# Detecting Spies in IoT Systems using Cyber-Physical Correlation

Brent Lagesse*, Kevin Wu†, Jaynie Shorb‡ and Zealous Zhu§
Computing and Software Systems
University of Washington Bothell
Bothell, WA USA
Email: *lagesse@uw.edu, †kevinw9@uw.edu, ‡jaynies@uw.edu, §zealousz@uw.edu

***Abstract*—The emerging ubiquity of IoT devices with monitoring capabilities has resulted in a growing concern for user privacy. Whereas previous research has focused on preventing an attacker from learning about user activities through analyzing data, we are concerned with the problem of attackers that utilize either hidden or compromised IoT devices to spy on users. Our work addresses the challenge of automatically identifying devices that are streaming privacy-intruding information about a user despite the presence of both encryption and a large number of wirelessly connected devices within range of the user.**

**We present a framework for inducing a signal in the physical world and then detecting its digital footprint when devices are monitoring the user. Our approach only requires the user to have a device such as a smart phone with the ability to enter into promiscuous packet capture mode. As an example application, we have set up a hidden camera and conducted 222 trials of devices that are not recording the user along with 680 trials of the hidden camera under a wide variety of configurations. In most of the environments examined, we were able to to detect over 90% of the hidden cameras while producing less than 6% false positives. As a result, this paper provides significant evidence that our approach is feasible for detecting spying devices.**

## I. Introduction

The increased deployment of Internet of Things (IoT) devices has resulted in more pervasive and useful services to users. Unfortunately, these devices raise privacy concerns since they may be placed without a person's knowledge as has happened in several notable cases in short-term rentals [1], [2], [3]. Hidden cameras are trivial for an attacker to acquire. Amazon.com currently lists 4,527 cameras in their Hidden Cameras category including 1,214 that list Wi-Fi as a feature. 828 of these Wi-Fi hidden cameras can be purchased for under $100. In addition to maliciously placed devices, devices might be used by an attacker who has compromised the device[4], [5], [6], [7].

Consider the following scenarios. In the first case, an attacker has placed a hidden web cam in a room that you are visiting. Such a web cam is designed to blend in with the surroundings and can easily occupy less than 5 $cm^2$ of surface area. The camera can utilize encryption or run on a wireless network for which we do not have an access key, so its video content is not easily distinguishable from the dozens of other Wi-Fi devices that are frequently visible in many houses, hotels, and apartments. In the second case, the many camera-enabled devices that surround us every day such as web cams, TVs, laptops, and cell phones are potential sources of spying. Such devices stream information about the user to remote sources because the user does not understand the functionality provided by the manufacturer or application. In the third case, an attacker has exploited a vulnerability and compromised the device [8]. In this case, as in the first, it is difficult for the user to detect that a device has been compromised, and prior work in the area of detecting hidden cameras focuses on detecting unknown hidden cameras rather than maliciously manipulated cameras that the user knows are present.

Previous research in privacy in IoT spaces has generally focused on protecting the user from an attacker that discovers their activities [9], [10], [11]. The focus of our work examines an attacker model that is a flipped version of the previously examined attacker model. In the flipped attacker model, the attacker attempts to record information about a user without the user's knowledge. In this case, the user wants to identify that they are being monitored.

Previous work in detecting devices that are surreptitiously recording user activity has largely focused on cameras. In particular, this work [3] focuses on identifying IR used in night vision or the presence of unexplained Wi-Fi devices in order to determine locations to manually search for cameras. Additional techniques exist that require the user to acquire and construct additional hardware based on BeagleBone[12] or purchase expensive commercial detectors [3].

