Watching the Watchers: Automatically Inferring TV Content From Outdoor Light Effusions

Yi Xu, Jan-Michael Frahm and Fabian Monrose
Department of Computer Science, University of North Carolina at Chapel Hill
Chapel Hill, North Carolina, USA
{yix,jmf,fabian}@cs.unc.edu

ABSTRACT

The flickering lights of content playing on TV screens in our living rooms are an all too familiar sight at night — and one that many of us have paid little attention to with regards to the amount of information these diffusions may leak to an inquisitive outsider. In this paper, we introduce an attack that exploits the emanations of changes in light (e.g., as seen through the windows and recorded over 70 meters away) to reveal the programs we watch. Our empirical results show that the attack is surprisingly robust to a variety of noise signals that occur in real-world situations, and moreover, can successfully identify the content being watched among a reference library of tens of thousands of videos within several seconds. The robustness and efficiency of the attack can be attributed to the use of novel feature sets and an elegant online algorithm for performing index-based matches.

Categories and Subject Descriptors: K.4.1 [Computers and Society]: Privacy

General Terms: Human Factors, Security

Keywords: Visual eavesdropping; Compromising emanation

1. INTRODUCTION

To most of us, it would come as no surprise that much of our population is addicted to watching television, due in part to the wide variety of entertainment (e.g., reality TV, game shows, movies, premium channels, political commentary, 24hr news, etc.) that is offered in today's competitive market place — be that online or via broadcast TV. Indeed, so-called catch-up TV and Internet connectivity now liberate viewers from restrictive schedules, making watching shows part of a wider and richer experience in homes. Admittedly, although familiar TV sets of the old days are not as popular as they once were, TV is here to stay and its role in delivering compelling viewing experiences will continue for decades.

The markedly richer content offered today has helped sustain living room screens as a dominant communication medium — both collectively (e.g., for watching a big game or season finale) and individually (e.g., for accessing specific content on demand). In fact, even though consumer viewing habits have undergone change in recent years (e.g., phone, tablet and computer viewing habits have steadily increased), nearly every U.S. home still owns at least one TV and 67% of Americans regularly watch television while having dinner [6]. The flickering lights of the scenes that play out on these TVs are easy to see when one walks through the street at nights. Yet, many of us may not have given a second thought to the amount of information these flickering patterns (caused by changes in brightness) might reveal about the programs we watch.

Our findings, however, suggest that these compromising emissions of changes of brightness provide ample opportunity to confirm what specific content is being watched on a remote TV screen, even from great distances outside the home. The key intuition behind why this threat to privacy is possible lies in the fact that much of the content we watch induces flickering patterns that uniquely identify a particular broadcast once a suitable amount of light emissions (i.e., on the order of a few minutes) has been recorded by the adversary. This surprisingly effective attack has significant privacy implications given that the video and TV programs that people watch can reveal a lot of information about them, including their religious beliefs, political view points or other private interests. For that reason, subscribers' viewing habits are considered sensitive under the U.S. Video Protection Privacy Act of 1998, which states that an individual's video viewing records must be kept confidential. Recently, a popular electronics firm came under investigation when it was revealed that its Smart TV was surreptitiously sending information on viewing habits back to the parent company in an effort to "deliver more relevant advertisements" [1].

While the observations we leverage in this paper have been part of folklore, to the best of our knowledge, we present the first automated, end-to-end, approach for realizing the attack. Undoubtedly, the academic community has long acknowledged that video viewing records are vulnerable to different attacks (e.g., due to electromagnetic or power line behavior [4, 7, 9]), but these attacks have not received widespread attention because they require access to smart power meters and other specialized equipment in order to capture the required signal. Moreover, because these attacks rely on specific TV/computer screen electronic properties they remain difficult to pull off in practice.

In this paper, we push the boundaries of these attacks by exploiting compromising emissions which are far easier to capture in practice. In fact, we do not rely on the adversary's ability to capture an image of the screen, or its reflection on a nearby surface (e.g., [1, 17]). Instead, our attack works by analyzing the changes in brightness in a room where the content is being watched, and matching the captured signal in real-time with reference signals

CCS'14, November 3–7, 2014, Scottsdale, Arizona, USA.

ACM 978-1-4503-2957-6/14/11 ...$15.00.
http://dx.doi.org/10.1145/2660267.2660358.

[1] See J. Brookman, *Eroding Trust: How New Smart TV Lacks Privacy by Design and Transparency* at `http://www.privacyassociation.org/`, Nov. 2013.

stored in a large database. The attack can be successfully carried out with inexpensive consumer devices (e.g., web cameras, digital SLRs) and works as long as illumination changes caused by the TV screen are perceptible to the camera's sensor.

To ensure that the attack is resilient to noise (e.g., from a passing vehicle, the turning on/off of a light switch, or from human movement), our approach focuses squarely on significant changes in the captured sequence, instead of directly leveraging all of the captured signal. Said another way, we exploit temporal brightness information that is not adversely affected by device-specific or environmental conditions. These environmental conditions (e.g., reflections off a wall) might result in a weakened and distorted overal signal, but the temporal information of significant intensity changes will remain largely intact.

A key contribution in this paper lies in the techniques we use to take advantage of temporal information to find matches among reference and captures signals, even in the face of significant noise and signal distortions. To do so, we extend traditional correlation measures to utilize temporal information when computing similarity scores between sequences. The resulting strategy significantly outperforms traditional correlation measures (e.g., [7]), for which we present an on-line approximation method. Our empirical analysis covering 54,000 videos shows that we can perform this confirmation attack with surprising speed and accuracy.

2. RELATED WORK

Techniques for undermining user's privacy via TV program retrieval has long been studied. The most germane of these works is that of Enev et al. [4] and Greveler et al. [7] wherein power usage and power line electromagnetic interference were investigated as side-channels. Unlike the approach we take, these works encode the TV signal in ways that largely depend on the model of the TV and the structure of the power system. Therefore, to successfully carry out the attack, an adversary must not only have specialized equipment and access to smart meters, but must also have a priori knowledge of the victim's TV model — all of which weaken the practicality of the attack. Moreover, other electronic devices (e.g., computers) within the vicinity of the TV can interfere with the captured signal, compounding the decoding challenges even further.

Other side-channels include the use of so-called compromising reflections, which was first introduced by Backes et al. [1]. Shiny object reflections (e.g., from a nearby teapot or off an eyeball) were used to recover static information displayed on the target screen. More recently, compromising reflections were also exploited by Raguram et al. [13] and Xu et al. [17] to reconstruct information being typed on virtual keyboards. In a similar manner, Torralba and Freeman [14] make use of reflections to reveal "accidental" scenes from within a still image or video sequence. The advantage for these approaches comes from the uniformity and easy-access of visual signals; while TV screen and computer screens come with different model using different technologies — resulting in extremely different electromagnetic behavior — they all share similar visual output. Due to market demand, the emanation of the visual signal has to cover a certain area and maintain a certain brightness level to ensure clarity of picture, which also makes them susceptible to compromising reflections. That said, these attacks require a view of the screen, either directly or via reflections.

Also related within the domain of computer vision is the process of image and video retrieval. Interested readers are referred to Zhang and Rui [18], which presents an excellent review of image retrieval techniques used to search through billions of images. Likewise, Liu et al. [11] presents a survey of near-duplicated video retrieval techniques that also focus on similarity of semantic content of the video sequences. In short, features are extracted to reveal detail information in the image and semantic labels are used to provide a high level understanding. Unfortunately, we have no such luxury in our application since we may have no visual access either directly or indirectly to the screen, and must therefore find ways to work with much more limited information.

