# LightSpy: Optical Eavesdropping on Displays Using Light Sensors on Mobile Devices

Supriyo Chakraborty
IBM T. J. Watson Research Center

Wentao Ouyang
Institute of Computing Technology
Chinese Academy of Sciences

Mani Srivastava
University of California, Los Angeles

***Abstract*—Light emanations from flat-panel displays are a side channel hinting towards the displayed content. Optical eavesdropping requires sensors in the proximity of such displays, necessitating physical access to the the target's environment. This requirement may be eliminated by exploiting the light sensor on the target's mobile device, though there are significant challenges. Such sensors measure one-dimensional light intensity, provide no chromatic information, and have very low sampling rate (normally up to $10$Hz).**

**In this paper, we demonstrate that in spite of these challenges, it is possible — based on intensity measurements from a mobile device's light sensor — to make quality inferences regarding the displayed content. We do so by selecting features of measured light that capture information related to transitions between samples. Such features are resilient to ambient noise.**

**In our experiments, involving over $60$ hours of collected data and $140$ movie clips, we were able to (i) classify content into categories (game, movie, etc) with approximately $90\%$ and $70\%$ accuracy for two-class and four-class classification, respectively; and (ii) identify specific movies or TV programs being played with $> 85\%$ accuracy. These findings suggest that access to raw light-sensor readings, which can currently be done without special access controls, may carry nontrivial security ramifications.**

## I. Introduction

Mobile devices are often equipped with a variety of sensors and actuators, including microphone, camera, GPS system, accelerometer, gyroscope, etc. To prevent these sensors from being used as unintended side channels for inferring private information [25], [26], [10], [14], mobile platforms typically either ask the user to explicitly grant an app access to the requested capabilities at install time (e.g., via the Android permission model), or they alert the user when a certain action is about to be executed (e.g., by requiring an image preview when using the camera).

An interesting side channel, which has received little attention to date and falls largely within the category of capabilities that apps can access with stealth [2], [29], is light sensing. An app can obtain readings from the built-in mobile device's light sensor without mediation (i.e., the need for install-time permission). Furthermore, a survey of the top $758$ Google Play apps reveals that around $24\%$ of those apps require access to the camera at install time [37]. While Android requires a preview of the clicked image to be shown, through careful configuration an app can hide the preview, and instead use the camera to obtain continuous light measurements without being noticed.

**The Threat:** In this paper, we explore the security ramifications due to the ability to stealthily obtain light readings on a mobile device. Specifically, we investigate two types of threats: (i) *classification* of content displayed by a nearby flat-panel device according to a fixed set of categories (game, movie, work, etc) and (ii) *identification* of the specific content being displayed. From a privacy perspective, these are serious concerns. Content accessed by the user is often personal [28], correlated with the person's mood [35] and indicative of current activities [7], [6].

The primary scenario that we focus on is of a person whose mobile device is infected by a malign app, masquerading as a game or some other popular app, that regularly obtains light readings "under the radar". Importantly, there is no need for the attacker to be (physically) near the target. The app can simply collect the light data and transmit it to a remote party.

In this scenario, a privacy threat emerges if (i) the light sensor is directed toward, and/or is proximate to, the target's computer screen or another device that may display private, or otherwise sensitive, content; and (ii) raw light data collected throughout a short time period already suffices to characterize the displayed content. Under these circumstances, placing the phone into a mount or charging pod by the computer, for even a few seconds, already poses as a potential privacy risk.

**The Challenge:** Display units (e.g., cathode ray tubes (CRTs), LCD monitors, laptop screens and plasma displays) all emit radiation in the visible band of the electromagnetic spectrum [20], [19]. The overall light emitted by a display unit is an information carrier that transmits, via light intensity modulation, the applied video signal. However, utilization of the captured light for content reconstruction or identification relies heavily on the particular display technology as well as on the light sensor used to sample the emitted light.

In a CRT, which is a raster-scan device, the image to be displayed is transmitted and updated as a sequence of scan lines that cover the entire display with constant velocity. The above fact, together with a high-bandwidth light sensor (i.e., $> 100$ MHz), has been used — under sufficiently dark ambient conditions — to sample emitted light and reconstruct the screen content with high accuracy [18]. A high sensor sampling rate ensures that the high-frequency components in the emitted light are captured, allowing pixel-wise reconstruction of the image.

Our focus, instead, is on flat-panel displays (FPDs), such as

LCD monitors and plasma screens. FPDs are more prevalent than CRTs, and in particular, these are the displays that are commonly found nowadays in workstations and are built into laptop computers. FPDs mark a significant departure from CRT technology, and as such, present unique challenges for an optical eavesdropper [19], [20]. First, instead of sequential pixel updates, FPDs update all pixels in a row simultaneously, making it impractical to separate the contribution of individual same-row pixels to the overall light emitted. This protects against complete reconstruction of the screen content. Second, FPDs operate under low voltages, do not amplify the video signal as CRTs do, and are also better shielded (under Tempest guidelines [30]), resulting in an overall reduction in radiation levels. Finally, the digital video signals from these displays undergo a mapping from the bit pattern encoding the displayed color to the light value observed by the eavesdropper, which varies greatly with the environment.

These characteristics of FPDs all stand as challenges toward content reconstruction. Still, opportunities to infer sensitive information about the content remain.

**Our Approach:** We build on the insight that different content often translates into different light intensity patterns over time, which can be used to characterize the content. For example, the emitted light during movie watching changes rapidly in brightness due to frequent variations in scenes, which is not true of using an email client or a text editor.

Starting from this observation, we have identified several features that are resilient to attenuation of the received light signal with distance. We incorporate these features into a machine-learning setting to perform both coarse- and fine-grained classification of content types.

We demonstrate — using a realistic setting involving over $60$ hours of collected video data and a physical configuration simulating usage of a mobile device in a standard workstation — that within a period of only $10$ seconds, and using the relatively low-quality light sensor built into the Nexus 7 tablet device, our technique is already able to perform content classification into $\{$*Media*,*Non-media*$\}$ with approximately $90\%$ accuracy, and into $\{$*Webpage*,*Work*, *Game*,*Video*$\}$ with about $70\%$ accuracy. That is, the threat outlined above is real.

We further demonstrate, given a database of $140$ distinct movie clips, our ability to identify the names of movies or TV programs being played with $> 85\%$ accuracy. The intuition here is that the shape of the light intensity stream emitted from the same content is largely reproducible (even in the presence of noise, scaling and other distortions). This aggravates the threat even more. To reiterate, both of these results are based on the low-quality built-in light sensor. With the camera acting as the primary light sensor, more granular distinctions can be made and with much greater robustness, and so the results in this paper are a lower bound on the efficacy of light sensing as a side channel.