Our work augments these existing approaches and provides a framework that is more widely applicable to IoT devices that could be used to surveil users. To this end, our work contributes the following advancements:

- A framework for manipulating the physical environment to induce detectable signals in the digital environment
- An implementation of this framework for Wi-Fi streaming cameras
- A computationally efficient system that runs on a mobile device with minimal additional hardware
- An analysis of the parameter space of the system to demonstrate its efficacy for detecting Wi-Fi streaming cameras

Additionally, our system addresses parts of the problem that previous approaches do not address. Previous work helps to identify the presence of hidden cameras, but not their status of actively streaming video of the user. Additionally, previous

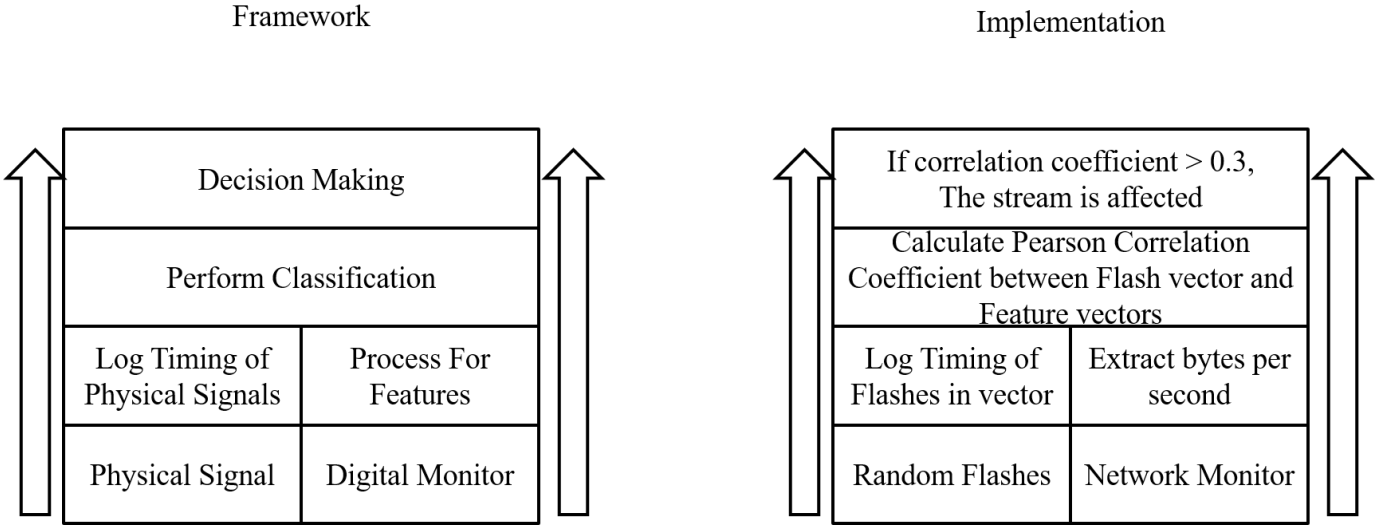



Fig. 1. Framework and Implementation

work relies on the hidden cameras being in night vision mode to detect RF signals. Furthermore, techniques that rely on intercepting the data and reconstructing it are not effective if the camera is using an encrypted connection that the user does not have the key to.

## II. Background

In this section we describe background information related to the two main techniques used in our approach. First, we describe previous work in signal detection and then we describe the basics of interframe compression techniques used in video codecs.

### A. Signal Detection

Work performed in [9], demonstrates that IoT smart home devices will frequently produce peaks of network traffic when the device is monitoring a user that is performing the activity it is designed to monitor. In their work, they used the sending/receiving rates of the streams to map live traffic to user behavior. This research indicates that network streams of IoT devices have attributes that can be manipulated by the users. Work [13] has been performed that focuses on identifying applications based on network traffic and header information by using Naïve Bayes Machine Learning. Meidan et al. [14] showed the feasibility of identifying IoT devices based on wireless network traffic. They used supervised learning to evaluate sessions of both IoT and other electronic devices. [15], [16] have induced delays in traffic and performed timing analysis on low-latency anonymizing systems in order to de-anonymize users.

### B. Interframe Video Compression

Interframe compression techniques examine each frame of video and if a section of the two adjacent frames have not changed, then the compression algorithm causes the unchanged values to be copied directly from the previous frame. Likewise, if uniform changes are made to a particular section, then the compression algorithm can cause a copy of the preceding frame to be used with a uniform transformation being made on those pixels. To the extent of our knowledge, all commonly used video codecs for streaming use interframe compression techniques including H.264[17], MPEG-2[18] and MPEG-4[19].