Lastly, our application domain shares similarities to genome sequence matching and database searching. In particular, considering only the average image intensity signal, the task at hand can be viewed as a sequence matching problem. For instance, in genome sequence matching, Langmead et al. [10] present a fast DNA sequence matching scheme that exploits time and space trade-offs. In database searching, Faloutsos et al. [5] and Moon et al. [12] present methods that perform fast matching from an input subsequence to those in a database. Unfortunately, these techniques suffer from several limitations that make them ill-suited for our setting. For example, in DNA sequence matching, many parts of a sequence may be missing and so to find the best matches, dynamic optimization methods are usually deployed to maximize the length of the best match. These algorithms typically have $O(mn)$ complexity, where m is the length of the query sequence and n is the length of the reference sequence. In our application, however, the only uncertainty is the starting point of the query sequence and so much more effective strategies (i.e., $O(nlog(m))$ or faster) can be applied.

In database searching, the problem is more similar to ours, but the state-of-the-art solutions utilize Fourier transformation and focus on low frequencies. In our application, the sudden intensity changes contain most of the information we utilize, but live in the high end of the frequency spectrum. As such, these approaches can not be directly applied. However, by combining many of the strengths of prior works together with our own enhancements, we provide a solution that boasts high accuracy and speed.

3. OVERVIEW

The key insight we leverage is that the observable emanations of a display (e.g., a TV or monitor) during presentation of the viewing content induces a distinctive flicker pattern that can be exploited by an adversary. This pattern is observable in a wide range of scenarios, including from videos capturing the window of the room housing the display, videos from cameras pointed at a wall in the room but not at the TV directly, videos observing the watcher's face (for example, via a Kinect or similar front-mounted camera), and of course, from video capturing the TV directly. To facilitate our attack, we convert the observed pattern of intensity changes into a suitable feature vector that is amenable to rapid matching of other stored feature vectors within the adversary's large corpus of processed videos.

In this paper, we compute the average pixel brightness of each frame in the video, resulting in a mean brightness signal for the video. To capture the sharp changes in brightness, we then use the gradient of the signal as the descriptor for the video. The overall process is illustrated in Figure 1. Similar to the captured video, every video in the adversary's collection is represented by a feature based on the gradient of the brightness signal. Note that while the mean brightness signal of the reference video and the captured videos signal may vary, their gradient-based features share more characteristics in common, and it is those commonalities that are used to identify the content being watched.

4. BACKGROUND

The ability to confirm which video is being watched based off compromising diffusions of changes in light hinges on several fac-

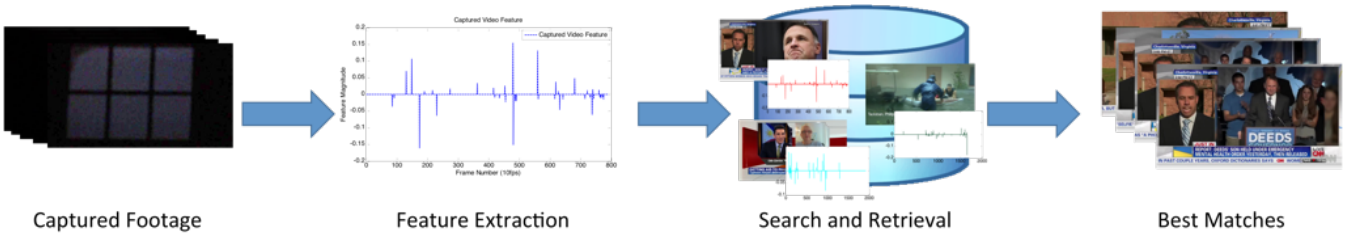

Figure 1: The high-level workflow of our approach. Features are extracted from the captured video and then compared with features from reference videos in a database. The reference video with the most similar feature is output as the most probable candidate.

tors including (i) the quality of the captured information (i.e., the signal-to-noise ratio), (ii) the entropy of the observed information (i.e., the amount of variation in the captured signal) (iii) the length of the captured signal (e.g., short clips have more ambiguity), and (iv) the amount of information required for successfully matching the unknown and reference signals, which is related to the size of the adversary's reference library and the distinctiveness of its contents. We discuss each in turn.

Noise Interference

For an arbitrary recording, our goal is to infer a signal, S, based on effusions of light from the display. In practice, this means that we also inadvertently capture an additive noise signal, N, which may be composed of a variety of other signals (e.g., sensor noise, photon noise). Consequently, the recording we capture is the composition of the signal S and noise N. Intuitively, the more significant the noise, the harder it will be to distinguish between the noise and the signal. This correlation is measured by the signal-to-noise ratio (SNR), which is the ratio of the signal variance σ_S^2 and the noise variance σ_N^2.

In general, the higher the SNR the less the noise influences the resulting signal, which leads to more robust signal analysis. In the case of capturing reflections of emanations, the SNR depends on a multitude of factors. More specifically, the amount of light emanated from the screen at any frame depends on the intensity of the video frame that is displayed on the screen, I_{ref}, the current brightness level of the screen (measured by unit area emanation power P_0), and the size of the screen S_{screen}. However, only a small fraction of this light might be captured by the camera, the amount of which depends on the distance the light travels from the screen to the reflecting object, the size and reflectance of the reflecting object, the aperture of the camera and the distance from the reflecting object to the camera. The captured signal also depends on the sensitivity α_{cam} of the imaging sensor of the recording device. In summary, assuming α_{cap} is the percentage of emanation captured by the camera, the recorded signal can be modeled as:

$$I_{cap} = I_{ref} P_0 S_{screen} \alpha_{cap} \alpha_{cam} \tag{1}$$

It is important to note that α_{cap} and α_{cam} are not constant in practice because of the different reflectance properties for colors and the non-linear color transformation of digital cameras [15]. Hence, they will depend on the actual color composition of the displayed video frame. Additionally, the intensity of light in the room influences the amount of incoming light and could be treated as another signal, but for simplicity, we consider it an additive constant as long as the lights are not being repeatedly turned on and off. As such, we omit its embedded signal in Equation 1, but instead simply consider it as a source of "impulse noise" [3], similar to the lights of a passing vehicle.

To complicate matters even further, there can be noise from a myriad of other sources that impact the measured brightness value in the adversary's recording of the emanations coming from the display. These include quantization noise of the camera during the A/D conversion to obtain pixel values [16], thermal noise from the sensor itself [8] and impulse noise. Again, for simplicity, we accumulate the above noise factors into a single noise variable I_{noise}. The SNR can then be computed as:

$$SNR = \frac{\sigma(I_{cap})^2}{\sigma(I_{noise})^2} \tag{2}$$

where $\sigma^2(\cdot)$ is the variance of the signal.

Intuitively, lower screen brightness levels, smaller and darker reflecting objects, and longer distances limit the amount screen light captured by the adversary. Fortunately for the adversary, a high quality camera can capture a good percentage of the incoming light and reduce quantization and electronic noise. Finally, note that while the intensity of a constant room light does not influence the SNR directly—since it does not influence the noise variance—it indirectly effects the quantization noise given that it affects the sensitivity of camera's sensor (i.e., higher room light intensity makes the camera less able to capture subtle illumination changes).

For our experimental evaluations we can directly acquire $I_{cap} + I_{noise}$ from the captured video. An estimate of the noise variance $\sigma(I_{noise})^2$ can be measured by having the adversary capture the reflection from a static image displayed on screen beforehand (e.g., at her house). Similarly, room brightness can be approximated. With these measurements at hand, I_{cap} can be estimated with linear regression using I_{ref}, and the SNR can be directly computed.

