Making Digital Cameras Less Attractive Targets for Theft

Henry Gordon Dietz and Oluwatofunmi Oyetan
Department of Electrical and Computer Engineering, University of Kentucky; Lexington, Kentucky

Abstract

Cameras are easy targets for theft. They are expensive, small, usually carried in the open, and not easily identifiable when stolen. Unlike cell phones, cameras typically do not have passwords or other login procedures, so the full functionality is generally available to anyone with physical access to the camera, and stolen cameras behave indistinguishably from ones operated by their legitimate owners. The current work examines various methods for making cameras less attractive targets for theft without significantly increasing either camera cost or the complexity of the user interface and interactions. Many of the new methods use various forms of anomalous behavior identification to enable the camera to passively recognize when it is likely that the person operating the camera is not the owner.

Introduction

Most computer-like devices use passwords or other login procedures to discourage theft. However, cameras typically use no such mechanisms, and introducing them would be viewed by many as a significant inconvenience and delay that causes unplanned photo opportunities to be missed.

Existing Approaches

Rather than focusing on discouraging a thief from stealing a camera, emphasis has been placed on tracking and recovery. For example, most cameras embed their serial number in each image captured, so scanning the WWW for posted images with the same serial number as an image that you provide can help locate your stolen camera[1]. Unfortunately, serial number tracking is most likely to find the (potentially innocent) person who purchased a camera that happened to be stolen than it is likely to find the thief. Another approach has been to mark expensive camera equipment with active trackers like Apple AirTag[2], and Nine Volt AirCap[3] allows you to hide an AirTag inside a camera body cap. These offer good potential to immediately locate a stolen camera, but it is easy for the thief to defeat trackers by either shielding them or removing them. In summary, existing methods focus on recovery of stolen cameras and often use systems that are not part of the camera hardware, i.e., Internet photo sharing or addition of tracking hardware.

Goals

In contrast, the current work seeks not just to aid in recovery, but also to make the camera itself less useful to the thief and more obviously stolen property. Making the camera aware that it (probably) has been stolen also allows it to secure user data, potentially including already captured images. The current work emphasizes methods that can be implemented using existing camera hardware, and includes proof-of-concept prototype implementations of some techniques using actual consumer cameras: Canon PowerShots reprogrammed via CHDK[4].

The current work can thus be seen as a proposal, suggesting to camera manufacturers that cameras can be made less attractive targets for theft without adding significant expense to their products nor requiring annoying login procedures. The new approach can be summarized as being guided by three goals:

1. **Do not interfere with normal user operation.** The problem with passwords and other login procedures is that they require a user action every time the device is to be used. Even fingerprint scanners require explicitly scanning your finger; any such action delays the time before the camera may be used. Although modern digital cameras have relatively fast start-up times, DPReview[5] used to include a table of timings for basic operations in every camera review. The time for a camera to go from power off to being ready to capture a photo was a key performance metric, with fast cameras around 3 seconds and disturbingly slow ones around 5 seconds. In sum, adding even just a couple of seconds for a login procedure would clearly be unacceptable to many photographers. The current work suggests the solution lies in passively detecting that the camera is being used by someone other than its owner, and the method proposed is a form of **Anomalous Behavior Detection** (**ABD**) detailed in the following section. Further, adding any hardware, such as a fingerprint scanner, to a camera is inherently likely to interfere with normal operation by requiring significant redesign and different ergonomics – all the methods discussed in the current work assume no significant hardware changes will be made to the camera.
2. **Render the stolen equipment worthless to the thief.** There is little motivation for theft where there is no profit to be made from equipment use or sale, nor access to personal information about the owner.
3. **Aid in recovery of the stolen camera.** This has been the primary goal in most prior work, but most approaches catch the potentially naive and innocent unauthorized user rather than the thief. For example, identifying a camera as stolen by recognizing the serial number in a photo posted online would only happen some time after the person who purchased the camera posted an image; that person would not have had any hint the camera was stolen at the time they purchased it at a flea market. It is preferable that the camera flag itself as stolen as early as possible so that a potential buyer could be aware before they make their purchase – or at least before the seller has had time to disappear.