As a result of the way these algorithms work, when there are significant changes to the video, significantly more data must be sent. We leverage this fact to create signals in the physical world that translate into signals that we can detect in the digital world.

## III. Design

### A. System Model

Our work is focused on pervasive IoT systems that primarily operate indoors in houses and offices. These systems contain large numbers of resource constrained devices with rich abilities to sense and distribute information with some capability of processing information. These devices are typically wirelessly connected. In particular, this system affects our design considerations in that there are limitations on the amount of data that can be stored on the individual devices, so we expect that wireless data streaming will be prevalent.

### B. Attacker Model

Our approach addresses three different attacker models. We refer to these models as the hidden device model, the compromised device model, and the unexpected recording model. Our system is useful in detecting attackers utilizing any of the three models.

*1) Hidden Device Model:* In the hidden device model, an attacker has hidden a monitoring device in the space in which a user will occupy. Examples of this may include a short-term rental [1] or could be a compromise of a person's own home.

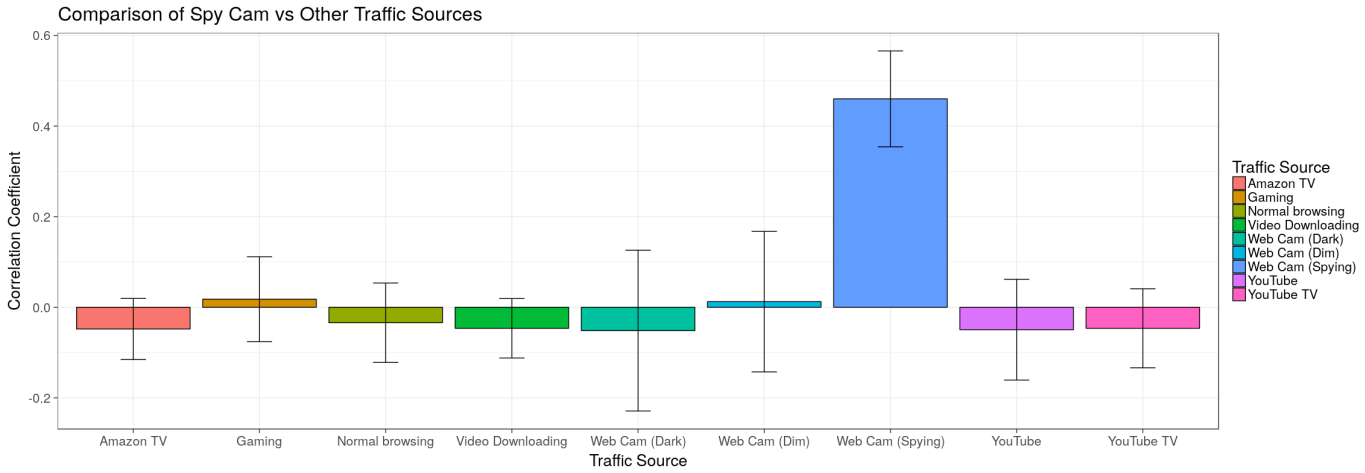



Fig. 2. Detection of Spy Cam vs Other Data Streams

In this model, the user is not initially aware that the device exists. This is the model that most previous work [3] focuses on detecting.

*2) Compromised Device Model:* In the compromised device model, an attacker utilizes a user's own device to record their actions. This model has been frequently documented[20], [4], [7] as a problem with IoT devices.

*3) Unexpected Recording Model:* In the unexpected recording model, the attacker is the device or application manufacturer. This model has been acknowledged multiple times as companies are frequently recording information from IoT devices for either improving functionality or targeting products and advertising [21], [22], [23], [24].

### C. System Design

Figure 1 shows our generic framework and the specific implementation used in our experiments. Our design consists of a physical and a digital side. In the physical side, a signal is induced that is designed to cause IoT devices to respond digitally. On the digital side, our system records digital outputs that occur during this time. The digital output is processed to extract features necessary for classification, and then the features and the information about the physical signal are processed by a classification algorithm (in our implementation, we only record whether or not the physical signal was active; however, other information such as intensity or frequency of the signal could also be recorded). The output of the classification algorithm is then used to decide which streams were recording the user, if any.