Our conclusion is that content inference and identification can be made with surprisingly high accuracy by analyzing collected light samples. To our knowledge, this is the first attempt to directly analyze the visible spectrum of PFDs and perform content characterization on its basis. Our results, and their security ramifications, bring into question the design choice of making light data accessible without mediation.

## II. Related Work

We group our survey of prior work into two broad categories: device emanation based attacks, and attacks mounted using sensors on mobile devices. On one hand, the categorization aims to highlight the rich body of work that exploits unintentional emissions to acquire sensitive information and on the other hand it is to emphasize that most of the attacks have relied on sophisticated sensors or receiver systems. We also note that some of the recent works in this space have started to use the mobile devices.

### A. Compromising Device Emanations

Most electronic devices emit unintentional signals that can be exploited by eavesdroppers to either completely reconstruct or, at the very least, gain coarse but useful information about various private data. One such consumer device is the ubiquitous flat-panel video display unit (e.g., LCD monitors, laptop screens, plasma displays). The display units emit compromising radiations in both the radio frequency (primarily $3$ MHz – $3$ GHz) and the visible ($385$ – $790$ THz) bands of the electromagnetic spectrum. Starting with the initial demonstration in [32], where the display of a cathode-ray tube (CRT) was successfully reconstructed at a distance using a *modified* TV receiver, the radio-frequency emanations from flat-panel video displays have also been well studied in [19], [20]. In [13], the electromagnetic interference signatures that the power supplies of modern TVs produce are used to determine the video content that is displayed. The signatures are discernible and are resilient to the presence of other noisy electronic devices connected to the same powerline. Similarly, in [15], smart meter data is used to identify the content playing on TV.

The security threats due to the optical radiation emitted from computer LED status indicators on data communication equipments have been studied in [22]. Using off-the-shelf equipments, reflections of the LCD screen's emanations on various objects such as eyeglass, teapots, even the eye of the user is used to recover the screen content at a distance of $10$m. This is different from our work. We utilize the optical signal radiated from the FPDs, and explore the light sensor available on the mobile devices.

Numerous other side channels have been explored, including using acoustic emanations from keyboards [8] and radio frequency emanations from wired and wireless keyboards [33] to infer keystrokes. Soft keyboards provide users with visual confirmation of keystrokes. The reflection of such visual cues from the users eyeball, sunglasses, mirrors or other objects in the room have been used in [36] to infer keystrokes.

### B. Attacks Using Mobile Phones

The ability to mount a privacy attack by installing a malicious app on a mobile phone helps achieve the element of stealth that is often difficult if external instrumentation of the

space is required for intercepting data. Most of these attacks are carried out using the benign sensors on the phones to which all the apps have unrestricted access. Recent works have shown that accelerometer together with gyroscope sensor can be used to perform keylogging (on both the softkeyboard on the phone [10], [27] and external keyboards [23]) and also to infer location [17] with over $80\%$ accuracy.

Acoustic emanations from the CPU of a computer have been used to infer the RSA encryption key using a nearby mobile phone [14]. Light as a side channel has been explored in [29], where the variation in light measurements caused by minor tilts and turns in mobile devices, enables inference of the user's PIN from among a set of PINs with $80\%$ accuracy (where success means guessing correctly within the first 10 attempts).

Our attack is different. We use the device's light sensor to capture the light emanated from an FPD towards inference of content type and content identification.

## III. Threat Model

The threat we focus on is compromise of user privacy via classification or identification of the content displayed on the user's device through processing of the light signal. Specifically, we focus on FPDs — the most common displays, used in virtually any workstation — and the ability to sense the light they emit via the sensors on a nearby mobile device [12].

The significance of this threat is amplified by the fact that none of the major mobile platforms places controls on access to the light sensor, and that light sensing is also possible via the cameras installed on the device. This implies that a malign application installed on the target's device can obtain light intensity readings without the user's awareness.

An adversary attempting to infer sensitive information about the content that the target is viewing on a display would operate as follows. First, the adversary will have a malicious app, tasked to collect light readings, installed on the mobile device belonging to the target. The malicious app can be bundled together with a popular game or any other app. Second, the app will record light values in the background. This operation need not occur continuously, but can instead fire asynchronously in response to timer-, call- or message-based wakeups or periodically via a control loop. Finally, the malicious application will (opportunistically) send the collected data out to the adversary directly via the internet or through any other communication channel.

We note, in conclusion, that though we focus on inference of content displayed on an FPD, there are broader privacy threats due to light sensing. Other forms of inference — which are perhaps not eminent privacy threats, but can serve as the basis for other attacks — are whether the target is indoors vs outdoors, walking vs standing or sitting, etc [24].

## IV. Description of Attacks

In this section, we present an overview of the two types of threats considered, namely *inference* and *identification* of private content. We then explain how raw light intensity data should be processed to mount the attacks. The attacks themselves are described in Sections V and VI in turn.

### A. Overview

An *inference* attack aims to classify the content that the target is viewing according to a fixed set of categories, e.g. as being a movie, a news article or a webpage. An inference attack consists of two phases: a learning phase and an attack phase. The adversary first defines the classes (or categories) of interest; then collects light readings per these classes from any available source, extracts features and trains a general-purpose supervised classifier; and finally, in the attack phase, the adversary collects light readings from the target, extracts features, and feeds this input into the trained classifier to infer the content type.

An *identification* attack aims to identify whether the target is viewing content of interest, such as a specific movie or TV channel. Here, instead of supervised learning, the adversary utilizes a database. This attack consists of two phases: a preparation phase and an attack phase. First, the adversary collects light readings for specific content of interest, and persists this information in a database. Then, in the attack phase, the adversary collects light readings from the target, and performs matching between the data from the target and the database records.

### B. Obtaining Light Readings

On Android, an app can access the light sensor readings using the platform-provided APIs [1]. The scalar value represents the intensity of light (in units of lux) and does not provide any chromatic information. Access to the light sensor is not mediated by the Android manifest (or any other access-control subsystem components). Thus, ambient light measurements can be obtained by an app without the user being aware of it. The light sensor consumes very low energy (cf. [24]), it is effectively always turned on, and it is standardly used by system apps to adjust the display backlight based on ambient light conditions.