Anomalous Behavior Detection

Although not currently used within cameras for theft prevention, the concept of ABD is well known in diverse fields. The author has used it for over a decade to automatically detect nodes that have suffered, or probably will soon suffer, a hardware failure or security breach within a machine room full of cluster supercomputers[6]. ABD also is often applied to detect inappropriate uses of social networking platforms[7]. In each application of ABD, the precise algorithms and data used vary, but the following steps are always present:

- Identify a fairly large number of potentially significant properties that are easily monitored and can be concisely summarized over time. For example, in monitoring health of a cluster computer node, things like processor temperature, processor load average, and volume of network traffic are all easily sampled time-varying properties that give potentially useful insights into node activity. There may be dozens or even hundreds of properties worth tracking.
- Collect time sequences of property data and recognize patterns and correlations in the time sequences. For example, processor temperature should roughly track load average, because simultaneous execution of more programs tends to generate more heat. There are many different algorithms that can be used for detecting these patterns and correlations, from simple hashing of property n-gram tuples to hidden Markov models and neural network classifiers. The goal is to construct a statistical model of what constitutes normal behavior.
- In real time, compare the property data streams to the patterns and correlations seen before, and flag data that is statistically inconsistent with normal behavior of the system. For example, if processor temperature is climbing when load average is not increasing, there is a high probability that a cooling problem is developing – perhaps airflow over a heat sink has become partially blocked by dust. Most often, the statistical model is not identifying anomalous behavior based on a single pattern being unmatched, but on multiple aspects of current system properties being unlikely based on observed past performance.

In the above description, the example properties were ones used to monitor health of cluster computer nodes. Here, the statistical model is intended not to monitor health of the camera, but to recognize when the user of the camera is not behaving like the camera's owner(s). Thus, the question becomes what potentially user-identifying properties can a camera easily access?

Half a century ago, film cameras generally provided very few controls or sensors: primarily shutter speed, aperture, and metered light level. More importantly, none of those values was recorded in an easily-recoverable way. With the advent of autofocus SLRs, the number of controls immediately expanded to include methods to direct the autofocus, but autofocus also meant that cameras began including microprocessors, and that made it practical to add far more controls and options than were available in the mechanical SLRs that came before them. When cameras moved from film to electronic sensors, the processing power inside cameras increased dramatically, as did the ability to digitally process and store a wide variety of user-interface interactions and image metadata.

As CHDK reveals, even low-end digital cameras now contain multi-core 32-bit processors and a surprising array of controls and sensors – all of which can be used to help ABD reliably detect when the operator of the camera is not the regular user. The types of data available fall into four categories: metadata, computed properties of captured images, event sequences, and directly sensed user behaviors.

User-Identifying Image Metadata

The most obvious source of potentially user-identifying information is the metadata recorded in the same file with each digitized image. Although some metadata fields are proprietary, ExifTool[8] can decode most metadata fields across over a hundred different image file formats. A typical JPEG image captured by a high-end camera has between 200 and 400 metadata fields decodable using ExifTool. Many of these fields describe fixed properties of the image or the camera, such as the pitch of the pixels on the sensor. However, many other fields describe properties that depend at least in part on the user of the camera and choices they made in operating it. Table 1 lists some of these types of metadata which are likely to be useful for ABD and are easily extracted from JPEG images created by recent Canon, Fujifilm, Nikon, Panasonic, and Sony cameras.

Many of these properties are really describing the basic exposure parameters, and it initially might not be not clear how metadata describing settings like the aperture or shutter speed used would identify the user. However, different photographers see the world differently, and their use of the camera reflects their artistic vision. As an example, one of the authors nearly always shoots in aperture priority mode, often with the lens wide open, but almost never stopped down past $f/11$ in order to avoid softening detail with diffraction. Many photographers prefer other modes and produce very different aperture and shutter speed usage statistics under similar lighting conditions with the same lens. The same types of preferences bring very different usage profiles for different photographers in attributes ranging from obvious things like which lens is used, focus mode selection, how flash is used, and picture style choices to relatively obscure decisions like how low one lets battery charge drop before changing batteries. Do you shoot lots of photos at particular times during the day, such as around sunrise or sunset? The orientation metadata simply indicates how the image must be rotated and/or mirrored in order to be in the correct orientation, and most photographers capture most images in the 0° horizontal orientation, but how often is the camera turned to capture a vertical image – and do you turn it 90° or 270° (clockwise or counter-clockwise)?