In the implementation we evaluate in this paper, we used the flash on a camera and logged a vector of 1 if the flash occurred in a particular second and 0 if a flash did not occur. Simultaneously, we recorded network traffic in promiscuous mode and created a vector of the number of bytes transmitted each second for each MAC address. We then calculated the Pearson correlation coefficient using equation 1 between the flash vector ($x$) and the bandwidth vectors ($y$). If the correlation coefficient was greater than 0.3, we classified the MAC address as recording the subject. We chose the value of 0.3 after experimental evidence in 3 showed that it provided a good trade-off between true positives and false positives. Note that in 2 we demonstrate the main point is the difference in correlation coefficient between a spying cam and a non-spying device, not the absolute value of the correlation coefficient. Since we were only recording bandwidth, we are able to perform this calculation on encrypted data. Likewise, we also do not need to store the data recorded for more than one second, so we do not consume a significant amount of memory.

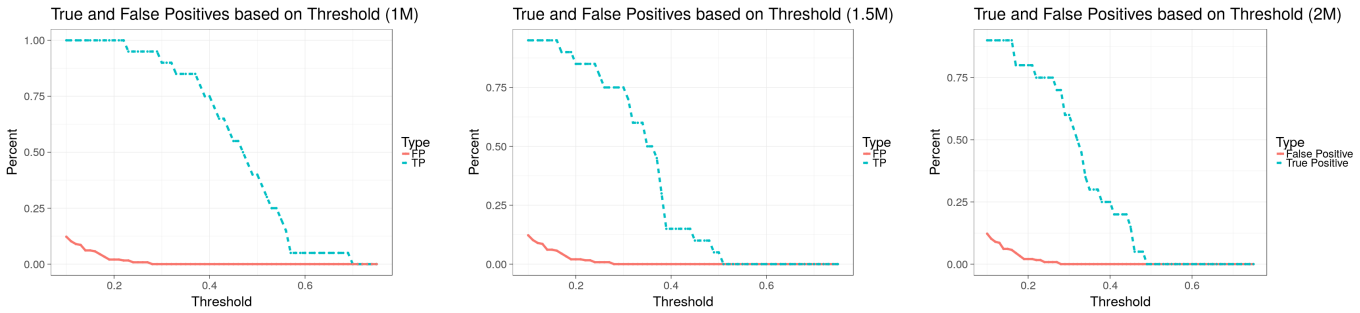
$$r = \frac{\Sigma(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\Sigma(x_i - \bar{x})^2 \Sigma(y_i - \bar{y})^2}} \quad (1)$$

## IV. Evaluation

We have implemented our system on several Android phones including a Moto Z running Android version 6.0.2, and an HTC M7 running Android 5.0.2. We collected PCAP data using Kismet Wi-Fi PCAP Capture 2012.12.1 on Android. It is not possible to capture packets on Android in promiscuous mode using a stock android firmware in userspace, so rather than rooting the phones for this experiment, we utilized a Sabrent NT-WGHU Wireless adapter based on the 8187L chipset attached to the Android phone via a USB OTG cable. We chose this approach since we believe it is more likely a user would be willing to attach a piece of inexpensive hardware than to root their phone; however, our framework is not directly tied to this design decision, so if mobile device OSes allow for promiscuous packet capture, there is no requirement for an external device.

We evaluated our approach by capturing live streams from a DLINK DCS936l web cam in a variety of settings and for a variety of traffic sources. Unless otherwise noted, our experiments take place in a large room with ambient light and the mobile phone placed 2 meters from the camera. Error bars represent one standard deviation above and below the mean.