In addition, (front and rear) cameras on the mobile device can also be configured to act as light sensors. However, to access the camera an app requires install-time permission from the user. In addition, the camera device, when accessed through the platform-provided API, mandates that a preview using `SurfaceView` (an area on the screen where the captured image is displayed to the user) be associated and initialized before the image is captured. Using the camera in the prescribed manner would alert the user when an image is taken and remove the element of stealth from the attack.

To overcome this restriction, we have discovered a specific configuration in which we associate a `SurfaceView` with the Android `WindowManager`, and set the size of the view to $1 \times 1$ pixels. Then, the `PixelFormat` of the view is set to `Transparent` mode [5] to hide it completely from the user. This specific configuration ensures compatibility of the malicious app with most versions of Android while removing the preview from the display.

Using this, an app with access to the camera can periodically (to conserve power) take pictures without the user being aware of it. Finally, the exposure value of the captured image can be


(a) Original light intensity readings

(b) Normalized light intensity readings

Fig. 1. Original and normalized light intensity readings when the light sensor is placed 20cm, 40cm, 60cm, 80cm and 100cm away from the target display (playing the same video clip)

computed from the camera object parameters [3], and serve to approximate the ambient light under which the image was taken. In fact, a popular Google Play app [4] allows users to configure their camera as a light sensor to automatically control their backlight settings.

### C. Preprocessing

Before an attack can be launched, a preprocessing procedure is necessary to prepare the data. It consists of two steps: framing and normalization.

First, the light intensity stream is segmented into frames, each consisting of $w$ seconds, to facilitate feature extraction, classification and matching. Consecutive frames are shifted by $w/2$ seconds.

Second, the light-sensor readings in each frame are normalized to be of unit norm. This is because the received light intensity is highly impacted by the distance and angle with respect to the target display. Figure 1(a) presents light-sensor readings over a period of 60 seconds when the light sensor is placed 20cm, 40cm, 60cm, 80cm and 100cm away from the target display (playing the same video clip). Interesingly, as the figure illustrates, light intensity decreases significantly with distance, but the shapes of the data streams collected at different distances are similar.

To mitigate the effect of distance, we normalize the data in each frame to be of unit norm. The normalized light readings are shown in Figure 1(b). They become much more similar, and are thus much less sensitive to distance. Different angles have a similar effect on light intensity, with the highest and lowest intensity levels occurring when the sensor is facing, and is parallel to, the target display. Normalization can also mitigate the angle effect. (Figures omitted for space.)

## V. Inference of Private Information

As stated above, an inference attack consists of a learning phase and an attack phase. We describe each in turn.

### A. Learning Phase

The learning phase comprises three steps: data collection, feature extraction and supervised learning.

**Step 1: Data Collection:** Once the categories for inference are decided (e.g., media vs non-media), the adversary collects light data corresponding to them. There are ample sources to harvest data from, including e.g. video sharing websites like YouTube, news websites and channels, providers of streaming content like Netflix, etc. Hence, data collection is a straightforward task. In addition, collecting data from these general sources mitigates the risk of overfitting.

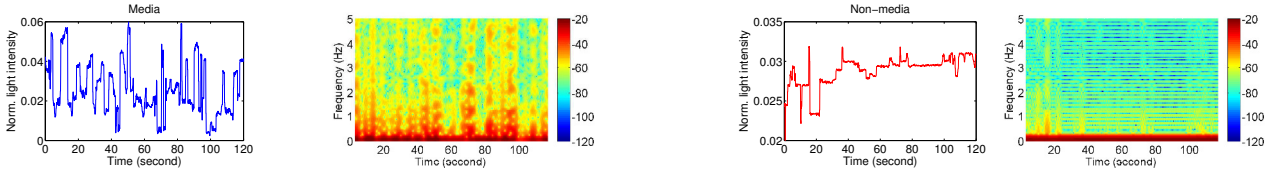
**Step 2: Feature Extraction** The most critical aspect of classification, or inference, is the feature set. Figure 2 depicts illustrative light-sensor readings as well as the corresponding spectrograms (representations of the spectrum of frequencies in a signal as they vary with time) when a user is viewing media-related content (e.g. a movie) vs non-media-related content (e.g. `amazon.com`).

Notice that the light intensity readings when viewing media-related content exhibit higher variance, a wider range and a more jerky trend compared to reading due to non-media-related content. Additionally, the spectrogram when viewing media-related content reflects many more high-frequency components over time compared to non-media-related contents.

These observations suggest distinct patterns to differentiate between the two classes, which also have intuitive justification: Movie scenes may feature significant sub-second changes in brightness., whereas browsing a website like Amazon gives rise to much slower and smoother changes in light intensity.

Based on the above analysis, we have identified the following *temporal* (first five) and *frequency* (last three) features:

1) Range: the difference between the maximum and minimum normalized light intensity. As Figure 2 suggests, media-related content yields a wider range than non-media-related content.
2) Standard deviation: variation of the normalized light intensity over the sample mean. Media-related content typically yields greater standard deviation.
3) Mean absolute derivative: the average of the absolute differences between consecutive normalized light intensity samples. While standard deviation compares each reading with the sample mean, this metric compares each reading with its preceding reading. Larger value for mean absolute derivative means that readings change sharply and rapidly, which is more characteristic of media-related content.
4) Mean crossing rate: the rate at which the normalized light intensity crosses the sample mean. Media-related content normally results in larger mean crossing rates. To reduce noise caused by small-scale variations around the mean, we only consider a difference of at least $2 \times 10^{-4}$ as crossing the mean.
5) Skewness: the asymmetry of the distribution of normalized light intensity around the sample mean. A negative value indicates that the signal values are spread more to the left of (or below) the mean, and vice versa for a positive value. Media-related content often features dark scenes, where bright scenes spread the distribution above




(a) (b)

Fig. 2. Raw (normalized) light intensity readings and corresponding spectrograms over 2 minutes displaying (a) media-related vs (b) non-media-related content

the mean. The opposite is characteristic of non-media-related content, where the background is often bright.