Some of the image metadata provides insight into what kinds of scenes you like to photograph. For example, how many faces are detected how often will be very different for a landscape photographer vs. a mom who is usually photographing her kids. Positions of focus points and bounding-boxes for faces are also recorded, which makes it possible to know something about composition without further analysis of the captured image.

Table 1: Some Potentially User-Identifying Type of Image File Metadata

Data description	Canon	Fujifilm	Nikon	Panasonic	Sony
AF area and mode	✖	✖	✖	✖	✖
AF points	✖	✖	✖	✖	✖
Aperture	✖	✖	✖	✖	✖
Battery status	✖	–	–	–	✖
Brightness	–	✖	✖	–	✖
Color space options	✖	✖	✖	✖	✖
Compression	✖	✖	✖	✖	✖
Contrast	✖	✖	✖	✖	✖
Digital zoom	✖	–	–	✖	✖
Distortion correction	✖	✖	✖	✖	✖
Dynamic range settings	✖	✖	✖	✖	✖
Exposure mode and parameters	✖	✖	✖	✖	✖
Exposure compensation	✖	✖	✖	✖	✖
Faces detected	✖	✖	✖	✖	✖
Flash configuration	✖	✖	✖	✖	✖
Lens data	✖	✖	✖	✖	✖
Focus mode	✖	✖	✖	✖	✖
HDR data	✖	✖	✖	✖	✖
Noise reduction settings	✖	✖	✖	✖	✖
Hyperfocal distance	✖	✖	✖	✖	✖
Image dimensions	✖	✖	✖	✖	✖
Image stabilization	✖	✖	–	✖	✖
IMU/accelerometer data	–	–	–	✖	–
Lens ID and specifications	✖	✖	✖	✖	✖
Lens serial number	✖	✖	✖	✖	–
Macro mode	✖	–	–	✖	–
Metering mode	✖	✖	✖	✖	✖
Multiple exposure data	✖	–	–	✖	✖
Orientation	✖	✖	✖	✖	✖
Picture mode or style	✖	✖	✖	✖	✖
Power up time	–	–	✖	–	–
Quality setting	✖	✖	✖	✖	✖
Rating	✖	✖	✖	✖	✖
Raw file type	–	–	–	–	✖
Red eye repair	✖	–	–	✖	–
Saturation	✖	✖	✖	✖	✖
Scale factor to 35mm	✖	✖	✖	✖	✖
Scene mode or type	✖	✖	✖	✖	✖
Sharpness	✖	✖	✖	✖	✖
Temperature	✖	–	–	–	✖
Time zone or city	✖	–	✖	✖	–
Timestamps	✖	✖	✖	✖	✖
User comment	✖	✖	✖	–	✖
Vignetting correction	✖	–	✖	–	✖
White balance	✖	✖	✖	✖	✖

User-Identifying Image Analysis

It is also possible to directly analyze each captured image. AI methods for creating textual descriptions of scenes are becoming very effective, and that would be a great way to recognize user preferences such as mostly shooting photos with cats in them. However, at this time, the computational facilities in most cameras are not sufficient to embed this level of analysis in-camera without a notable drop in performance or battery life. Typical cameras have a small number of ARM processor cores with dedicated hardware support for operations like JPEG encoding and neural networks used for intelligent autofocus, but support for more general-purpose high-performance computing is minimal.

User-Identifying Event Sequences

Any user-interface interaction internally inserts an event record into a queue from which the camera's processors can easily obtain both sequence and timing. Events range from pressing a button or turning a dial to more subtle actions such as bringing the viewfinder to your eye (many cameras have sensors intended to automatically switch between powering the viewfinder and rear LCD panel). In the early days of CHDK development, the primary way CHDK allowed scripts to control camera functions was by inserting fake "logical event" records into the event queue, so the event queue was one of the first internal data structures to be understood. It is easy for camera firmware to record event sequences and, as CHDK's Lua scripting interface reveals, there are a multitude of different types of events. Every button has both `Press` and `Unpress` events, and some external actions like `InsertMedia` and `ConnectUSBCable` also produce event records.