For each environment represented in the figures in this paper, we performed 20 experiments that lasted for 60 seconds




(a) 1 Meter from Camera (b) 1.5 Meters from Camera (c) 2 Meters from Camera

Fig. 3. Comparison of True and False Positives for Varying Classification Thresholds

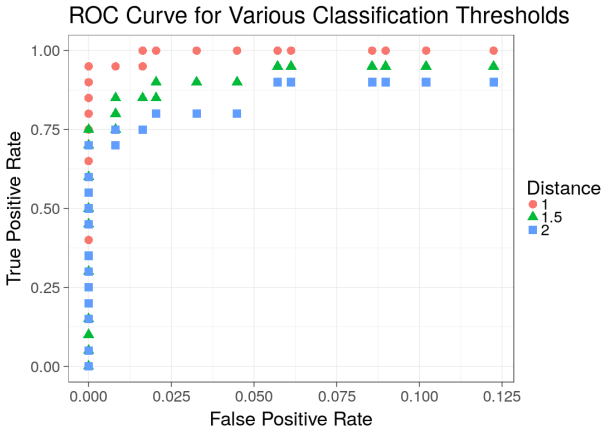



Fig. 4. ROC Curve for 1, 1.5, and 2 Meters

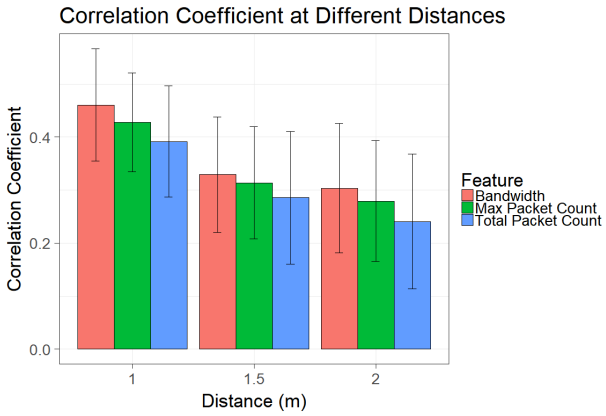



Fig. 5. Comparison of Correlation Coefficient From 1, 1.5, and 2 Meters using 3 Different Features

each for each combination of variables resulting in 680 total experiments for spying devices. For the data recorded in figure 2 we recorded 222 60 second experiments with devices that were not spying.

After we showed that we can identify the signal when a web cam is present in the room, we confirmed that our system would not identify non-web cams incorrectly. We identified traffic from a variety of types of applications and ran our algorithm on that traffic.

Figure 2 demonstrates the differences between spy cameras that are observing us and other types of traffic. The web cams used in this experiment result in most of the false positives shown in Figure 3 at lower thresholds. Note that the webcam used for the dark room and dim room experiments is the same camera as used in the spying example. In addition to measuring true and false positives as a function of the threshold, we have also plotted these values for each distance in the form of a ROC curve in figure 4. Note that in this figure we zoom in on the leftmost portion of the curve because after the 12.5% false positive rate, the true positive rate goes to 100%.

As demonstrated in Figure 5, the correlation coefficients all had similar variances (as shown by the standard deviation error bars in the figure); however, in our tests, bandwidth was consistently a better feature to measure correlation than the other features that we tested. We also tested the bandwidth only at greater distances. Figure 6 shows that the signal does degrade over time. At 4 meters, the correlation coefficient is still noticeably different than the non-signal traffic, but produces significant false positives. In section V-B we address our future work plans to extend the system capabilities to greater distances.

Figure 3 demonstrates the relationship between true positives, false positives, and the correlation coefficient threshold for classification. All 3 distance experiments show a similar trend. False positives approach 0 quickly as the threshold approaches 0.3 and true positives remain high until around 0.3. Through further experimentation, we intend to adapt this threshold value based on empirical evaluation of the quality of the flash on the phone.

Figure 7 demonstrates the effect of pointing the phone at different angles from the camera. This experiment was run to determine whether or not the phone must be pointed directly at the camera or if there is tolerance for ”missing” the camera. These results show that the ability to detect a camera appears dependent on the flash of the individual phone as we believe the angle of lighting varies from camera LED and enclosure. For phones that can detect the camera at a variety of angles, this would drastically reduce the amount of