Takeaway. The factors that influence the signal we are interested in can be approximated by Equation 1. Moreover, by using Equation 2, we can infer the SNR directly from the captured data, which ranged from 5 to 107 in our empirical evaluations.

Point of Capture

Obviously, the point at which the recording of the light diffusions is taken can influence how well the attacker can confirm her hypotheses. Intuitively, the more she is able to record sudden intensity changes, the higher the chances are that the correct content will be inferred. A key challenge for the adversary is that the average intensity of one frame is highly dependent on that of the previous frame. For instance, in our empirical evaluations, nearly 95% of consecutive frames have the same average intensity (up to rounding error precision).

To improve our ability to carry out the attack, we do not use the raw data directly, but instead, use its gradient to reduce the correlation. To see why that helps, assume that $x_t = I_{ref}(t+1) - I_{ref}(t), y_t = I_{cap}(t+1) - I_{cap}(t), t = 1,2,3,....$ Then, given that the vast majority (i.e., 95%) of the average intensities are similar, this

means that 95% of the xs would be 0. Assuming the gradients are independent of one another, the information with a particular frame sequence $\{x_t, t = 0, 1, 2, 3, ...\}$ can be measured as:

$$Info_{ref}(x) = -\Sigma_{t=0}^{N} log(f(x_t)\Delta x) \quad (3)$$

where $f(x_t)$ is the probability density function (PDF) of x_t in a single frame. $Info_{ref}(x)$ can be viewed as the logarithm of the inverse probability of the reference sequence. The higher its value, the less likely another reference sequence will "accidentally" be the same as it, which means that the sequence has less ambiguity and contains more information. Consequently, the more intensity changes the adversary observes, the more likely it is that the correct content will be inferred.

To gauge how well the attack should work, we can compute the mutual information between x and y using Equation 4. $Info_{mutual}(x,y)$ estimates the information captured by the adversary on average.

$$Info_{mutual}(x,y) = \int p(x)p(y|x)log(\frac{p(y|x)}{p(y)}) \quad (4)$$

In practice, $p(x)$ can be observed directly from a reference video. Likewise, $p(y)$ can be computed by ignoring impact noises (which are rare) and assuming that the noise follows a Gaussian distribution. In fact, since I_{ref} and I_{cap} are linearly related, we can also assume $y = x + noise$, where $Var\{noise\} = Var\{x\}/SNR$. In doing so, we can now compute the mutual information with the SNR we acquired. For context, we note that in our evaluations that follow at an SNR of 5 every frame conveyed roughly 1.5 bits of information. Under much better conditions with SNR of 107 (observed when the diffusions were captured while the victim watched an action scene on a 50-inch TV) every frame conveyed 3.2 bits of information.

Takeaway. The above analysis tells us what one would expect: the more intensity changes observed, the less the resulting ambiguity. Therefore, if the adversary is lucky enough to observe several sharp changes in intensity, she will have an easier time to identify the content being watched by the victim. Not surprisingly, Equation 4 also tells us that bigger and brighter screens provide more than twice as much information (compared to the smaller and darker ones used in our experiments).

Length of Recording

Given the previous discussions, longer recordings are obviously better for the adversary. To see that, assume that the arrival of intensity changes are Markov, meaning that the distribution of arrival time and magnitude of the next intensity change only depends on the current state of the video being watched. If that is the case, then the information learned by the adversary is linearly related to the mutual information per frame. Ideally, the attacker's best hope is for a high SNR environment, a good starting point, and a suitable recording length capturing multiple changes in intensity.

Size of the Reference Library

The last factor that affects the speed and accuracy of the attack is the size of the reference collection the adversary must test her hypotheses against. In the worst case, the amount of information we need to uniquely identify a video is logarithmic with its total length, which in turn, is linearly related to the size of the attacker's library. Therefore, linearly increasing the size of the library will only have marginal influence on her ability to successfully confirm which content the victim is watching.

5. AUTOMATED VIDEO RETRIEVAL

Our approach (as shown in Figure 1) consists of two main parts that comprise a *feature extraction* step from the captured recording and a *video retrieval* step using a precomputed library of features from reference content (i.e., the set of videos for which the adversary wishes to confirm her hypotheses against). The feature extraction stage converts the captured video into a representative encoding that encodes the changes in brightness. This feature is then compared during the video retrieval to the features in the database to identify the content on the victim's display.

5.1 Feature Extraction

Intuitively, in our feature representation we want to preserve the brightness changes of the displayed image. Hence, for each frame t of the video, with M frames, we calculate the average intensity $s(t)$ with $t = 0, \ldots, M$, by averaging all the brightness values of all pixels in the image. The sequence s of these average brightness values $s(t)$ with $t = 0, \ldots, M$, provides us with a coarse characterization of the captured video's brightness. An example brightness sequence for a video s_c captured through the window and the corresponding original video s_d is provided in Figure 2. Notice that while the variation of the two signals is comparable, the magnitude of the brightness signals $s(t)$ is significantly different.

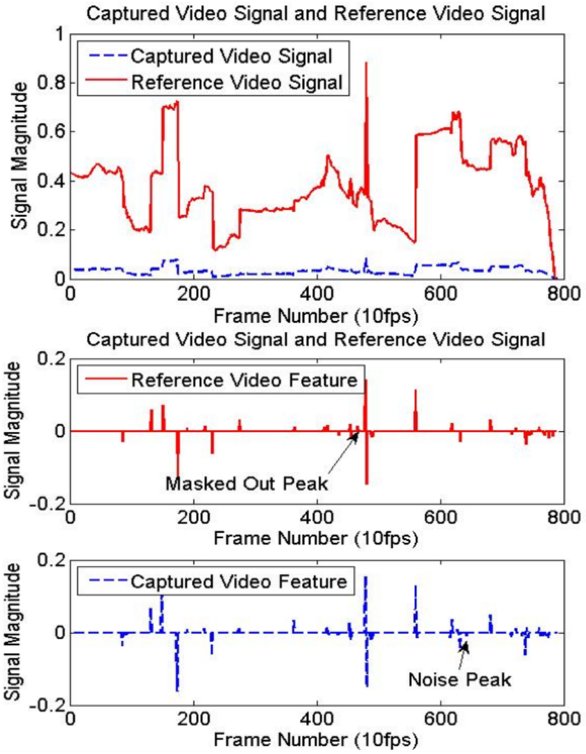

Figure 2: The intensity signal (top) and respective features (middle and bottom). For illustrative purposes, the sequences are manually aligned. Noises occur as peaks or masked out peaks in the feature sequence

To achieve comparability of the two signals s_c and s_d we characterize them by their frame-wise intensity gradient over time $ds(t)$. Given the average intensity signals s_c and s_r respectively, the temporal gradient $ds(t)$ can be calculated as $ds(t) = s(t+1) - s(t)$.