How should event sequences be recorded? A simple method that is likely to provide very useful data would be to record either n-gram statistics or Markov chain state transition probabilities. Each type of event is internally assigned a number. A simple n-gram hash table could be created using the types of the last n events as the key. For example, each time the 3-gram {`PressPBButton`, `UnpressPBButton`, `PressFuncButton`} occurs, it could be looked-up in a 3D hash table and the corresponding occurrence count incremented. Although n-gram statistics for large n can be complex to manage, this type of bookkeeping is computationally very efficient for 1-grams, 2-grams, and 3-grams. These occurrence counts are trivially convertible into conditional probabilities. For example, the probability of `PressFuncButton` is the 1-gram count for `PressFuncButton` divided by the total number of 1-grams recorded. Similarly, the conditional probability of `UnpressPBButton` being followed by `PressFuncButton` is the count of the 2-gram {`UnpressPBButton`, `PressFuncButton`} divided by the 1-gram count of `UnpressPBButton`. The camera's processor easily can compute these probabilities in real time, and ABD could be as simple as flagging when multiple low probability sequences occur over a short interval.

User-Identifying Sensed Behaviors

In addition to user actions that are internally processed as events, there are a variety of sensors that can be used to determine how the camera is being handled by the user.

Any information the camera writes into an image file is information that must be available to the microprocessor within the camera – and there is usually additional information known but not written into image metadata. Much of this data can be observed by the microprocessor continuously even when an image is not being captured. For example, nearly all cameras contain an orientation sensor so the camera can know that a photo was captured in the 0° horizontal orientation (as the file metadata tag would record). CHDK does not reveal any events being recorded for orientation changes, but the orientation sensor can be polled quite quickly. Thus, the camera could know that 3.4 seconds before that image was captured horizontally the photographer tried composing the image in the 90° vertical orientation for 2.1 seconds. Combining this type of sensor data with event statistics might significantly increase the reliability of ABD.

Another example of pollable sensor data would be measurements of camera shake. In many cameras, there are high-performance **IMUs** (**inertial measurement units**) often used to drive **IBIS** (**in-body image stabilization**) that avoids blurring the image by moving the sensor within the camera to compensate for camera movement. Reading these IMUs can build-up a model of how the camera shakes when held by its owner. In earlier work, we have shown that simple analysis of a live view stream using CHDK can provide similar shake measurements even without an IMU[9]. In another work[10], we showed that camera shake characteristics depend significantly on things like how the camera is held. For example, when using the rear LCD for framing, gripping a camera with both hands might reduce total movement, but tends to cause more roll than if the camera is held in one hand. The result is that simple statistical features of the shake profile can help identify the user's preferred grip and therefore the user.

Response To Potentially Being Stolen

Given that ABD mechanisms can allow the camera to detect when it is likely to have been stolen, the question becomes what to do in response to that.

Most cameras contain speakers and focus assist, flash, or other lights that can be lit. Thus, one possibility is for the camera to use those mechanisms to attract as much attention as possible to expose the thief. However, it is difficult to recommend that if the false-positive rate for detecting theft is even slightly above zero.

A more subtle, but still very effective, response would be to display some kind of theft notice on the camera's rear LCD and have the camera simply refuse to function until something akin to a password has been entered. To be precise, this is what we would recommend:

- Display a message that provides a method to return the camera to its owner. This need not be traditional contact information, such as name, address, or phone number, but could instead be a link to a unique WWW address that allows contacting the owner without exposing any of the owner's personal information. Such a WWW address might be provided as free service by the manufacturer in return for registering that you have purchased the camera; it could even be a fixed WWW address with the camera's serial number as a suffix or HTML `GET` argument. The WWW site managing these unique WWW addresses also could be used to confirm legal ownership when a camera is resold and to provide an owner who has forgotten their password with a secure method for factory reset of their camera.
- Keep the message on the screen until a password or combination-lock-like sequence of camera button presses has confirmed the identity of the user. Until the correct key has been entered, all other camera functions should be completely disabled: no shooting, playback, remote control (including tethering via wired or wireless connections), nor use of the camera as a USB mass storage interface to access the memory card(s). Power cycling the camera after

Table 2: Some Camera-Identifying Image File Metadata

Data description	Canon	Fujifilm	Nikon	Panasonic	Sony
Artist	✖	✖	✖	✖	–
Camera serial number	✖	✖	✖	✖	✖
Copyright	✖	✖	✖	✖	–
Lens ID and specifications	✖	✖	✖	✖	✖
Lens serial number	✖	✖	✖	✖	–

an ABD check has triggered the disabling of other functions should return the camera to the same locked state.