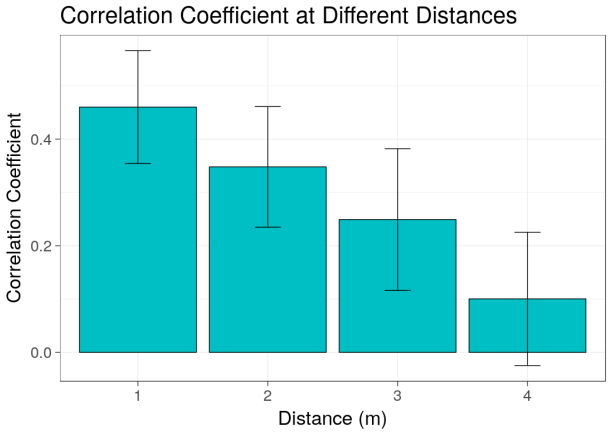



Fig. 6. Correlation coefficient degradation over distance

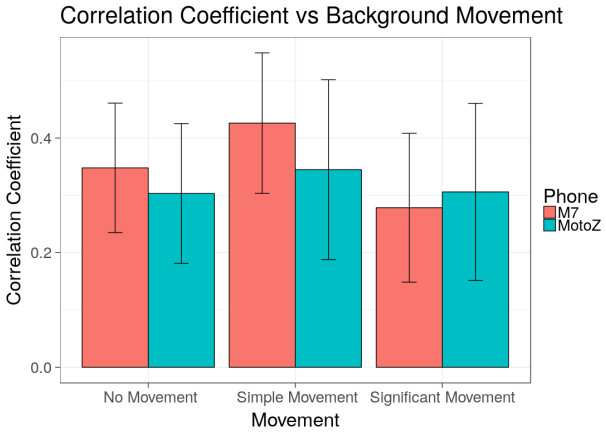



Fig. 8. Correlation coefficient with varying levels of movement

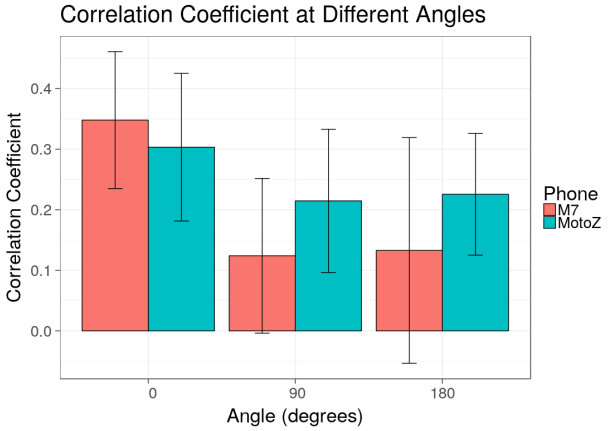



Fig. 7. Correlation coefficient when the flash is facing the camera, at a 90 degree angle, and facing away from the camera

scan time necessary for the user to detect hidden cameras in their environment. In our future work, we will be examining a custom-built array of LEDs for flash and will more thoroughly examine the effects of different LEDs.

Figure 8 demonstrates the effect of background movement on the correlation coefficient. In this experiment, simple movement was caused by a rotating fan in the room and significant movement was caused by a person walking around the room during the recording. The results show that while there may be differences in the quality of signal we can recover, the presence of movement does render the system unusable.

## V. Discussion

### A. Limitations

While our approach has been shown to be highly effective in a number of environments, it is not without limitations. First, the IoT device must transmit data in a form that we can learn the time and size of its transmissions. This means that without some existing compromise of the network, it is unlikely that we could apply this approach to wired IoT devices. While we focused on Wi-Fi in this paper, our approach would work on any wireless communications protocol that we had a network interface capable of listening in promiscuous mode with the exception of possibly unidirectional wireless transmissions.

Our approach also assumes that the hidden IoT device is streaming our information in real time. If the device only saves the data to a memory card, then we would not be able to use this approach to detect it. Likewise if the device records the data and the adversary downloads it as a file in the future, then we would fail to detect it. If the adversary stores the data and then streams it a later time, we would be able to detect it if our our system was still observing the network at that later time; however we have not implemented this functionality in the current system.

If an adversary was aware of our approach and modified their video compression algorithm so that it did not use interframe communication, it manipulates the bandwidth usage so that it avoids peaks, or it sends cover traffic, they might be able to evade our current approach; however, these approaches would degrade the quality of the data stream or consume significantly more resources which might be noticeable. More work would need to be done in detecting these types of data normalization techniques in our environment, but previous work [25] has demonstrated that these approaches are not always effective.