Based on our conjecture that the brightness changes uniquely characterize a video, we convert the temporal gradient ds into a feature f by only preserving its significant ($|ds(t)| > 1$) local maxima and minima

$$f(t) = \begin{cases} ds(t), & if\ |ds(t)| > 1 \wedge |ds(t)| > |ds(t-1)| \\ & \wedge |ds(t)| > |ds(t+1)| \\ 0, else & \end{cases} \quad (5)$$

If the video sequence's brightness does not have scene changes, flashes or other sudden changes, the average intensity is nearly constant, leading to zero values in $f(t)$. By contrast, if there is a sudden intensity change (e.g., a drastic scene change or flashes of gun shots) $f(t)$ will capture a "peak" which is either positive or negative, representing a sudden increase or decrease in average intensity. Accordingly, $f(t)$ can be viewed as a composition of peaks. For a captured video, some of the peaks might correspond to noise or noise might mask some peaks in f. Additionally, the magnitude of the peaks might be still scaled by an unknown factor. Example features f_c and f_d for a captured video and the retrieved database video are shown in Figure 2 (middle, bottom).

5.2 Creating the Reference Library

Our video retrieval requires a database of reference videos to retrieve the corresponding video being watched. This database is typically obtained ahead of time by obtaining all content of interest. If only the content for live TV is of interest to the adversary, she can just record all the currently running TV channels. If the adversary is interested in online videos, a database of popular videos (e.g., from YouTube, Netflix, or her home collection) would be helpful. Once all videos of interest are obtained, they are converted to feature vectors using the same feature extraction technique used for the captured sequences (see Section 5.1).

5.3 Locating the Best Matching Sequences

To identify the best match we use a nearest neighbor search across subsequences because the captured sequence typically only covers a small part of the overall content being watched on the display. For ease of exposition, we first introduce our similarity metric for the case that both the captured length $length(f_c)$ and the reference video length $length(f)$ are the same and start at the same time. Later, we generalize the metric to account for different lengths and starting points of the captured and the reference videos.

Intuitively, to measure the similarity of the feature vectors for the captured video f_c and a reference video $f_i \in \{f_{ref}\}$, we can examine how many extrema match between the features. The amount of disturbance caused by erroneous noise peaks is represented by

$$E_{noise}(f_i, f_c) = \frac{\Sigma_{t=1}^{L} f_c(t)^2 1(f_i(t) == 0)}{\Sigma_{t=1}^{L} f_c(t)^2} \quad (6)$$

where L is the length of the videos and $1(x)$ is the indicator function, which is one if x is true and zero otherwise. Similarly

$$E_{miss}(f_i, f_c) = \frac{\Sigma_{t=1}^{L} f_i(t)^2 1(f_c(t) == 0)}{\Sigma_{t=1}^{L} f_i(t)^2} \quad (7)$$

measures the energy of missing peaks in the reference sequence. Note that while E_{noise} and E_{miss} characterize the magnitude of difference in the number of peaks, we must also measure the amount of difference in energy of the peaks by characterizing how similar the extrema themselves are. This can be measured as the correlation $Corr(f_i, f_c)$

$$Corr(f_i, f_c) = \frac{\Sigma_{t=1}^{L} f_c(t) f_i(t)}{\sqrt{(\Sigma_{t=1}^{L} f_c(t)^2)(\Sigma_{t=1}^{L} f_i(t)^2)}} \quad (8)$$

between the two sequences, which has a value between -1 and 1. In this paper, we use a similarity metric d that combines E_{noise}, E_{miss} and $Corr(f_i, f_c)$ into a single metric:

$$\begin{aligned} d(f_c, f_i) = & \alpha (E_{noise}(f_i, f_c) + E_{miss}(f_i, f_c)) \\ & + (1 - Corr(f_i, f_c)) \end{aligned} \quad (9)$$

with α representing the weighting between the energy of the missing or noise peaks and the correlation between the correct extrema; the latter is necessary when distinguishing features in the case of perfectly agreeing peaks. Given that the magnitudes of the peaks may be different between the captured and reference signals, we rely on the temporal information which is more accurate. As such, we empirically chose $\alpha = 50$ for all our experiments so that the temporal agreement of peaks dominates the metric. It is only when the temporal position of peaks matches perfectly that $Corr(f_i, f_c)$ is used to evaluate their similarity based on magnitude.

Returning to §2, it is important to remind the reader that our metric is based on the gradient of average intensity. Therefore, it captures sharp intensity changes and ignores smooth terms such as ambient light condition, the auto exposure of camera and other gradual changes. Even impulse noise (e.g. turning on/off the room light) only result in a single extra peak in the feature vector and thus has minor impact on the overall result. Other alternatives such as using the correlation directly (e.g., as proposed by Greveler et al. [7]) fail in our scenario since these approaches are significantly impacted by signal magnitudes which are often heavily distorted. Likewise, the FFT transformation used in sequence matching schemes [5, 12] also fails because the peaks are too sparse for frequency analysis and the localized changes are too subtle to be useful.

In our evaluations that follow, the reference video that best matches under our similarity metric d is reported as the likely content being watched. We note that in practice the temporal position of the extrema may vary by one frame due to encoding and sampling of the original video sequence. Therefore, when determining whether $f_i(t)$ or $f_c(t)$ is non-zero, we consider the adjacent two frames $(t-1, t+1)$ in addition to the frame at time t by using the modified indicator function $\tilde{1}(x)$ in Equations (6)-(9), which is one if none of x or its temporal neighbors is true.

Notice that thus far, the retrieval using the similarity metric d from Equation (9) assumes equal length and starting point of the videos. To relax this assumption, for a recording of length $l_c = length(f_c)$ we search all subsequences of length l_c among all database sequences of length greater than or equal to l_c. This has the added benefit that we not only identify what content was watched, but also what part of the video was watched at the time the recording was taken. The problem, however, is this type of exhaustive search comes with a significant computational burden. In what follows, we discuss how to achieve a more efficient solution in practice.

6. ILLUMINATI: EFFICIENT ATTACKS USING COMPROMISING EFFUSIONS

To tackle large-scale databases of tens of thousands of videos, we employ a matching algorithm that only needs to search a small fraction of the database. Recall that our similarity metric (Equation 9) mainly matches significant intensity changes (peaks in our feature representation) in the captured video with the peaks in the database videos. Next, we leverage the fact that these peaks are only present in a small fraction of the video frames and propose a new *peak-feature* that efficiently characterizes the distribution of peaks. This distribution can then be used to narrow down the search space and speed up the search by an order of magnitude. Our proposed algorithm consists of two steps. The first step is the extraction of

the features based on only the peaks and the second step uses an efficient index-based search.

6.1 Peak-feature Extraction

Our proposed peak-feature aims at capturing the distribution of the peaks caused by sudden intensity changes in the video. As shown in Figure 3, the peak-feature is computed within a sliding window, of size $w = 512$, over the gradient feature, i.e. the peak-feature is computed from the w consecutive feature values. The value 512 is chosen empirically since our experiments indicate that subsequences shorter than 512 frames (at 10 Hz video frame rate) do not provide enough information for retrieval. To limit sensitivity to peaks caused by noise, all peaks with a magnitude lower than a predefined threshold (30% in our experiments) of the strongest peak's magnitude are omitted. The remaining dominant peaks are assumed to stably represent the gradient feature within the window and are encoded into our proposed peak-feature.

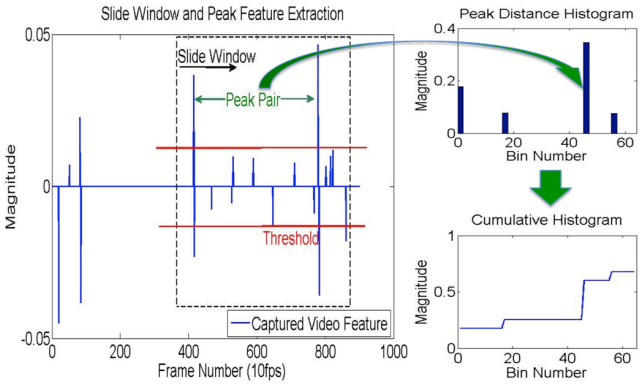

Figure 3: Depiction of a sliding window for extracting the peak-descriptor. To suppress noise related peaks, peaks below a predefined threshold are ignored. Effective peaks pairs create a histogram and the cumulation is used as the descriptor of the window.

