardware support from Prof. Changzhi Li in Texas Tech University. This work was in part supported by the National Science Foundation under grant No. 1718375.

REFERENCES

[1] Y. Chen, T. Li, R. Zhang, Y. Zhang, and T. Hedgpeth, "Eyetell: Video-assisted touchscreen keystroke inference from eye movements," in *2018 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2018, pp. 144–160.

[2] Y. Hayashi, N. Homma, M. Miura, T. Aoki, and H. Sone, "A threat for tablet pcs in public space: Remote visualization of screen images using em emanation," in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2014, pp. 954–965.

[3] D. Genkin, M. Pattani, R. Schuster, and E. Tromer, "Synesthesia: Detecting screen content via remote acoustic side channels," *arXiv preprint arXiv:1809.02629*, 2018.

[4] M. Enev, S. Gupta, T. Kohno, and S. N. Patel, "Televisions, video privacy, and powerline electromagnetic interference," in *Proceedings of the 18th ACM conference on Computer and communications security*. ACM, 2011, pp. 537–550.

[5] *2018 Identity Fraud: Fraud Enters a New Era of Complexity*. https://www.javelinstrategy.com/coverage-area/2018-identity-fraud-fraud-enters-new-era-complexity, Accessed: 2019-3-14.

[6] *2017 Data Breaches - The Worst Breaches, So Far*. https://www.identityforce.com/blog/2017-data-breaches, Accessed: 2019-3-11.

[7] M. Kumar, T. Garfinkel, D. Boneh, and T. Winograd, "Reducing shoulder-surfing by using gaze-based password entry," in *Proceedings of the 3rd symposium on Usable privacy and security*. ACM, 2007, pp. 13–19.

[8] R. Raguram, A. M. White, D. Goswami, F. Monrose, and J.-M. Frahm, "ispy: automatic reconstruction of typed input from compromising reflections," in *Proceedings of the 18th ACM conference on Computer and communications security*. ACM, 2011, pp. 527–536.

[9] S. Fang, I. Markwood, Y. Liu, S. Zhao, Z. Lu, and H. Zhu, "No training hurdles: Fast training-agnostic attacks to infer your typing," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2018, pp. 1747–1760.

[10] M. G. Kuhn, "Optical time-domain eavesdropping risks of crt displays," in *Security and Privacy, 2002. Proceedings. 2002 IEEE Symposium on*. IEEE, 2002, pp. 3–18.

[11] J. A. Castellano, *Liquid gold: the story of liquid crystal displays and the creation of an industry*. World Scientific, 2005.

[12] A. Dhekne, M. Gowda, Y. Zhao, H. Hassanieh, and R. R. Choudhury, "Liquid: A wireless liquid identifier," in *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys '18. New York, NY, USA: ACM, 2018, pp. 442–454. [Online]. Available: http://doi.acm.org/10.1145/3210240.3210345

[13] Z. Li, Z. Yang, C. Song, C. Li, Z. Peng, and W. Xu, "E-eye: Hidden electronics recognition through mmwave nonlinear effects," in *Proceedings of the 16th ACM Conference on Embedded Networked Sensor Systems*. ACM, 2018, pp. 68–81.

[14] C. Wang, J. Liu, Y. Chen, H. Liu, and Y. Wang, "Towards in-baggage suspicious object detection using commodity wifi," in *2018 IEEE Conference on Communications and Network Security (CNS)*. IEEE, 2018, pp. 1–9.

[15] R. Efron, "Conservation of temporal information by perceptual systems," *Perception & Psychophysics*, vol. 14, no. 3, pp. 518–530, 1973.

[16] M. R. Fernández, E. Z. Casanova, and I. G. Alonso, "Review of display technologies focusing on power consumption," *Sustainability*, vol. 7, no. 8, pp. 10 854–10 875, 2015.

[17] D. C. Thompson, J. Papapolymerou, and M. M. Tentzeris, "High temperature dielectric stability of liquid crystal polymer at mm-wave frequencies," *IEEE Microwave and wireless components letters*, vol. 15, no. 9, pp. 561–563, 2005.

[18] E. C. Economou, "Liquid crystal based tunable bandpass and bandstop filters for millimeter wave signal processing applications," Ph.D. dissertation, University of Colorado at Colorado Springs, 2017.