6) Entropy: the entropy of the spectrum distribution in terms of the Fast Fourier Transform (FFT) of the signal. This is a measure of how flat the spectral distribution is. Media-related content normally yields highly variable normalized light intensity that covers multiple different frequency bands, and thus has higher entropy. Non-media-related content, on the other hand, corresponds to low frequencies and thus lower entropy.
7) High-frequency energy ratio: the ratio of energy in the frequency bands $> 2.5$Hz to the total energy of the signal. We picked $2.5$Hz as the threshold, since our light sensor can only sample at up to $10$Hz, and so the FFT can cover up to $5$Hz. We utilize energy ratio, rather than absolute energy, to mitigate bias due to different signal magnitudes. Media-related content yields light intensities that cover more high-frequency bands, and so greater high-frequency energy ratios.
8) Spectral centroid: the balancing point of the spectral power distribution [21], defined as $sc = (\sum_{i=1}^{n} i \times p_i^2) / \sum_{i=1}^{n} p_i^2$, where $p_i$ is the $i$-th frequency bin in the computed FFT spectrum. (The DC component is not considered.) Media-related contents usually involves more high-frequency components, which push the spectral centroid higher.

**Step 3: Supervised Learning:** The above features are organized into a feature vector, where we extract a vector from each frame. The next step, given the per-frame vectors, is to build a classification model. To this end, the attacker can utilize an off-the-shelf framework for supervised learning [9], [16].

### B. Attack Phase

The steps in the attack phase are similar to those in the learning phase, including data collection, feature extraction and classification. First, the malicious application collects light intensity readings from the target's display. Different from that in the learning phase, these collected data are unlabeled. The adversary then processes the data into frames, and extracts the same features for each light data frame as in the learning phase. Finally, the adversary applies the trained classifier to each of the feature vectors to infer the content type.

## VI. Identification of Specific Content

An identification attack aims to identify the particular content that a target is viewing, such as a specific movie or TV channel out of a database of candidate content. This attack

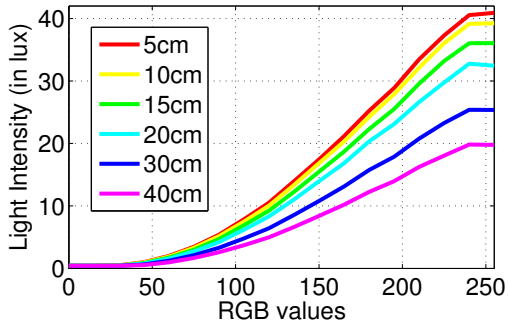



Fig. 3. Mapping between average RGB values for different grayscale images and observed light intensity values from a light sensor on a Google Nexus 7 tablet (The curves correspond to tablet placement at varying distances from the display.)

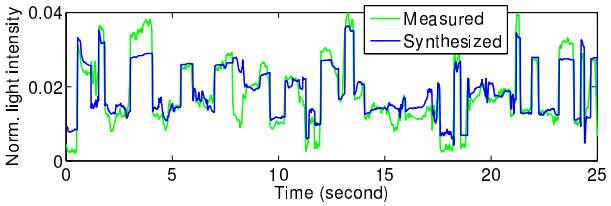



Fig. 4. Comparison between light intensity values generated synthetically vs through a light sensor

consists of a preparation phase and an attack phase, which we discuss in turn.

**Preparation Phase:** An identification attack requires a database containing the candidate contents and their identities. The adversary can prepare the database either *offline* (by playing the content of interest and collecting the corresponding light intensity readings) or *online* (e.g., by collecting light intensity readings from currently showing TV programs). The light intensity readings corresponding to content of interest then form a sequence in the database. Each sequence is labeled with the corresponding identity.

Yet another possibility is to prepare the database by approximating the light intensity values directly from RGB values encoded for the pixels of the target content. In this way, an adversary can efficiently create a large database without actually rendering the movies. This assumes a mapping between the average RGB values applied to the display and the light intensities measured by the sensor. We perform experiments to characterize such mappings by rendering known grayscale images on the display and measuring the light intensities at varying distances. The plots due to a Galaxy Nexus 7 tablet are in Figure 3. The horizontal axis is RGB values from 0 to

255, and the vertical axis denotes light intensity (in lux).

We have compared our synthesized light intensity readings with those collected by a light sensor across 40 movie clips (Figure 4). The average Pearson correlation coefficient between a synthetic light intensity stream and that collected by a light sensor 40cm away from a display is 0.8523. This number is very close to the average Pearson correlation coefficient between two sets of light sensor readings collected at different locations, which is 0.8802, confirming that this method is both efficient and reliable.

**Attack Phase:** The attack phase contains two steps: data collection and data matching. Data collection is the task of recording light intensity readings $l$ from the target's display for $w$ seconds. Matching is the subsequent act of comparing between the collected data and the light intensity sequences in the database.

For matching, we utilize a *matched filter*, which is obtained by correlating a known signal, or template, with an unknown signal to detect the presence of the template in the unknown signal [31]. In our case, for each sequence $l_c$ with identity $c$ and sample index $i$, the matched filter extracts a chunk $l_c(i)$ of duration $w$. It then computes the Pearson correlation coefficient as the match score $s(l, l_c(i))$ between the two sequences. (Normalization is not necessary, since the coefficient is not affected by it.) The match score $s(l, l_c(i))$ is computed for all the chunks extracted from the database sequence $l_c$ (i.e., all suitable values of $i$). The final matching score between sequences $l$ and $l_c$ is then the maximum across all chunks: $s(l, l_c) = \max_i s(l, l_c(i))$. Finally, the identity corresponding to the light intensity sequence $l$ from the target's display is determined as follows:

$$id = \begin{cases} \arg\max_c s(l, l_c) & \text{if } \max_c s(l, l_c) \geq \theta \\ \text{none} & \text{otherwise} \end{cases}$$

where $\theta$ is a parametric threshold value.

We note, in conclusion, that We have also considered the use of Jaccard similarity on the spectrogram peak locations as the matching score [34]. However, as we confirmed experimentally, due to the low sampling rate of the light sensor (10Hz), information and noise exist in the same frequency band. Thus, the noise component can easily distort the distribution of the frequency components. Another option is to compute the Pearson correlation coefficient on the whole spectrogram. However, in our experiments, this did not lead to significant improvement compared to temporal light intensity streams, and at the same time doing so was significantly less efficient.

## VII. Experimental Setup

We now describe our experimental setup.

**Physical Environment and Hardware:** In our emprical studies, we simulated a workstation environment consisting of a mobile device and a display. We experimented with different configurations by varying the FPD as specified in Table II; the mobile device performing light sensing as specified in Table I; as well as the displayed content and the distance between the display and the device.

The results we obtained across different devices and displays were similar, and so — for space reasons — we later report only the results for the iMac display and the Nexus 7 tablet. The light sensor on tablet, and the two other mobile devices, is at the single-pixel resolution; the output is a scalar real number corresponding to the sampled ambient light intensity measured in the SI unit of lux. The sensor output does not provide any information about the chromatic composition of the display content. The technology underlying the iMac display, as well as the two other displays, is LED-backlit LCD panels.

