Are There Wireless Hidden Cameras Spying on Me?

Jeongyoon Heo
Samsung Research
Seoul, Republic of Korea
jr.heo@samsung.com

Sangwon Gil
Samsung Research
Seoul, Republic of Korea
sangwon.gil@samsung.com

Youngman Jung
Samsung Research
Seoul, Republic of Korea
yman.jung@samsung.com

Jinmok Kim
Samsung Research
Seoul, Republic of Korea
jinmok.kim@samsung.com

Donguk Kim
Samsung Research
Seoul, Republic of Korea
donguk14.kim@samsung.com

Woojin Park
Samsung Research
Seoul, Republic of Korea
woojin1.park@samsung.com

Yongdae Kim
KAIST
Daejeon, Republic of Korea
yongdaek@kaist.ac.kr

Kang G. Shin
The University of Michigan
Ann Arbor, United States
kgshin@umich.edu

Choong-Hoon Lee
Samsung Research
Seoul, Republic of Korea
choonghoon.lee@samsung.com

ABSTRACT

The proliferation of IoT devices has created risks of their abuse for unauthorized sensing/monitoring of our daily activities. Especially, the leakage of images taken by wireless spy cameras in sensitive spaces, such as hotel rooms, Airbnb rentals, public restrooms, and shower rooms, has become a serious privacy concern/threat. To mitigate/address this pressing concern, we propose a Spy Camera Finder (SCamF) that uses ubiquitous smartphones to detect and locate wireless spy cameras by analyzing encrypted Wi-Fi network traffic. Not only by characterizing the network traffic patterns of wireless cameras but also by reconstructing encoded video frame sizes from encrypted traffic, SCamF effectively determines the existence of wireless cameras on the Wi-Fi networks, and accurately verifies whether the thus-detected cameras are indeed recording users' activities. SCamF also accurately locates spy cameras by analyzing reconstructed video frame sizes. We have implemented SCamF on Android smartphones and evaluated its performance on a real testbed across 20 types of wireless cameras. Our experimental results show SCamF to: (1) classify wireless cameras with an accuracy of 0.98; (2) detect spy cameras among the classified wireless cameras with a true positive rate (TPR) of 0.97; (3) incur low false positive rates (FPRs) of 0 and 0.031 for non-camera devices and cameras not recording the users' activities, respectively; (4) locate spy cameras with centimeter-level distance errors.

CCS CONCEPTS

• **Networks → Network privacy and anonymity**; • **Security and privacy → Mobile and wireless security**.

KEYWORDS

Encrypted Traffic Analysis, Hidden Camera, Smartphone

ACM Reference Format:
Jeongyoon Heo, Sangwon Gil, Youngman Jung, Jinmok Kim, Donguk Kim, Woojin Park, Yongdae Kim, Kang G. Shin, and Choong-Hoon Lee. 2022. Are There Wireless Hidden Cameras Spying on Me?. In *Annual Computer Security Applications Conference (ACSAC), December 5–9, 2022, Austin, TX, USA.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3564625.3564632

1 INTRODUCTION

In recent years, camera-enabled devices (surveillance camera, baby monitor, IP camera, etc.) have been widely deployed to facilitate a variety of protection functions ranging from personal security to public safety. For user convenience, most of these devices provide capabilities of streaming live videos from the camera to the cloud via wireless networks. On the other hand, there have been increasing incidents of streaming live videos of individuals' activities in sensitive spaces like living rooms and hotels via wireless spy cameras [6, 7, 39, 43] that are easy to deploy (without requiring additional wiring) and provide Internet connection through pervasively available Wi-Fi access points (APs). These have created a serious privacy threat and become a social problem. Our goal is to detect a "spy camera" which (1) is Wi-Fi-based and placed in the same space as victims who do not want to be spied on, and (2) continuously records and streams victims' activities.

Most commercialized spy camera detectors [34, 36, 38] such as RF signal detectors and camera lens detectors are not easy to use and suffer from poor detection accuracy [5]. Even worse, users should carry separate dedicated-purpose devices. In order to overcome these shortcomings, academic researchers have proposed the use of smartphones to detect wireless spy cameras. One of the promising approaches is to analyze Wi-Fi network traffic to detect spy cameras recording and streaming users' activities over Wi-Fi networks in real time [4, 5, 18, 21, 37]. They use the characteristics of camera traffic patterns and video encoding, where the data rate of encoded video frames depends on the changes in a recorded video scene. A user's movement in front of a camera causes an increase in the encoded video frame sizes, and prior work exploits this phenomenal feature. Due to the network encryption, most of the prior work uses only limited information in network packet levels such as packet length, the number of packets, and data rate, to infer the presence of a wireless camera streaming the user's activity. However, in a packet flow generated by a wireless camera, there are not only

ACSAC, December 5–9, 2022, Austin, TX, USA
© 2022 Association for Computing Machinery.
ACM ISBN 978-1-4503-9759-9/22/12…$15.00
https://doi.org/10.1145/3564625.3564632

video packets but also other types of (such as control and audio) packets. Sometimes, network congestion causes packet loss. These conditions make the precise identification of wireless camera traffic harder. Therefore, most existing spy camera detection schemes [4, 5, 21] using a network packet level analysis suffer from accuracy degradation depending on the network condition and the type of camera, and also cannot localize spy cameras which requires accurate detection of changes of video frame size according to the user's movement.