### B. Future Work

In the future, we plan to extend our techniques and experiments to other IoT devices that record and stream physical user data. [9] has noted that other IoT devices demonstrate similar pattern as video cameras when observing an environment, so we believe that this approach will be applicable to other IoT devices. For example, we plan to expand our work to audio recording devices and devices triggered by motion detection.

The experiments we have conducted analyze a raw data stream. We are beginning to test pre-processing techniques that we believe will help improve classification. In particular, we are examining algorithms to filter noise caused by the environment, the protocols, or the device itself prior to running our analysis. Additionally, we will examine other classification

techniques. Correlation coefficients were chosen because of the low cost of calculating those values on a mobile device and the lack of requirement to train and distribute a model beforehand. This approach has proven to be highly effective in the environments we tested, but we will also be applying machine learning techniques to see if we can further improve our results.

We will also test the extent to which brighter lights affect video recording devices. In these experiments we used the stock flash LED on a variety of phones. We are currently building a Moto Mod for the MotoZ phone that will enable us to test the flash at a variety of brightnesses and color temperatures beyond the capabilities of the stock LED used in phones. For all these additional classifiers, we also want to examine the effects of various parameters on the speed of detection including the true and false positives as a function of the amount of time spent inducing signals.

In the case that the techniques we use for filtering and classification are too computationally expensive for the phone, we are also building an extension for our system that will enable us to offload computation to the cloud when available.

## VI. Conclusion

In this paper we have presented our work on detecting hidden IoT devices that are monitoring users without their knowledge. We have demonstrated the feasibility of manipulating the physical world in a way that produces a digital footprint that can be detected by eavesdropping on wireless communications, even if those communications are encrypted and we do not have permission to join the network. We have demonstrated that we can accomplish this without the burden of expensive or cumbersome additional hardware. We have also conducted a number of experiments to demonstrate the extent to which our technique is feasible.

The main take-away from this work is that it is possible to affect the digital world with actions in the physical world and detect hidden IoT devices that are violating a users' privacy with devices that are commonly owned by the users. This approach can be augmented with existing approaches to create a system that is better at detecting privacy-violating IoT devices than currently exists. As we continue to evolve these techniques, users will gain additional confidence in maintaining their privacy in a world where hidden, privacy-violating devices are becoming increasingly more available and inexpensive.

## References


[1] "Yvonne Edith Maria Schumacher vs Airbnb, Inc., a foreign corporation, and Fariah Hassim and Jamil Jiva." [Online]. Available: https://cdn2.vox-cdn.com/uploads/chorus_asset/file/5398067/1-main.0.pdf

[2] J. Steinberg, "These Devices May Be Spying On You (Even In Your Own Home)." [Online]. Available: https://www.forbes.com/sites/josephsteinberg/2014/01/27/these-devices-may-be-spying-on-you-even-in-your-own-home/

[3] P. Polstra, "Am I Being Spied On? Low-tech Ways Of Detecting High-tech Surveillance," Las Vegas, NV, Aug. 2014.

[4] Y. M. Pa Pa, S. Suzuki, K. Yoshioka, T. Matsumoto, T. Kasama, and C. Rossow, "IoTPOT: Analysing the Rise of IoT Compromises | USENIX." Usenix, 2015. [Online]. Available: https://www.usenix.org/conference/woot15/workshop-program/presentation/pa

[5] B. Krebs, "Hacked Cameras, DVRs Powered Todays Massive Internet Outage Krebs on Security." [Online]. Available: https://krebsonsecurity.com/2016/10/hacked-cameras-dvrs-powered-todays-massive-internet-outage/

[6] S. Fogie, "Abusing and Misusing Wireless Cameras," Sep. 2007. [Online]. Available: http://www.informit.com/articles/article.aspx?p=1016099

[7] B. Herzberg, D. Bekerman, and I. Zeifman, "Breaking Down Mirai: An IoT DDoS Botnet Analysis," Oct. 2016. [Online]. Available: https://www.incapsula.com/blog/malware-analysis-mirai-ddos-botnet.html

[8] R. M. Ogunnaike and B. Lagesse, "Toward consumer-friendly security in smart environments," in *2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, Mar. 2017, pp. 612–617.