We set up the tablet, using its casing, to capture the light emanating from the display while in upright position. We performed the recordings under controlled laboratory conditions with standard ambient light settings, as in previous studies, with up to 5 lux [18], [32].

The collected light data was uploaded to a remote server: a MacBook Pro with 2.8GHz CPU and 8GB memory. The server ran the data-processing, classification and identification algorithms to analyze the light values.

| Phone Model | Light Sensor | Highest Sampling Rate | Resolution |
|---|---|---|---|
| Google Nexus 4 | Avago LGE | 2 Hz | $10^{-2}$ |
| Google Nexus 7 | LSC Lite-On | 10 Hz | $10^{-6}$ |
| Samsung S2 Plus | CM3633 | 10 Hz | 5 |

TABLE I
PHONES USED FOR RECORDING LIGHT DATA.

| Display Model | Dimension | Resolution/ Refresh Rate |
|---|---|---|
| iMac | 21.5" | $1920 \times 1080$, 60 Hz |
| MacBook Pro | 13" | $1280 \times 800$, 60 Hz |
| Acer | 23" | $1920 \times 1080$, 60 Hz |

TABLE II
DISPLAYS USED IN THE EXPERIMENTS

**Content:** To mount the inference attacks, we downloaded from the Internet more than 60 hours of free videos corresponding to browsing of web sites, game walkthroughs, coding lessons and tutorials and movie clips. We also recorded the screens for over 4 hours when four of our lab mates were browsing web sites, coding and playing games on their respective computers. Clips were selected such that each class of content has almost the same total duration. The clips are listed in Table III.

To mount a targeted content identification attack, we created a database of roughly 140 movie clips of length 5-10 minutes. We selected movies with high ratings and high diversity to ensure adequate representation for the following genres: action, adventure, animation comedy, drama and sci-fi.[1] (Ratings and genre were extracted from `imdb.com`.)

For the identification experiment, we randomly picked 40 clips to form a test set (to be matched) and 120 clips to form the database. The first 20 test clips have corresponding

[1] In line with previous studies [13] (involving only 26 movies), we consider the ability to distinguish betweeen 140 clips as evidence that an adversary can perform TV channel identification.

TABLE III
LIST OF VIDEO CLIPS.

| 2 Classes | 4 Classes | Content |
|---|---|---|
| Non-media | Webpage | Amazon, CNN, Facebook, Reddit, Yahoo, Basic HTML, CSS tutorial, Wikipedia and more |
| | Code | C++, Java, python, Matlab |
| Media | Game | Angry birds (Epic , Hero rescue), Clarence (Save the day), Lego (The hobbit), Plants vs. Zombies (Garden warface), Monkey go happy, Rayman Legends, South Park (Stick of truth), Little Ninja, Sponge Bob and more |
| | Video | Aladdin, Aliens, Batman Begins, Book Of Eli, Die Hard 2, Dr Horrible Sing along Blog, Harry Potter 3, Harry Potter 4, Indiana Jones, Kung Fu Panda, RHPS, Rocky 2, Terminator 2, The Expendables, The Matrix, The Matrix revolutions, Transformers 2, Transformers 3 and more |

templates in the database, while the remaining 20 do not. We then recorded the light intensity data corresponding for the test clips when they played on the display, and synthetically generated the light intensity data for the clips in the database. (See Figure 3.)

## VIII. EXPERIMENTAL RESULTS

We now describe the experiments we performed to evaluate the feasibility of inference and identification attacks.

### A. Inference Attacks

To evaluate the accuracy of inference, we performed two sets of experiments; the first, at a coarse grainularity of two classes: media vs non-media, and the second, at the finer granularity of four classes: webpage vs code vs game vs video (where video denotes either video or movie or TV). To avoid ambiguity, in our experimental setting, if a browser is used to play games or watch videos, then the expected class is game or video rather than webpage.

We experimented with four different classifiers:, $k$-nearest neighbor (KNN) with $k = 3$ and cosine similarity; naive Bayes (NB); decision tree (DT); and support vector machine (SVM) with radius basis function kernels. Classifications were performed using Matlab and the libsvm library [11]. We use 10-fold cross validation.

The training/testing partitioning is performed at the video-clip granularity, such that the testing clips have no overlap with the training clips. As a consequence, all the test frames have no overlap with the training frames. Note, importantly, that this setting is more realistic and more challenging than randomly sampling 10% of all the feature vectors for testing. The latter case results in significant overlap between testing and training frames, easily leading to good performance.

We use precision, recall and F-score as our performance metrics. Precision of matching is defined as the number of correct matches over the number of claimed matches by the matched filter. Recall is the number of correct matches over the number of chunks that indeed match one of the sequences in the database. These are computed separately for each class. A weighted average F-score is also reported where the weights are the proportions of testing feature vectors in respective classes.

*a)* **Experiment I: Media vs Non-media***:* Figure 5 plots the empirical cumulative distribution functions (ECDFs) of the extracted features (from Section V-A) for the media vs non-media classes with a window size of 90s. As the figure reveals, media-related and non-media-related contents are clearly distinguishable by the light intensity patterns per all features, and in particular range, mean absolute derivative, entropy, high-frequency energy ratio and spectral centroid.

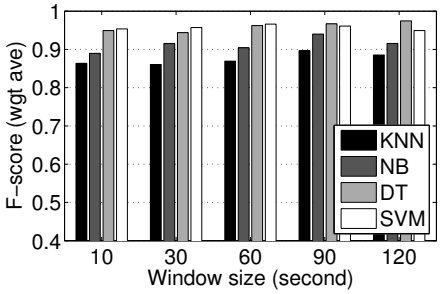



Fig. 6. Weighted average F-score with different classifiers for 2-class classification.

Figure 6 shows the weighted average F-scores for classification using all the features with different window sizes for framing and using different classifiers. We make two observations. First, all classifiers — and in particular DT and SVM — achieve high performance (with an F-score of close to, or above, 0.9), indicating that light intensity is an effective data source for inferring whether a target is viewing media. Second, given a window size of *only* 10s, all classifiers are well above the 0.8 F-score mark, where DT and SVM achieve an F-score of roughly 0.95. This confirms our hypothesis that *within a window of only 10 seconds, it is possible to perform highly accurate binary classification of the content shown on an FPD via the light sensor on a mobile device.*

For completeness, and given the high performance of DT, we further report per-class precision, recall and F-score for this classifier in Figure 7. We observe that the non-media class achieves slightly higher precision while the media class achieves slightly higher recall at 120s framing. The F-scores are fairly similar. All the precision, recall and F-score values are above 0.9.