The encoding scheme works as follows. A histogram of the pairwise distances between all pairs of peaks is computed. The histogram uses a bin size of eight, which roughly corresponds to a one second distance quantization. The resulting histogram has 64 bins for our window size of 512 frames. Each pair of peaks increases the count in the bin corresponding to their distance (measured by their frame number difference). To model the fact that the stronger peaks are more reliable, the amount of increase is equal to the product of the peaks magnitude. In that way, peaks with larger magnitudes contribute more significantly to the histogram. To ensure comparability between feature windows with different numbers of peaks, we normalize the histogram to sum to one. Our peak-feature is the cumulative histogram of the normalized histogram. For the remainder of the paper, we use this cumulative histogram as it is less prone to the influence of noise caused, for example, by the quantization through the histogram bins.

In summary, our proposed peak-feature is a monotonically increasing 64-dimensional vector with the final element being 1. An example histogram and the corresponding peak-feature is illustrated in Figure 3. The distance between two peak-features can be measured by the Euclidean distance of the 64 dimensional vectors. The peak-feature is invariant to the starting point of the window given that it only encodes the pairwise peak distances. When the window slides across the feature, the peak-feature remains stable as long as there is no peak coming in or going out. For completeness, the exact process is given in Algorithm 1.

An entire video can then be represented as the set of its peak-features, which typically leads to a large set of features describing the video. However, since the peak-feature only depends on the peaks within the window, shifting the window by one frame often results in the same peak-feature (as long as all peaks remain in the window). Empirically, this is the case for about 95% of the peak-features. Accordingly, we represent a video using only its unique peak-features and remove all redundant peak-features from the set of computed peak-features.

Algorithm 1 Extracting peak-feature from window f_{win}

1: $Threshold \leftarrow 0.3$
2: **for** $i = 1$ to N **do**
3: **if** $|f_{win}[i]| < Threshold * max(|f_{win}|)$ **then**
4: $f_{win}[i] \leftarrow 0$
5: **end if**
6: **end for**
7: **for** $i = 1$ to 64 **do**
8: $Histogram[i] \leftarrow 0$
9: **end for**
10: **for** every 2 peaks p_i, p_j in f_{win} **do**
11: $Histogram[dist(p_i,p_j)] \leftarrow Histogram[dist(p_i,p_j)] + |p_i p_j|$
12: **end for**
13: $Histogram \leftarrow Histogram/sum(Histogram)$
14: **for** $i = 1$ to 64 **do**
15: $PeakFeature[i] = \Sigma_{k=1}^{i} Histogram[k]$
16: **end for**
17: return $PeakFeature$

6.2 Efficient Searching

Next, we detail our proposed efficient search algorithm, which leverages the introduced peak-feature for efficient search. Algorithm 2 provides the pseudo-code for our method and will be detailed below. Given a recording of interest, we first extract the peak-features for the video (see *line 6*). Peak features with a high number of strong peaks are typically very distinguishing, having a Euclidian norm that is typically larger than peak-features with weaker or fewer peaks. During the matching process, we select the peak-feature with the largest norm first (see *line 7*).

To search the database for a likely match (see *line 9*), we index the peak-features using a data-structure known as K-d tree, which is widely used for search in high-dimensional search spaces [2]. The main idea of the K-d tree is to recursively split the space with hyperplanes, which iteratively refines the possible location of the data point under examination. In our empirical evaluations, the reference library contains 27-million peak-features representing the 54,000 videos. By leveraging a K-d-tree, we can quickly search for all reference videos that are likely matches. Here, a likely matching video has to be within a Euclidean distance of $\delta \leq 0.7$ from the peak-feature of the captured video[2].

From the likely matches we select the one with the smallest Euclidian distance to the captured video (see *line 10*). For this video our similarity metric from Equation 9 is computed. If the similarity is the best observed similarity thus far, this video is retained as the top candidate and its confidence is increased (see *line 16*). Then, the next strongest peak-feature is obtained (see *line 18*) and evaluated in the same manner (see *line 9-16*). If the retrieved video is the same as the previously selected one, the confidence assigned to this potential match increases (see *line 16*). Otherwise the newly

[2]The value for δ was empirically chosen based on a rudimentary analysis of the resulting accuracy.

found best match replaces the previously selected best video (see *line 12-14*). This process is repeated until the best-matching video remains stable for three consecutive trials.

Algorithm 2 Efficient searching captured feature f_c

1: $BestScore \leftarrow INF$ // best so far score
2: $BestId \leftarrow INF$ // database id of best candidate
3: $ConsecutiveHits \leftarrow 0$ // number of consecutive confirmation of best candidate
4: $MaxHits \leftarrow 3$
5: $Radius \leftarrow 0.7$
6: $PeakFeature \leftarrow extractPeakFeature(f_c)$
7: $CurFea \leftarrow featureOfStrongestPeak(PeakFeature)$
8: **while** $exist(CurFea)$ and $ConsecutiveHits < MaxHits$ **do**
9: $RefFea \leftarrow searchKdtree(CurFea, Kdtree, Radius)$
10: $[CurScore, CurId] \leftarrow findMinSMetric(CurFea, RefFea)$
11: **if** $CurScore < BestScore$ **then**
12: $BestScore \leftarrow CurScore$
13: $BestId \leftarrow CurId$
14: $ConsecutiveHits \leftarrow 0$
15: **else**
16: $ConsecutiveHits \leftarrow ConsecutiveHits + 1$
17: **end if**
18: $CurFea \leftarrow featureOfNextStrongestPeak(PeakFeature)$
19: **end while**
20: return $BestId$

The algorithm proposed above is an offline approach, which can be extended to operate in an online fashion. For offline retrieval, we have access to all the peak-features at once. Hence, we have the luxury of ranking the features by strength. In contrast, for online operation, the video is streamed. Once a new frame is captured a new peak-feature is computed using the 512 most recent frames. If the newly computed feature is unique for the video, i.e., has not been extracted from the video before, the K-d tree is used to search for likely matches within the reference library. Then the best video (i.e., with the smallest Euclidian distance) is fully evaluated using our proposed similarity metric from Equation (9). If the best video is identical to the previously identified one, its confidence is increased. Otherwise it replaces the current best choice.

On Efficiency: Levering the peak-features and the K-d tree based search reduces the search time on average to less than 10*s* (2.8 seconds for each K-d tree search) for a database of 54,000 reference videos. The achieved query time is more than an order of magnitude faster than searching exhaustively through the database, which took 188*s*. The online search can in fact be executed in real time when allowing a latency of 512 frames due to the required temporally preceding information for the peak-feature computation.

7. EVALUATION

For our empirical evaluation, we collected a large collection of reference videos spanning a wide variety of content. Our reference library contains 10,000 blockbuster movies of at least an hour in length, 24,000 news clips ranging from 5 min to 20 min each, 10,000 music videos ranging from 2 min to 7 min each, and 10,000 TV-shows ranging from 5 min to 20 min each. In total, the library indexes over 18,800 hours of video. All features and peak-features from the library are precomputed by leveraging our proposed methods from Sections 5 and 6. For our experimental evaluation we randomly selected 62 sequences as our test set of videos.