[9] N. Apthorpe, D. Reisman, and N. Feamster, "A Smart Home is No Castle: Privacy Vulnerabilities of Encrypted IoT Traffic," *arXiv:1705.06805 [cs]*, May 2017, arXiv: 1705.06805. [Online]. Available: http://arxiv.org/abs/1705.06805

[10] A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram, "Blockchain for IoT security and privacy: The case study of a smart home," in *2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, Mar. 2017, pp. 618–623.

[11] D. Chen, D. Irwin, P. Shenoy, and J. Albrecht, "Combined heat and privacy: Preventing occupancy detection from smart meters," in *2014 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, Mar. 2014, pp. 208–215.

[12] "BeagleBoard.org - bone." [Online]. Available: http://beagleboard.org/bone

[13] A. W. Moore and D. Zuev, "Internet traffic classification using bayesian analysis techniques," *ACM SIGMETRICS Performance Evaluation Review*, vol. 33, no. 1, p. 50, Jun. 2005. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1071690.1064220

[14] Y. Meidan, M. Bohadana, A. Shabtai, J. D. Guarnizo, M. Ochoa, N. O. Tippenhauer, and Y. Elovici, "ProfilIoT: a machine learning approach for IoT device identification based on network traffic analysis." ACM Press, 2017, pp. 506–509. [Online]. Available: http://dl.acm.org/citation.cfm?doid=3019612.3019878

[15] S. J. Murdoch and G. Danezis, "Low-cost traffic analysis of Tor," in *2005 IEEE Symposium on Security and Privacy (S P'05)*, May 2005, pp. 183–195.

[16] V. Shmatikov and M.-H. Wang, "Timing analysis in low-latency mix networks: attacks and defenses." Springer-Verlag, Sep. 2006, pp. 18–33. [Online]. Available: http://dl.acm.org/citation.cfm?id=2163273.2163275

[17] "H.264:Advanced video coding for generic audiovisual services." [Online]. Available: https://www.itu.int/rec/T-REC-H.264

[18] "MPEG-2 | MPEG." [Online]. Available: https://mpeg.chiariglione.org/standards/mpeg-2

[19] "MPEG-4 | MPEG." [Online]. Available: https://mpeg.chiariglione.org/standards/mpeg-4

[20] T. Yu, V. Sekar, S. Seshan, Y. Agarwal, and C. Xu, "Handling a trillion (unfixable) flaws on a billion devices: Rethinking network security for the Internet-of-Things." ACM, Nov. 2015, p. 5. [Online]. Available: http://dl.acm.org/citation.cfm?id=2834050.2834095

[21] "Samsung's warning: Our Smart TVs record your living room chatter." [Online]. Available: https://www.cnet.com/news/samsungs-warning-our-smart-tvs-record-your-living-room-chatter/

[22] "5 Domestic Smart Devices That Are Spying On You Right Now." [Online]. Available: http://www.makeuseof.com/tag/5-domestic-smart-devices-spying-right-now/

[23] S. Burke, "Google admits its new smart speaker was eavesdropping on users," Oct. 2017. [Online]. Available: http://money.cnn.com/2017/10/11/technology/google-home-mini-security-flaw/index.html

[24] N. Nguyen, "If you have a smart TV, take a closer look at your privacy settings," Mar. 2017. [Online]. Available: https://www.buzzfeed.com/nicolenguyen/here-are-the-privacy-settings-you-should-look-at-if-you-have?utm_term=.nczbY9Ezkz#.txPE5dGjnj

[25] B. N. Levine, M. K. Reiter, C. Wang, and M. Wright, "Timing Attacks in Low-Latency Mix Systems," in *SpringerLink*. Springer, Berlin, Heidelberg, Feb. 2004, pp. 251–265. [Online]. Available: https://link-springer-com/chapter/10.1007/978-3-540-27809-2_25