*b)* **Experiment II: Webpage, Code, Game or Video***:* Figure 8 plots the ECDFs of the extracted features for four classes of content — webpage, code, game and video — with a window size of 90s. Notice that some features, such as range, standard deviation and skewness, can discriminate different types of content, while other features, such as high-frequency energy ratio and mean absolute derivative, may confuse some types of content such as webpage vs code. Moreover, the ECDF curves are more crowded for four classes than those for

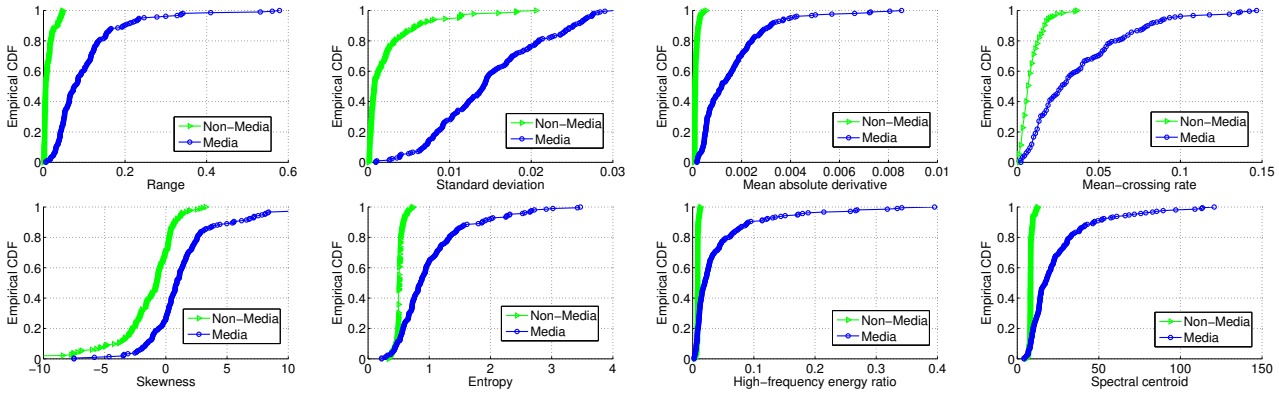
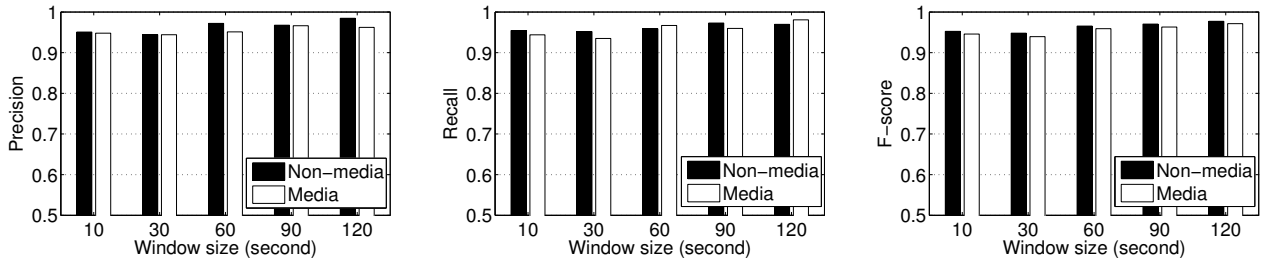



Fig. 5. Empirical cumulative distributions of features for two classes {Media, Non-media}.




(a) Precision (b) Recall (c) F-score

Fig. 7. Per-class precision, recall and F-score for 2-class classification using a decision tree.

two classes, which is expected since four-class classification is more difficult.

Figure 9 shows the weighted average F-score values for classification using all the features with different window sizes for framing and using different classifiers. Two important observations are the following: First, as expected, there is a performance decrease to the 0.6-0.7 range due to the increase from 2 to 4 classes. Second, the 10s window is comparable in performance to other window sizes, and in particular, it enables an F-score of about 0.65 (for SVM).

In general, DT and SVM remain the top performers among the four classification algorithms. For window sizes of 60s and beyond, both achieve a weighted average F-score of almost 0.7, which is significantly better than random guessing, demonstrating that light intensity readings are also effective in making finer-grained distinctions than binary judgments.

Figure 10 further examines the per-class precision, recall and F-score using DT. Consistent with the trend visualized in Figure 8, we observe that it is easiest to infer video, followed by code, game and webpage. This is because the target may spend time in a given scene before switching to a different scene (e.g., when playing Angry Birds), such that light intensity patterns become similar to those due to web browsing. In addition, certain 3D-game intensity patterns are similar to a movie. The best precision, recall and F-score values are in the $0.6-0.8$ range, indicating that these contents can be distinguished using light intensity.

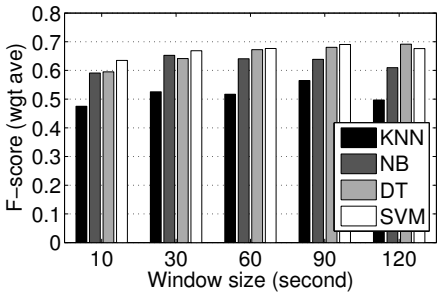



Fig. 9. Weighted average F-score with different classifiers for 4-class classification.

### B. *Identification Attack*

In the second set of experiments, where the database and testing sequences were prepared as explained in Section VII, we randomly picked 5 chunks out of each sequence, with a window size of $w$ seconds for matching, yielding $40\times5=200$ test chunks. We then applied matched filtering to each of the chunks against all the sequences in the database. The ideal matched filtering should result in correct matching for the first 20 sequences ($20\times5=100$ chunks) and no matching for the last 20 sequences (another 100 chunks). In the following, we examine the effect of the decision threshold, distance from the display and window size on identification quality.