[19] R. H. Chen, *Liquid crystal displays: fundamental physics and technology*. John Wiley & Sons, 2011.

[20] R. Camley, Z. Celinski, Y. Garbovskiy, and A. Glushchenko, "Liquid crystals for signal processing applications in the microwave and millimeter wave frequency ranges," *Liquid Crystals Reviews*, pp. 1–36, 2018.

[21] A. C. Polycarpou, M. A. Christou, and N. C. Papanicolaou, "Tunable patch antenna printed on a biased nematic liquid crystal cell," *IEEE Transactions on Antennas and Propagation*, vol. 62, no. 10, pp. 4980–4987, 2014.

[22] M. Yazdanpanahi, S. Bulja, D. Mirshekar-Syahkal, R. James, S. E. Day, and F. A. Fernandez, "Measurement of dielectric constants of nematic liquid crystals at mm-wave frequencies using patch resonator," *IEEE Transactions on Instrumentation and Measurement*, vol. 59, no. 12, pp. 3079–3085, 2010.

[23] D. R. Jackson, A. A. Oliner, and C. Balanis, "Modern antenna handbook," in *Leaky-Wave Antennas*. Wiley, 2008.

[24] A. Singh and V. Lubecke, "A heterodyne receiver for harmonic doppler radar cardiopulmonary monitoring with body-worn passive rf tags," in *Microwave Symposium Digest (MTT), 2010 IEEE MTT-S International*. IEEE, 2010, pp. 1600–1603.

[25] B. R. Mahafza, *Introduction to radar analysis*. Chapman and Hall/CRC, 2017.

[26] Z. Peng, L. Ran, and C. Li, "A 24-ghz low-cost continuous beam steering phased array for indoor smart radar," in *2015 IEEE 58th International Midwest Symposium on Circuits and Systems (MWSCAS)*. IEEE, 2015, pp. 1–4.

[27] J. Kim and A. F. Molisch, "Fast millimeter-wave beam training with receive beamforming," *Journal of Communications and Networks*, vol. 16, no. 5, pp. 512–522, 2014.

[28] K. V. Mardia, "Measures of multivariate skewness and kurtosis with applications," *Biometrika*, vol. 57, no. 3, pp. 519–530, 1970.

[29] R. I. Davis, W. Qiu, N. J. Heyer, Y. Zhao, M. Q. Yang, N. Li, L. Tao, L. Zhu, L. Zeng, D. Yao *et al.*, "The use of the kurtosis metric in the evaluation of occupational hearing loss in workers in china: Implications for hearing risk assessment," *Noise and Health*, vol. 14, no. 61, p. 330, 2012.

[30] *Crest factor*. https://en.wikipedia.org/wiki/Crest_factor, Accessed: 2019-2-16.

[31] B. L'Huillier and A. Shafieloo, "Model-independent test of the flrw metric, the flatness of the universe, and non-local estimation of h0 rd," *Journal of Cosmology and Astroparticle Physics*, vol. 2017, no. 01, p. 015, 2017.

[32] Z. Qawaqneh, A. A. Mallouh, and B. D. Barkana, "Deep neural network framework and transformed mfccs for speaker's age and gender classification," *Knowledge-Based Systems*, vol. 115, pp. 5–14, 2017.

[33] S. Sinha, P. S. Routh, P. D. Anno, and J. P. Castagna, "Spectral decomposition of seismic data with continuous-wavelet transform," *Geophysics*, vol. 70, no. 6, pp. P19–P25, 2005.

[34] S. G. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 11, no. 7, pp. 674–693, 1989.

[35] U. Gudaru and V. B. Waje, "Analysis of harmonics in power system using wavelet transform," in *Electrical, Electronics and Computer Science (SCEECS), 2012 IEEE Students' Conference on*. IEEE, 2012, pp. 1–5.

[36] J. Jia, Z. Liu, X. Xiao, B. Liu, and K.-C. Chou, "ippi-esml: an ensemble classifier for identifying the interactions of proteins by incorporating their physicochemical properties and wavelet transforms into pseaac," *Journal of theoretical biology*, vol. 377, pp. 47–56, 2015.

[37] Z. Li, A. S. Rathore, C. Song, S. Wei, Y. Wang, and W. Xu, "Printracker: Fingerprinting 3d printers using commodity scanners," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2018, pp. 1306–1323.