For the first set of evaluations the test videos were played on a 24 inch screen with no additional room lighting turned on. We then capture the reflection of the screen emanation from a white wall at a distance of three meters from the screen. To capture the video, a Logitech HD Pro Webcam C920 and a 60D canon DSLR were used. We run the experiment in a home environment as well as in a lab environment. The setting of our experiment is illustrated in Figure 4. These captured videos were then used to execute our attack. For these evaluations we assess the success of the attack with respect to the duration of the captured video and the size of the reference library.

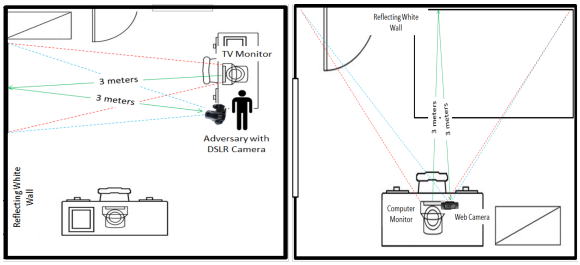

Figure 4: Lab environment (left) and home environment setting (right)

Lights Off

First we evaluate the success rate of our method using a room with the lights off, as commonly occurs when watching TV. A success is the correct identification of the video being watched. We do not leverage any knowledge about the video being played nor do any of our experiments use any knowledge of the scene or the capture distances. The time at which the adversary starts capturing the emanations from the display is chosen at random.

Capture Length	60s	90s	120s	180s	240s	270s
Success Rate	39%	49%	54%	70%	85%	94 %

Table 1: Retrieval success rate with random start point.

For the 62 test sequences we analyzed segments from 60 to 270 seconds long. These segments are processed by the feature and peak-feature extraction procedures. The resulting features and peak-features are then used to infer the best match among the reference library. The experiment is repeated 100 times for each of the different segment lengths, each time choosing a random starting position. Table 1 shows the resulting average success rate over all starting positions. As expected, the longer the captured sequence, the higher the attack's success rate. The results shows that the success rate increases from 39% for a 60 second segment to 94% for 270 seconds, and has nearly a 50% success rate using only 90 seconds of captured emanations. A more detailed analysis of the data reveals that in the limit, the success rate is 100% for each video as subsequences within these videos can always be uniquely identified.

To better quantify the robustness of our approach, we evaluate the ratio in similarity between the video sequence returned as the best match and the true positive. If the ratio is larger than one, that implies the correct video will always be identified. The higher that ratio, the more distinct the retrieval result. Obviously, the outcome also depends on the contents of the reference library itself. The experimental results of the ratio evaluation are shown in Figure 5. The median similarity score ratio rises above one (successful retrieval) between 100 and 120 seconds. For longer sequences, it monotonically increases with increasing segment length.

Beyond the average success rate and robustness, it is also important to understand the best and worst case results. The worst case is

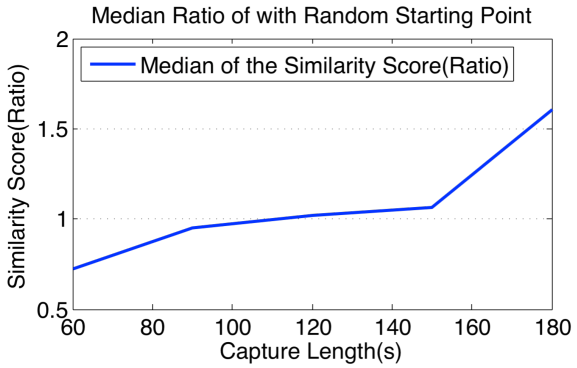

Figure 5: The median ratio within the dataset of 54,000 references.

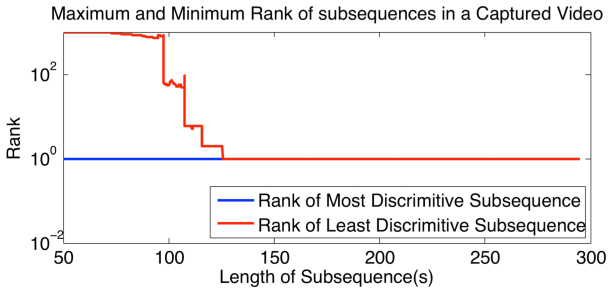

Figure 6: Rank of correct video in the best case (blue) and worst case (red).

especially useful since it provides a measurement for an attacker of how much video is needed to reliably achieve a successful attack. To measure these boundaries we evaluate the retrieval success rates for all possible sub-sequences longer than 10 seconds and all possible starting points for our 62 test sequences. For each of these tests we then rank the retrieved videos by their similarity scores and report the rank of the ground-truth video. If the corresponding video is ranked first the retrieval was successful, otherwise it was not.

Figure 6 shows the rank of the corresponding video with respect to the captured video's length for one of our test videos. The results for the other videos are comparable. In the best case, the corresponding reference video is always ranked first, which means if the attacker is lucky enough, she will be able to retrieve the correct video even if she only captures 10 seconds of video. The results also shows that any captured segment longer than 120 seconds within this particular video can always be successfully retrieved.

Next, we summarize the results on a per-video basis by assigning a video its worst segment's similarity ranking, i.e., its worst possible ranking obtained by any of the corresponding video for any of its segments. This captures the lower bound of the attacks performance for each of our 62 test videos. The results are shown in Figure 10. Expectedly, the variation is the largest for the shortest segments of less than 100 seconds and converges to one with capture length longer than 240 seconds.

7.1 Lights On

The illumination of a scene (e.g., both room and natural light) contribute significantly to the amount of light entering the camera, which in turn influences the brightness level of the captured video. Obviously, screens with lower brightness naturally reduce the light emanation. Therefore, we evaluate the influence of the illumination

Illumination settings	SNR	Segment Length
Normal brightness level room light off	70	180s
50% brightness level room light off	33	270s
Normal brightness level room light on	15	300s

Table 2: Worst case capture length with different illumination settings.

on the performance of our proposed attack. In this experiment we use a 24 inch screen, and the attacker's camera captures the reflection of the screen of a white wall, which is three meters away from the screen. The camera used in the attack is a Canon Rebel T4i DSLR. We captured five videos in each of three different illumination settings: 1) normal screen brightness with room light off, 2) 50% reduced screen brightness level with room light off, 3) normal screen brightness with the room light on. The obtained retrieval results are shown in Table 2 and Figure 7.

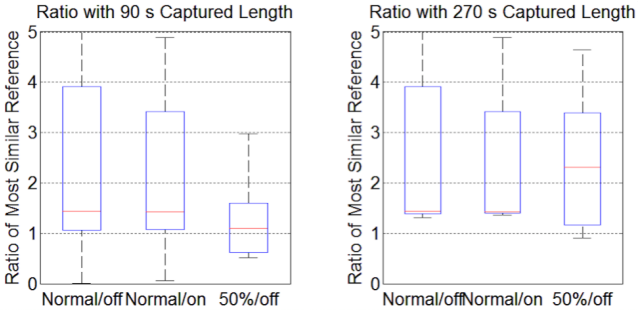

Figure 7: Ratio to second-best under different illumination conditions.