*c)* **Effect of Decision Threshold***:* Figure 12(a) plots precision, recall and F-score values for matching as the decision threshold changes from 0.5 to 0.95 (with a window size of 120s and a distance of 40cm from the display). We observe that a high threshold increases precision while decreasing recall, whereas and a low threshold decreases precision while

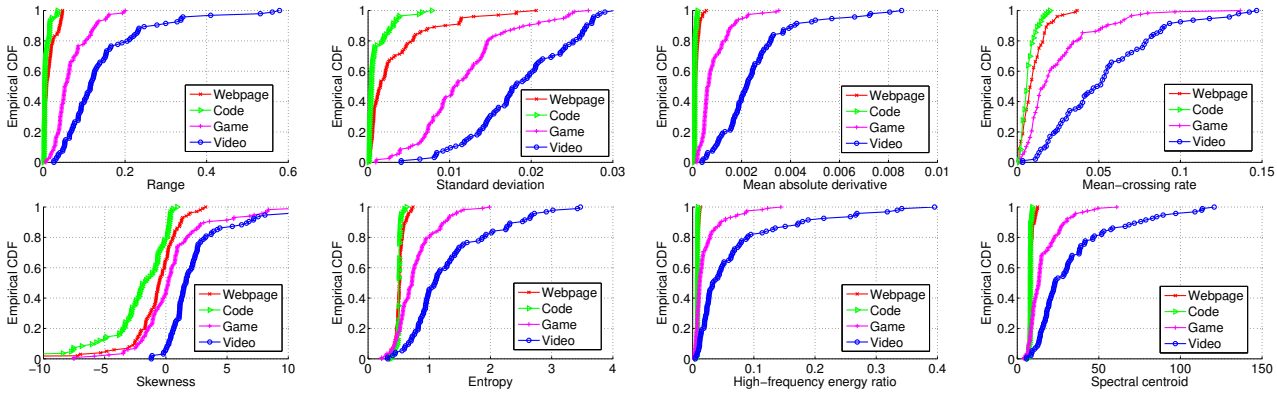
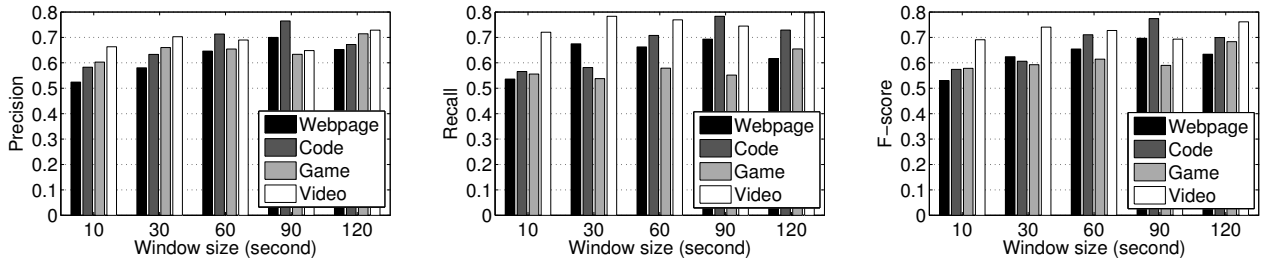



Fig. 8. Empirical cumulative distributions of features for four classes {Webpage, Code, Game, Video}.




(a) Precision (b) Recall (c) F-score

Fig. 10. Per-class precision, recall and F-score for 4-class classification using a decision tree.

increasing recall. The optimal F-score (0.85) is achieved when the decision threshold is 0.65. The F-score value is relatively stable when the decision threshold changes between 0.6 and 0.8. The observed F-score value of 0.85 confirms that light intensity is sufficient to identify the content a target is viewing against a database with reasonable accuracy.

*d)* **Effect of Distance***:* Figure 11 shows the mean and standard-deviation values of the matching scores for movies collected with different distances from the display. On average, assuming a window size of 120s, each movie chunk has a relatively high matching score (i.e., in the $0.8 - 0.9$ range) against the corresponding template (denoted "same") in the database, and a much lower matching score (below 0.6) against other templates (denoted "different"). We conclude that different movies generally result in different patterns of the light intensity, which can be used to differentiate them.

Another important observation is that the matching score for the same movie decreases only marginally as the distance from the display increases: from an average of 0.9 to 0.8 as distance increases from 20cm to 100cm. This is because of the attenuation in light intensity and the increase in the ambient noise. Figure 12(b) plots the precision, recall and F-score values matching when the distance to the display changes from 20cm to 100cm (with a window size of 120s and a decision threshold of 0.65). Recall decreases rapidly, while precision decreases more smoothly. However, even if the distance is 80cm, the F-score value remains high (around 0.80).

*e)* **Effect of Window Size***:* Figure 12(c) plots precision, recall and F-score of matching when the window size of the data chunk varies from 10s to 120s (with a decision threshold

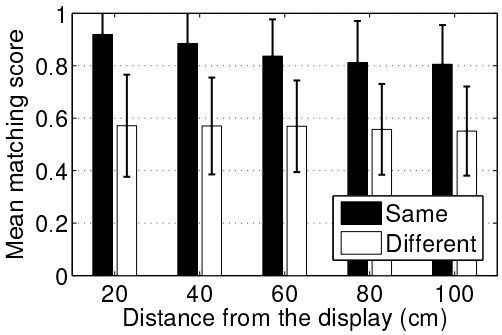



Fig. 11. Means and standard deviations of matching scores for movies collected at different locations using the light sensor on a tablet.

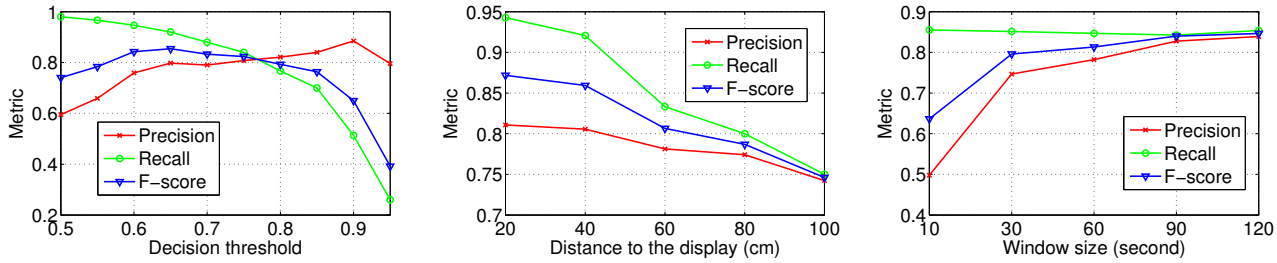
of 0.65 and a distance of 40cm from the display). As expected, a larger window size achieves better performance. We also observe that the recall of matching is not sensitive to the window size while precision is.

## IX. Conclusion

In this paper, we have demonstrated the feasibility of using the light sensor of a mobile device to recover information about the content shown on a nearby FPD. While such single-pixel light sensors have limited power, a judicious choice of features that capture information pertaining to changes in light intensity over time allows us to infer sensitive information about the content type.