[38] G. Hinton, L. Deng, D. Yu, G. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, B. Kingsbury *et al.*, "Deep neural networks for acoustic modeling in speech recognition," *IEEE Signal processing magazine*, vol. 29, 2012.

[39] S. Qian and D. Chen, "Joint time-frequency analysis," *IEEE Signal Processing Magazine*, vol. 16, no. 2, pp. 52–67, 1999.

[40] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks." in *CVPR*, vol. 1, no. 2, 2017, p. 3.

[41] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[42] L. Zhang, C. Bo, J. Hou, X.-Y. Li, Y. Wang, K. Liu, and Y. Liu, "Kaleido: You can watch it but cannot record it," in *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*. ACM, 2015, pp. 372–385.

[43] S. Zhu, C. Zhang, and X. Zhang, "Automating visual privacy protection using a smart led," in *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*. ACM, 2017, pp. 329–342.

[44] A. Y. Pawar, D. D. Sonawane, K. B. Erande, and D. V. Derle, "Terahertz technology and its applications," *Drug invention today*, vol. 5, no. 2, pp. 157–163, 2013.
[45] D. M. Mittleman, "Perspective: Terahertz science and technology," *Journal of Applied Physics*, vol. 122, no. 23, p. 230901, 2017.
[46] P. T. Kurahashi, C.-Y. Lu, T. S. Rane, C. F. Nieva-Lozano, and H.-J. Lee, "High-speed analog-to-digital converter," Mar. 14 2019, uS Patent App. 16/191,427.
[47] D. Balzarotti, M. Cova, and G. Vigna, "Clearshot: Eavesdropping on keyboard input from video," in *Security and Privacy, 2008. SP 2008. IEEE Symposium on*. IEEE, 2008, pp. 170–183.
[48] P. Marquardt, A. Verma, H. Carter, and P. Traynor, "(sp) iphone: decoding vibrations from nearby keyboards using mobile phone accelerometers," in *Proceedings of the 18th ACM conference on Computer and communications security*. ACM, 2011, pp. 551–562.
[49] A. Maiti, O. Armbruster, M. Jadliwala, and J. He, "Smartwatch-based keystroke inference attacks and context-aware protection mechanisms," in *Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security*. ACM, 2016, pp. 795–806.
[50] X. Liu, Z. Zhou, W. Diao, Z. Li, and K. Zhang, "When good becomes evil: Keystroke inference with smartwatch," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2015, pp. 1273–1285.
[51] G. Ye, Z. Tang, D. Fang, X. Chen, K. I. Kim, B. Taylor, and Z. Wang, "Cracking android pattern lock in five attempts," in *The Network and Distributed System Security Symposium*, 2017.
[52] A. Maiti, M. Jadliwala, J. He, and I. Bilogrevic, "Side-channel inference attacks on mobile keypads using smartwatches," *IEEE Transactions on Mobile Computing*, 2018.
[53] J. Sun, X. Jin, Y. Chen, J. Zhang, Y. Zhang, and R. Zhang, "Visible: Video-assisted keystroke inference from tablet backside motion." in *NDSS*, 2016.
[54] Q. Yue, Z. Ling, X. Fu, B. Liu, K. Ren, and W. Zhao, "Blind recognition of touched keys on mobile devices," in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2014, pp. 1403–1414.
[55] M. Negulescu and J. McGrenere, "Grip change as an information side channel for mobile touch interaction," in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, 2015, pp. 1519–1522.
[56] A. J. Aviv, K. L. Gibson, E. Mossop, M. Blaze, and J. M. Smith, "Smudge attacks on smartphone touch screens." *Woot*, vol. 10, pp. 1–7, 2010.
[57] M. Li, Y. Meng, J. Liu, H. Zhu, X. Liang, Y. Liu, and N. Ruan, "When csi meets public wifi: Inferring your mobile phone password via wifi signals," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2016, pp. 1068–1079.
[58] M. Zhou, Q. Wang, J. Yang, Q. Li, F. Xiao, Z. Wang, and X. Chen, "Patternlistener: Cracking android pattern lock using acoustic signals," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2018, pp. 1775–1787.
[59] M. Backes, M. Dürmuth, and D. Unruh, "Compromising reflections-or-how to read lcd monitors around the corner," in *Security and Privacy, 2008. SP 2008. IEEE Symposium on*. IEEE, 2008, pp. 158–169.
[60] M. Backes, T. Chen, M. Duermuth, H. P. Lensch, and M. Welk, "Tempest in a teapot: Compromising reflections revisited," in *Security and Privacy, 2009 30th IEEE Symposium on*. IEEE, 2009, pp. 315–327.
[61] Y. Xu, J. Heinly, A. M. White, F. Monrose, and J.-M. Frahm, "Seeing double: Reconstructing obscured typed input from repeated compromising reflections," in *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*. ACM, 2013, pp. 1063–1074.
[62] A. Al-Haiqi, M. Ismail, and R. Nordin, "The eye as a new side channel threat on smartphones," in *Research and Development (SCOReD), 2013 IEEE Student Conference on*. IEEE, 2013, pp. 475–479.
[63] F. Lin, Y. Zhuang, C. Song, A. Wang, Y. Li, C. Gu, C. Li, and W. Xu, "Sleepsense: A noncontact and cost-effective sleep monitoring system," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 11, no. 1, pp. 189–202, Feb 2017.
[64] M. C. Huang, J. J. Liu, W. Xu, C. Gu, C. Li, and M. Sarrafzadeh, "A self-calibrating radar sensor system for measuring vital signs," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 10, no. 2, pp. 352–363, April 2016.
[65] F. Lin, C. Song, Y. Zhuang, W. Xu, C. Li, and K. Ren, "Cardiac scan: A non-contact and continuous heart-based user authentication system," in *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '17. New York, NY, USA: ACM, 2017, pp. 315–328.
[66] J. Lien, N. Gillian, M. E. Karagozler, P. Amihood, C. Schwesig, E. Olson, H. Raja, and I. Poupyrev, "Soli: Ubiquitous gesture sensing with millimeter wave radar," *ACM Trans. Graph.*, vol. 35, no. 4, pp. 142:1–142:19, Jul. 2016. [Online]. Available: http://doi.acm.org.gate.lib.buffalo.edu/10.1145/2897824.2925953
[67] I. V. Mikhelson, S. Bakhtiari, T. W. E. II, and A. V. Sahakian, "Remote sensing of heart rate and patterns of respiration on a stationary subject using 94-ghz millimeter-wave interferometry," *IEEE Transactions on Biomedical Engineering*, vol. 58, no. 6, pp. 1671–1677, June 2011.
[68] F. Adib and D. Katabi, *See through walls with WiFi!* ACM, 2013, vol. 43, no. 4.
[69] Y. Zhu, Y. Zhu, B. Y. Zhao, and H. Zheng, "Reusing 60ghz radios for mobile radar imaging," in *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*. ACM, 2015, pp. 103–116.
[70] T. Wei and X. Zhang, "mtrack: High-precision passive tracking using millimeter wave radios," in *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*. ACM, 2015, pp. 117–129.