The results indicate that higher screen brightness levels make the retrieval slightly more successful as it takes shorter segment lengths for successful retrieval and the similarity ratio is higher. However, the influence of the screen brightness seems marginal. It can also be seen from Table 2 and Figure 7 that even with the room light on, our attack is successful with moderate segment length, which we attribute that to our robust similarity metric. The only effect that both the lower screen brightness and the active room light have is that it mandates that longer segments are necessary for successful retrieval in the worst case. This is expected, as in both cases, smaller, less significant brightness changes are not detectable anymore. Accordingly, there are fewer distinguishing elements we can use. In the case of the active room light, we only failed once when retrieving a video based on a segment that was 270 seconds, but succeeded with a 300 second segment. It is worth noting that in the case of an active room light, a human observer is not able to perceive the resulting subtle intensity changes on the wall.

7.2 Impact of Screen Size

The amount of light captured by a camera not only depends on the screen's illumination setting but also on the actual screen size as it influences the amount of light emitted into the environment. Generally, bigger screens emanate more light, which leads to higher quality video capture. To evaluate the impact of screen size, we performed an experiment in which we used differently sized LCD displays. In particular, we used displays with 24 inch, 30 inch, and 50 inch screen sizes. We again use a Canon Rebel T4i DSLR to capture the video of the back wall, which is 3 m away from the screen. For each screen size we capture five videos. The resulting required worst case segment lengths for successful retrieval are shown in Table 3 while Figure 8 shows their distribution. The SNR

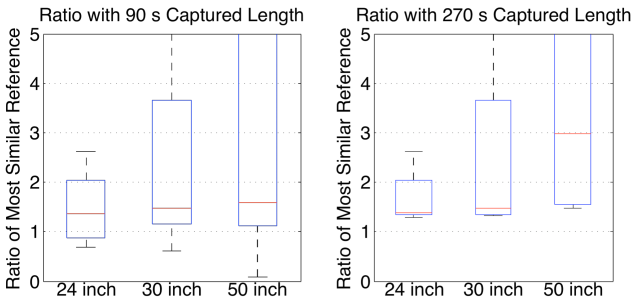

Figure 8: Boxplot of the second-best ratio w.r.t. different screen sizes.

is lower because the experiment was performed in a different room with a lot of light-absorbing materials.

Screen Size	SNR	Worst Case Length
24 inch	5	270s
30 inch	48	180s
50 inch	109	180s

Table 3: Worst case capture length with different screen size.

Expectedly, the larger screen size supports better retrieval for shorter segments. The shorter segments that fail on the 24 inch screen can often be successfully retrieved with the 30 and the 50 inch screens. The similarity ratio is higher on larger screens leading to more robust identification.

7.3 Impact of Reference Library Size

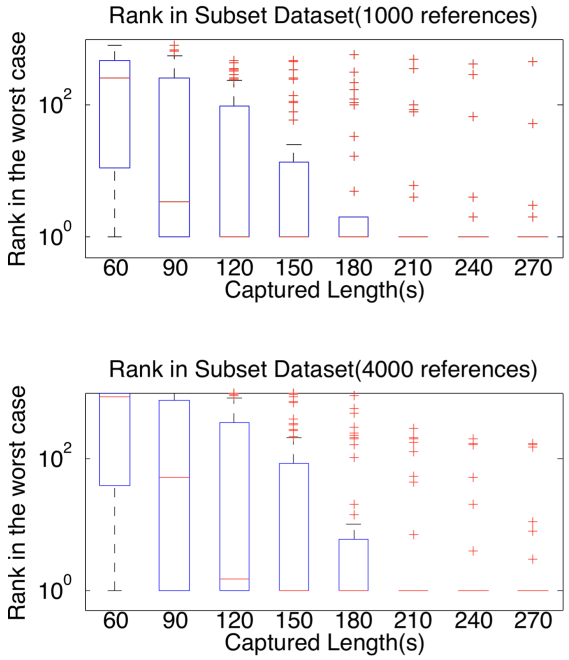

Figure 9: Rank of correct video among libraries of size 1,000 and 4,000.

The retrieval results are influenced by the distribution of the videos within the database and the size of the database. To characterize the change in behavior we compute the worst case ranking for two reference libraries consisting of 1000 and 4000 videos respectively. The results are shown in Figure 9. As expected, it can be seen that the larger the database, the longer the segments have to be to guarantee a successful retrieval. However, the increase in segment length with respect to the increase in database size is moderate. For example, for an increase in database size from 4,000 to 54,000 videos (13.5x), the segment length only increases by 20% (from approximately 200 seconds to 240 seconds). We predict that this increase will decline even more for larger databases as the probability of two identical video segments appearing in different videos exponentially decreases with the length of the segment.

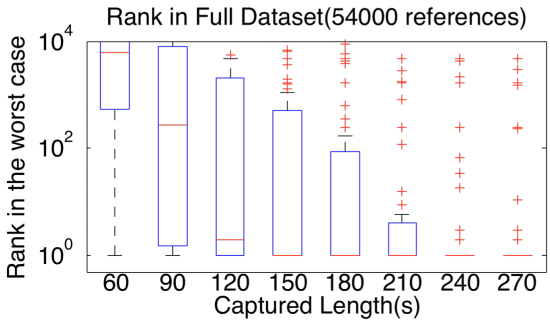

Figure 10: Rank of correct video (among 54,000 videos).

7.4 As Seen From Outdoors

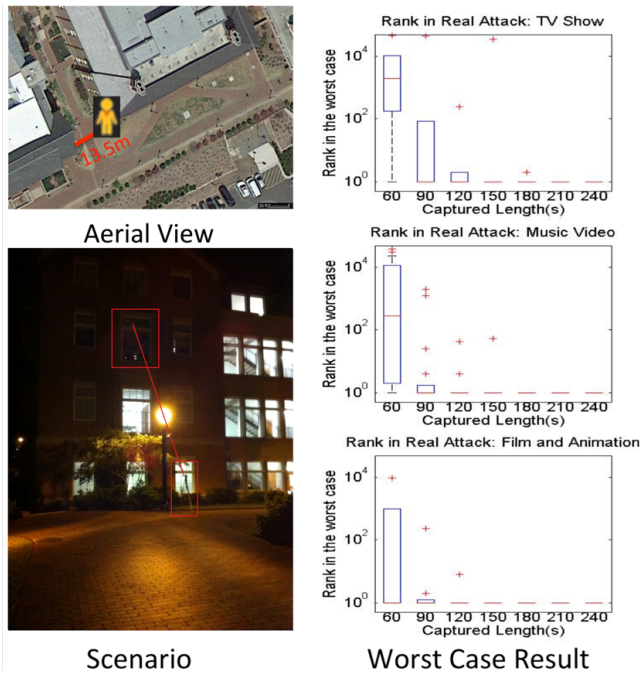

Figure 11: TV reflection in the room is captured from a distance of 13.5 meters (left). The worst case results (right) are illustrated for different types of videos: TV shows, music and film from top to bottom. All segments longer than 180s were successfully retrieved.

To further demonstrate the practicality of our proposed attack, we tested its effectiveness from outdoors. We captured the emanations seem on an outside window of a room with a TV showing 60 of our test sequences. In this scenario, the attacker was positioned on the sidewalk observing the third floor office window of

the room with the TV (see Figure 11). The TV emanations reflected off the beige ceiling of the room and towards the window which was situated orthogonal to the TV. The TV is 13.5 meters away from the adversary. For completeness, we evaluated our approach using videos from varying categories of media that include TV shows, music videos and films. 20 samples of each video type were captured. Figure 11 (right) shows the worst case result with respect to different subsequences. The results indicate similar success across all videos tested, and in all cases, we were able to perform the confirmation attack.