Since even a small interruption to camera function might be problematic under certain circumstances, such as shooting a wedding, we would suggest that the user be given the option of temporarily disabling the ABD checking. However, the default should be that ABD checking is enabled, and if it is temporarily disabled, when the camera detects a reasonably long temporal gap in the camera's usage (e.g., several hours), ABD checking should be automatically re-enabled.

In order to prevent a thief from obtaining personal data by removing the memory card(s) and using a separate card reader, we would further suggest that the camera should provide the user with the configuration option that all image and personal data written to the card by the camera would be encrypted as it is written to the card. The decryption key can be applied by the camera automatically in normal operation, so that camera playback of captured images and even use of the camera as a USB mass storage interface transparently decrypts. Processing associated with reasonably secure data encryption can be a significant overhead, but cameras supporting WiFi generally already have hardware to accelerate encryption and decryption. It additionally would be possible to make encrypted images unmodifiable by the user, which could be a valuable feature for reporters and law enforcement using cameras to collect legal evidence.

Recovery of Stolen Cameras

Most prior work involves not prevention of theft, but aiding in the recovery of stolen cameras. These approaches can be broadly divided into passive and active methods.

Passive Location of Stolen Cameras

A very passive approach to aiding in locating a stolen camera involves searching photos posted on the Internet for images tagged as coming from the stolen camera. Where Table 1 listed types of metadata can help identify the user of a camera, Table 2 lists some metadata fields that uniquely identify a particular camera. For example, stolencamerafinder[1] states that it searches for image files with the same camera serial number as an image you provide. It is important to note that serial numbers can be stored in various fields and, for example, stolen camera finder does not recognize the serial numbers within images captured by most Sony cameras. Locating an image with a matching camera serial number online does not necessarily provide a way to trace the image back to the poster's real-world location and identity. There is also the issue that the bulk of images found to embed the same camera serial number might often be images that were posted by the legitimate owner. In fact, there is no guarantee that the thief would ever post an image from a stolen camera, and it is most likely that images are posted by the potentially innocent person who purchased the camera from the thief at a flea market or similar venue.

A better approach would be to search for images from the same camera using a variety of forensic markers rather than just the camera serial number. It is easy for an internet image search to compare any set of camera-identifying metadata fields (i.e., from Table 2) rather than just the camera serial number. However, there are many free software tools that can strip all potentially identifying metadata from an image for posting. Using properties that are computable from the image itself would continue to work even if most metadata has been removed.

One example of this is using the sensor noise pattern to uniquely identify the camera used. These stochastic noise patterns have been shown to allow distinguishing between cameras of the same model even after JPEG compression of the raw sensor data[11]. Alone, such matching might be unreliable in distinguishing between all digital cameras ever made, but it should be very effective when constrained to matching among only images that for other reasons are suspected to come from the same camera. For example, even when a software tool has been used to strip identifying metadata, the basic JPEG compression parameters involve a variety of free choices that can be used to determine the set of camera models that could possibly have captured the image[12][13].

Another possibility would be combining camera model recognition with recognition of unique characteristics of an individual lens by analysis of the out-of-focus point spread function (OOF PSF), as was introduced in earlier work by Dietz[14]. The OOF PSF can be extracted from any image containing defocused point light sources that are not completely saturated, and the interference patterns caused by dust particles within a lens essentially form a fingerprint that can reliably distinguish between otherwise identical lenses. Figure 1 clearly shows that the OOF PSF measured using the same Sony NEX-7 body with two apparently clean and identical Minolta MD 50mm $f/1.7$ easily distinguishes the two lenses. Although the internal dust spots in each are quite tiny, each speck causes diffraction that makes it appear surprisingly clearly as a small Airy disc within the OOF PSF. This pattern is invariant with amount of defocus and scaling of the images, is turned inside-out (effectively rotated 180°) by changing from focusing before to focusing after the point depth in the scene, and is clipped by stopping-down the lens aperture or sampling off-center – all transformations easily accounted for in matching.