## Acknowledgement

This research was sponsored by the U.S. Army Research Laboratory (ARL) and the U.K. Ministry of Defence under Agreement




(a) Effect of decision threshold (b) Effect of distance to the display (c) Effect of window size

Fig. 12. Effect of the decision threshold, the distance to the display and the window size for matching on the identification performance.

Number W911NF-16-3-0001. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the ARL, the U.S. Government, the U.K. Ministry of Defence or the U.K. Government. The U.S. and U.K. Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

## REFERENCES

[1] Accessing Light Sensor. http://developer.android.com/guide/topics/sensors/sensors_environment.html.

[2] Android Security Overview. http://source.android.com/devices/tech/security/.

[3] Exposure Value. http://dougkerr.net/Pumpkin/articles/APEX.pdf.

[4] Lux Auto Brightness. https://play.google.com/store/apps/details?id=com.vito.lux&hl=en.

[5] Pixel Format. http://developer.android.com/reference/android/graphics/PixelFormat.html.

[6] Targeted TV Ads Set For Takeoff. http://tinyurl.com/mnladav.

[7] Targeting Ads By Tracking Web Surfing. http://tinyurl.com/nfpf7pk.

[8] D. Asonov and R. Agrawal. Keyboard acoustic emanations. In *Security and Privacy, 2004. Proceedings. 2004 IEEE Symposium on*, pages 3–11, 2004.

[9] C. M. Bishop et al. *Pattern recognition and machine learning*, volume 1. springer New York, 2006.

[10] L. Cai and H. Chen. Touchlogger: Inferring keystrokes on touch screen from smartphone motion. In *Proceedings of the 6th USENIX Conference on Hot Topics in Security*, HotSec'11, 2011.

[11] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011.

[12] G. Disterer and C. Kleiner. Using mobile devices with byod. *Int. J. Web Portals*, 5(4):33–45, 2013.

[13] 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*, CCS '11, pages 537–550, 2011.

[14] D. Genkin, A. Shamir, and E. Tromer. Rsa key extraction via low-bandwidth acoustic cryptanalysis. Cryptology ePrint Archive, Report 2013/857, 2013.

[15] U. Greveler, B. Justus, and D. Loehr. Multimedia content identification through smart meter power usage profiles. In *Computers, Privacy and Data Protection*, 2012.

[16] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The weka data mining software: An update. *SIGKDD Explor. Newsl.*, 11(1):10–18, 2009.

[17] J. Han, E. Owusu, L. Nguyen, A. Perrig, and J. Zhang. Accomplice: Location inference using accelerometers on smartphones. In *Communication Systems and Networks (COMSNETS), 2012 Fourth International Conference on*, pages 1–9, 2012.

[18] M. Kuhn. Optical time-domain eavesdropping risks of crt displays. In *Security and Privacy, 2002. Proceedings. 2002 IEEE Symposium on*, pages 3–18, 2002.

[19] M. Kuhn. Compromising emanations of lcd tv sets. In *Electromagnetic Compatibility (EMC), 2011 IEEE International Symposium on*, pages 931–936, Aug 2011.

[20] M. G. Kuhn. Electromagnetic eavesdropping risks of flat-panel displays. In *Proceedings of the 4th International Conference on Privacy Enhancing Technologies*, PET'04, pages 88–107, 2005.

[21] D. Li, I. K. Sethi, N. Dimitrova, and T. McGee. Classification of general audio data for content-based retrieval. *Pattern recognition letters*, 22(5):533–544, 2001.

[22] J. Loughry and D. A. Umphress. Information leakage from optical emanations. *ACM Trans. Inf. Syst. Secur.*, 5(3):262–289, Aug. 2002.

[23] 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*, CCS '11, pages 551–562, 2011.

[24] S. Mazilu, U. Blanke, A. Calatroni, and G. Trster. Low-power ambient sensing in smartphones for continuous semantic localization. In *4th International Joint Conference, Ambient Intelligence*, pages 166–181, 2013.

[25] Y. Michalevsky, D. Boneh, and G. Nakibly. Gyrophone: Recognizing speech from gyroscope signals. In *23rd USENIX Security Symposium*, pages 1053–1067, Aug. 2014.

[26] Y. Michalevsky, A. Schulman, G. A. Veerapandian, D. Boneh, and G. Nakibly. Powerspy: Location tracking using mobile device power analysis. In *24th USENIX Security Symposium,*, pages 785–800, 2015.

[27] E. Miluzzo, A. Varshavsky, S. Balakrishnan, and R. R. Choudhury. Tapprints: Your finger taps have fingerprints. In *Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services*, MobiSys '12, pages 323–336, 2012.

[28] A. Narayanan and V. Shmatikov. Robust de-anonymization of large sparse datasets. In *IEEE Symposium on Security and Privacy*, 2008.

[29] R. Spreitzer. Pin skimming: Exploiting the ambient-light sensor in mobile devices. In *Proceedings of the 4th ACM Workshop on Security and Privacy in Smartphones & Mobile Devices*, SPSM '14, pages 51–62, 2014.

[30] H. Tanaka. Information leakage via electromagnetic emanations and evaluation of tempest countermeasures. In *Proceedings of the 3rd International Conference on Information Systems Security*, ICISS'07, pages 167–179, 2007.

[31] G. L. Turin. An introduction to matched filters. *IRE Transactions on Information Theory*, 6(3):311–329, 1960.

[32] W. van Eck. Electromagnetic radiation from video display units: An eavesdropping risk? *Comput. Secur.*, 4(4):269–286, Dec. 1985.

[33] M. Vuagnoux and S. Pasini. Compromising electromagnetic emanations of wired and wireless keyboards. In *Proceedings of the 18th Conference on USENIX Security Symposium*, SSYM'09, pages 1–16, 2009.

[34] A. Wang et al. An industrial strength audio search algorithm. In *ISMIR*, pages 7–13, 2003.

[35] B. J. Wilson. Media and Children's Aggression, Fear, and Altruism. *The Future of Children*, 18(1):87–118, 2008.

[36] 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*, CCS '13, pages 1063–1074, 2013.

[37] Z. Zhang, P. Liu, J. Xiang, J. Jing, and L. Lei. How your phone camera can be used to stealthily spy on you: Transplantation attacks against android camera service. In *Proceedings of the 5th ACM Conference on Data and Application Security and Privacy*, CODASPY '15, pages 99–110, 2015.