APPENDIX A
SUPPLEMENTAL MATERIALS

A. Impact of Screen Change Rate

In real practice, the user inputs the password on an electronic device at a moderate speed to avoid feeding any incorrect information. However, the preferred speed of an individual may still vary based on the comfortability with the password. Therefore, we examine the impact of typing speed on the inference accuracy of WaveEye. To acquire the data with different input speeds, we ask three participants to input the PIN information on an MSI GL62. The speeds can be categorized in three types, i.e., moving the fingertip moderately (average $0.9s/digit$), quickly (less than $0.6s/digit$), or slowly (more than $1.2s/digit$).

TABLE III
THE SYSTEM PERFORMANCE FOR SENSITIVE INFORMATION RETRIEVAL AT THREE DIFFERENT INPUT SPEEDS: QUICK, MODERATE AND SLOW.

Attack Num.	Quick	Moderate	Slow
Top-1 (%)	74.05	81.31	82.82
Top-2 (%)	79.26	85.09	86.16
Top-3 (%)	85.31	90.03	90.77

Table III demonstrates the inference accuracy under different input speeds with results showing that a quick speed can adversely influence the system performance. It is because when the PINs are fed too fast, the associated button UI correspondences are not completely displayed on the screen and the mmWave signal is unable to capture the salient characteristics of the respective button. However, the Top-3 inference accuracy of quick input speed also reaches up to more than 85%. In addition, the input speed of PIN in practice will not vary as extremely as considered in our experiments. Therefore, WaveEye is robust to the changes in input speeds.