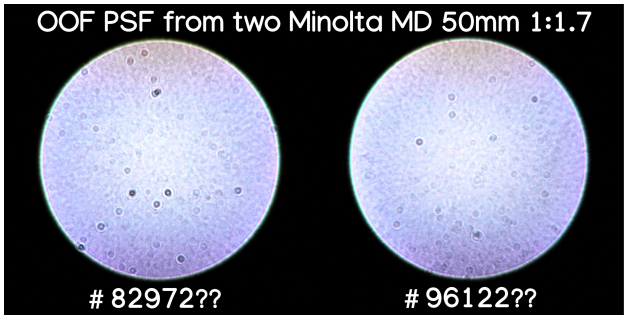

Figure 1. *Measured OOF PSF of two identical MD 50mm $f/1.7$ lenses*

Figure 2. *CHDK feasibility test for a few measures*

Active Location of Stolen Cameras

Whereas passive approaches depend on the user of a stolen camera taking some action that makes images captured by the camera visible, active mechanisms do not: the camera itself takes the action.

Perhaps the most obvious such approach is to mark expensive camera equipment with trackers like Apple AirTag[2], SmartTag[15], Tile Pro[16], Chipolo[17], or Cube Shadow[18]. These trackers work by sending their ID over local networks, usually via BlueTooth. BlueTooth itself is limited to communication within tens of meters, but by creating an ad-hoc network using other BlueTooth devices that are in range, tags can be tracked anywhere that they are within range of a device on the tracking network. The tracking network created by Apple devices is called *Find My*, but there are numerous similar alternatives. Common to all of them is that the tag is a separate device, which would be quite visible and easily removed, or shielded to prevent transmission, by a thief. Nine Volt AirCap[3] allows hiding an AirTag inside a camera body cap, but many photographers never use a body cap, instead keeping a lens mounted.

The main advantage in use of these trackers is that immediately after your camera has been stolen, before the thief has disabled the tracker, these tags can quickly locate and track your equipment in the real world. This increases the probability that the thief will be caught before transferring your equipment to someone else.

The other main type of active location of stolen cameras involves using the facilities built-into the camera. Most high-end cameras now have 802.11 WiFi and BlueTooth networking support built-in, but those networking facilities draw enough power so that they are usually enabled only when the camera is explicitly turned on. Thus, a camera stolen in the "off" state would not be able to emulate the separate tags; it would certainly be possible to build-in a low-power tracker, but that hardware does not seem to be present in current cameras. However, when the thief or someone they sell the stolen equipment to turns power on, it is certainly feasible for the camera to use WiFi or BlueTooth to "call home" and report its location using whatever networks it can reach. The concept of regularly "calling home" is widely used in IoT devices, but often is criticized for leaking information as well as excess power use. We would suggest that it would be relatively simple for a camera to use the ABD mechanisms described earlier to "call home" only when the camera has indications that it might have been stolen.

Perhaps one of the most interesting attributes of camera network interfaces is that networked cameras are rarely intended to stand alone: typically, they are bonded to the owner's cell phone. Cell phones are commonly used for archiving or posting camera images, remote control of camera exposure, etc. This implies that powering up and not seeing the owner's cell phone, which would normally be bonded to it via a wireless connection, might in itself be a strong indication that the camera has fallen into the hands of a thief and should call home.

Conclusion

The current work constitutes a very preliminary investigation of the feasibility of using various methods to make digital cameras less attractive targets for theft – a goal that previously has not been given much attention. There are three key concepts:

- Do not interfere with normal user operation of the camera, primarily by passively using anomalous behavior detection (ABD) to determine when the operator is probably not one of the usual users of the camera
- Render the stolen equipment worthless to the thief by effectively bricking the camera and refusing to decrypt any personal data it contains until ownership has been confirmed
- Aid in recovery of the stolen device, preferably recovering it from the thief rather than from a later user of the camera who might not even know it was stolen

The current work has provided a list of metrics and discussed various methods which meet our goals and do not require additional hardware. The feasibility of implementing many of these have been confirmed by prototype implementations using Canon PowerShot ELPH180 and SX530 HS cameras reprogrammed using CHDK[4]. For example, figure 2 shows a CHDK Lua script running on a Canon PowerShot SX530 HS to monitor lens, CCD (actually a CMOS sensor in this camera), and battery temperatures, battery voltage, current orientation and percentage time spent in each orientation, and even the level of camera shake detected from the live view stream. The overhead for such measurements was not prohibitive even using Lua scripts which suffer significantly more overhead than compiled C code would for the same operations. However, to be effective in discouraging theft, camera firmware needs to directly incorporate these methods. It is our hope that the current work will inspire various manufacturers to at least consider implementing these types of anti-theft mechanisms.