To guage the robustness of our approach, we further experimented with recordings captured at much further distances. In this case, the attacker was positioned on the sidewalk 70.9 meters from the building; the TV was playing in the same third-floor room as in the previous experiment. TV emanations were captured from the ceiling reflection with the same Canon Camrecorder. 20 sequences randomly selected from different categories are tested. The proposed approach successfully retrieved 18 sequences out of them within 5 minutes. The experimental setting and results are depicted in Figure 12. The results are compared with that of direct view and 13.5 meter reflection (Figure 12 bottom). In the worst case, the sequence can usually be retrieved within 100 seconds at 13.5 meters away, compared to 190 seconds, on average, from 70 meters away.

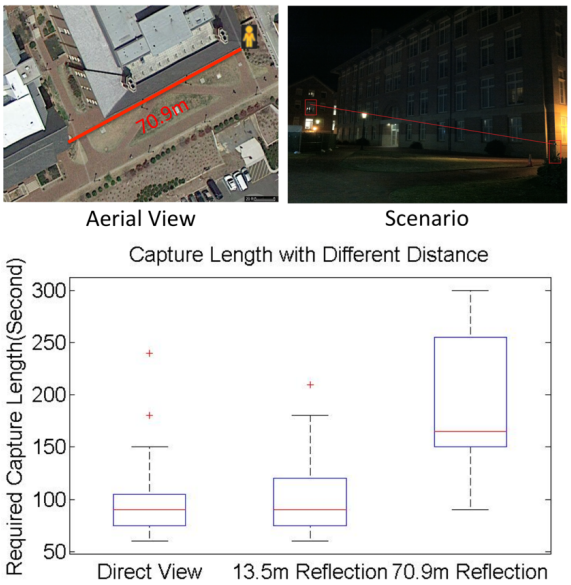

Figure 12: TV reflection in the room is captured from a distance of 70.9 meters(top). The camera and the window are labelled in red(top right). The required capture length is compared with direct view and 13.5 meter reflection (bottom). It takes longer for successful retrieval with longer distance.

8. MITIGATIONS

The simplest mitigation is to cover the windows of the room with blackout curtains to effectively avoid the leakage of the light to the outside. To guage the effectiveness of such a defense we performed a rudimentary experiment with vinyl blinds and curtains (see Figure 13)[3]. The setup was the same as for the attack carried out at 13.5 meters outdoors, except for the use of shades. In this experiment, only two samples were tested in each case. For the case of vinyl blinds and a standard beige curtain with brown stripes, we were still able to determine 3 of the 4 videos being watched after capturing 270s worth of footage. The other video failed to be recovered even after 5 mins. We were unable to confirm any of the watched content when thicker, room darkening, (black) curtains were used.

[3]The brighter pattern in the middle picture is caused by a reflection on the vinyl blinds from an outside street lamp.

Figure 13: Captured image directly from window (left), through vinyl blinds (middle) and through a curtain (right).

If the use of curtains is not desired the screen brightness could be lowered to increase the SNR of any captured video. Our experimental evaluation demonstrated though that this has only a limited effect on twarting the attack. Our experiments show that retrieval will still be possible as long as the brightness change is perceptible. Although this strategy would not prevent the attack altogether, lowering the screen brightness will increase the burden on the attacker as longer observations would be required to successfully carry out the attack. Similarly, the burden on the attacker can be increased if a bright room light is used as that would increase the noise level in the captured signal.

Another defensive strategy may be to install a flood light next to any window of the room so as to effectively blind a camera that tries to observe the diffusions through the window. Doing so would prevent the camera from capturing the subtle brightness changes required to successfully execute the attack. That said, a motivated attacker could overcome this defense by using sophisticated high dynamic range image cameras, which can capture a large dynamic range of light intensities. Alternatively, our attack could be mitigated by installing an adaptive lighting system, which measures the emitted light and counters any brightness change of the emitted light. Doing so would help maintain a constant amount of light emission and would not reveal the brightness change information to an outside observer. Obviously, these defenses would not be popular in densely populated areas as the outdoor light emissions would likely not be appreciated by neighbors.

9. CONCLUSIONS

We propose a novel method to identify the video content shown on a victim's screen using recordings collected in a number of practical scenarios (e.g., observations of light effusions through the windows or off the walls) and at great distances (e.g., 70m away). Our attack shows reliable identification of the content being watched in a wide range of evaluated scenarios. The robustness of the attack is due to a novel application of unique feature sets, a well suited similarity metric, and the development of efficient indexing structures for performing rapid matches in near real-time. Our empirical results show that we can successfully confirm hypotheses while capturing short recordings (typically less than 4 minutes long) of the changes in brightness from the victim's display.

10. ACKNOWLEDGEMENTS

We are thankful to Michael Bailey, Kevin Snow, and the anonymous reviewers for their insightful comments and suggestions. This work is supported in part by a grant from the National Science Foundation, under award number 1148895.

References

[1] M. Backes, M. Durmuth, and D. Unruh. Compromising reflections-or-how to read LCD monitors around the corner. In *Proceedings of the IEEE Symposium on Security and Privacy*, 2008.

[2] J. L. Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9): 509–517, 1975.

[3] A. Buades, B. Coll, and J.-M. Morel. A review of image denoising algorithms, with a new one. *Multiscale Modeling & Simulation*, 4(2):490–530, 2005.

[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*, pages 537–550. ACM, 2011.

[5] C. Faloutsos, M. Ranganathan, and Y. Manolopoulos. Fast subsequence matching in time-series databases. *ACM International Conference on Management of Data (SIGMOD)*, 23(2), 1994.

[6] D. Gomery. As the dial turns. *The Wilson Quarterly*, pages 41–46, 1993.

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

[8] G. E. Healey and R. Kondepudy. Radiometric ccd camera calibration and noise estimation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 16(3):267–276, 1994.

[9] M. G. Kuhn. Compromising emanations of lcd tv sets. *Electromagnetic Compatibility, IEEE Transactions on*, 55(3): 564–570, 2013.

[10] B. Langmead, C. Trapnell, M. Pop, S. L. Salzberg, et al. Ultrafast and memory-efficient alignment of short dna sequences to the human genome. *Genome Biol*, 10(3):R25, 2009.

[11] J. Liu, Z. Huang, H. Cai, H. T. Shen, C. W. Ngo, and W. Wang. Near-duplicate video retrieval: Current research and future trends. *ACM Computing Surveys (CSUR)*, 45(4): 44, 2013.

[12] Y.-S. Moon, K.-Y. Whang, and W.-S. Han. General match: a subsequence matching method in time-series databases based on generalized windows. In *Proceedings of the 2002 ACM SIGMOD international conference on Management of data*, pages 382–393. ACM, 2002.

[13] R. Raguram, A. M. White, Y. Xu, J.-M. Frahm, P. Georgel, and F. Monrose. On the privacy risks of virtual keyboards: automatic reconstruction of typed input from compromising reflections. *Dependable and Secure Computing, IEEE Transactions on*, 10(3):154–167, 2013.

[14] A. Torralba and W. T. Freeman. Accidental pinhole and pinspeck cameras: revealing the scene outside the picture. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 374–381. IEEE, 2012.

[15] Y. Tsin, V. Ramesh, and T. Kanade. Statistical calibration of ccd imaging process. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 1, pages 480–487. IEEE, 2001.

[16] B. Widrow and I. Kollár. *Quantization noise: roundoff error in digital computation, signal processing, control, and communications*. Cambridge University Press, 2008.

[17] 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*, pages 1063–1074. ACM, 2013.

[18] L. Zhang and Y. Rui. Image search–from thousands to billions in 20 years. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP)*, 9(1s):36, 2013.