References

[1] https://www.stolencamerafinder.com/ (accessed 1/25/2023)

[2] T. Roth, F. Freyer, M. Hollick and J. Classen, "AirTag of the Clones: Shenanigans with Liberated Item Finders," 2022 IEEE Security and Privacy Workshops (SPW), San Francisco, CA, USA, 2022, pp. 301-311, DOI: 10.1109/SPW54247.2022.9833881

[3] https://nine-volt.com/products/nine-volt-aircap-camera-body-cap-for-apple-airtag (accessed 1/25/2023)

[4] "*Canon Hack Development Kit (CHDK)*," http://chdk.wikia.com/wiki/CHDK (accessed 1/8/2023)

[5] https://www.dpreview.com/ (accessed 1/25/2023)

[6] James Frank Roberts, *Automatic Detection of Abnormal Behavior in Computing Systems*, University of Kentucky Theses and Dissertations – Computer Science, 2013, https://uknowledge.uky.edu/cs_etds/11

[7] V. Chauhan et al., "Anomalous behavior detection in social networking," 2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Delhi, India, 2017, pp. 1-5, DOI: 10.1109/ICCCNT.2017.8204141

[8] Phil Harvey, *ExifTool*, https://exiftool.org/ (accessed 1/28/2023)

[9] Henry Dietz, "Camera support for use of unchipped manual lenses," Proc. IS&T Int'l. Symp. on Electronic Imaging: Imaging Sensors and Systems, 2020, pp 327-1 – 327-7, DOI: 10.2352/ISSN.2470-1173.2020.7.ISS-327

[10] Henry Dietz, William Davis, and Paul Eberhart, "Characterization of camera shake," Proc. IS&T Int'l. Symp. on Electronic Imaging: Imaging Sensors and Systems, 2020, pp 228-1 – 228-7, DOI: 10.2352/ISSN.2470-1173.2020.7.ISS-228

[11] Jan Lukas, Jessica Fridrich, and Miroslav Goljan. "Digital Camera Identification From Sensor Pattern Noise," IEEE Transactions on Information Forensics and Security, 2006, pp 205 – 214, DOI: 10.1109/TIFS.2006.873602

[12] K. S. Choi, E. Y. Lam and K. K. Y. Wong, "Source Camera Identification by JPEG Compression Statistics for Image Forensics," TENCON 2006 - 2006 IEEE Region 10 Conference, Hong Kong, China, 2006, pp. 1 – 4, DOI: 10.1109/TENCON.2006.343943

[13] S. Mandelli, N. Bonettini, P. Bestagini, "Source Camera Model Identification," In Sencar, Verdoliva, and Memon (eds) *Multimedia Forensics. Advances in Computer Vision and Pattern Recognition*, Springer, Singapore, 2022 DOI: 10.1007/978-981-16-7621-5_7

[14] Henry Dietz, "Out-of-focus point spread functions," Proceedings Volume 9023, Digital Photography X; 90230J (2014) IS&T/SPIE Electronic Imaging, 2014, San Francisco, California, United States DOI: 10.1117/12.2040490

[15] T. J. Fink, "Samsung Galaxy SmartTag review: A stellar smart key finder for Samsung phones," June 17, 2021 https://www.tomsguide.com/reviews/samsung-galaxy-smarttag

[16] Kyle VanHemert, "Tile Might Be a Revolutionary Gizmo For Finding Lost Keys and Stolen Purses," August 1, 2013 https://www.wired.com/2013/08/tile-a-better-way-to-find-your-lost-keys-and-maybe-your-stolen-bike-too/

[17] Philip Michaels, "Chipolo One key finder review," October 6, 2020 https://www.tomsguide.com/reviews/chipolo-one-key-finder

[18] "Cube Shadow," https://cubetracker.com/products/cube-shadow (accessed 2/4/2023)