To overcome prior work's limitations, we propose a fine-grained analysis of encrypted traffic, SCamF, which uses both network packet and video frame level information inferred from the encrypted Wi-Fi traffic. By using the nature of video encoding, SCamF reconstructs and extracts the video frame sizes from the encrypted Wi-Fi traffic being generated and transmitted by wireless spy cameras. With this information, SCamF detects the presence of a wireless camera's traffic and whether or not that traffic is transmitted by a spy camera, and also determines the location of the spy camera more accurately than other approaches that use network packet level information only.

This paper makes the following main contributions:

- We propose a fine-grained encrypted traffic analysis approach, SCamF, to extract encoded video frame-size information from encrypted Wi-Fi traffic by utilizing the characteristics of wireless camera traffic patterns and Wi-Fi packet transmission patterns. SCamF uses only limited information available in the Wi-Fi MAC header.
- SCamF determines both the presence and the location of spy cameras by analyzing video frame-size changes in accordance with a user's position and movement. It achieves high accuracy with low false positives by analyzing the correlation between video frame-size changes and smartphone sensor values which represent the smartphone owner's movement.
- We design and implement SCamF on commodity Android smartphones without requiring any dedicated device. By enabling the Wi-Fi monitor mode with custom firmware, SCamF does not have to be connected to the same Wi-Fi AP as spy cameras. In addition, we demonstrate SCamF's robustness through extensive experiments on a real testbed across 20 types of wireless cameras.

The remainder of this paper is structured as follows. §2 discusses related work while §3 reviews our system and adversary model. In §4, we describe how SCamF detects and localizes wireless spy cameras. In §5, we describe our testbed and experimentally evaluate the performance of SCamF for various types of wireless cameras. Finally, we discuss deployment considerations and limitations in §6, and conclude the paper in §7.

2 RELATED WORK

Most existing spy camera detection products and solutions try to detect intrinsic hardware features, such as lens' light reflection [10], electromagnetic wave [15], and RF signals (2.4/5 GHz radio frequency) [36] emitted by wireless cameras. These approaches have been widely used in specific products [34, 38] and commercialized in applications [10, 15] on Android/iOS platforms. However, they all suffer two practical limitations: i) poor detection accuracy due to the interference from nearby electronic devices; ii) poor usability due to the requirement of significant user's effort during detection, and poor information embedded in detection results.

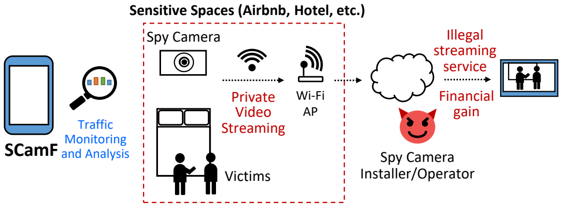

Sensitive Spaces (Airbnb, Hotel, etc.)
Spy Camera
Private Video Streaming
Wi-Fi AP
Victims
SCamF
Traffic Monitoring and Analysis
Illegal streaming service
Financial gain
Spy Camera Installer/Operator

Figure 1: System Model.

To address these limitations, researchers proposed numerous ways to detect spy cameras and also identify their location.

Spy camera detection: DeWiCam [4, 5] and Blink [21] statistically analyze network characteristics, such as traffic volume and packet length distribution, to detect spy camera traffic on wireless networks. In addition, for in-room camera inference which determines if the thus-found wireless camera is monitoring the room, [4, 5, 21, 37] observe the changes in a packet flow according to the user's movement or changes in the lighting condition. Although these techniques work well to detect the wireless camera traffic in a limited environment, they do not generalize to a variety of wireless cameras and practical environments in private spaces because they only utilize indirect information from a network packet level analysis. Furthermore, most of them do not provide any mechanism to identify the exact locations of spy cameras.

Spy camera localization: LAPD [32] localizes a spy camera by utilizing ToF sensors that emit laser signals in a smartphone. LAPD detects unique reflections from the camera lens to localize a spy camera. However, it is inconvenient to use LAPD. In particular, users should scan only one suspicious object at a time at an ideal distance to find a spy camera. SNOOPDOG [37] detects and localizes Wi-Fi-based sensors, like wireless cameras, monitoring a user's activity. It determines the location of a wireless camera by showing a laptop screen whose color continues to change in various locations and directions. It gradually eliminates a fraction of a space where no bitrate changes are detected. This trial-based approach takes a long time to localize a camera since it needs 30 seconds for every different location and direction, and requires the user to try many times. MotionCompass [18] localizes a wireless camera with a motion sensor by observing the network traffic. A wireless camera equipped with a motion sensor generates a large amount of traffic when it detects motion. It determines when a user enters or exits a camera's field of view by detecting the duration of high traffic generation to localize the camera. However, MotionCompass only deals with cameras that provide motion detection and cannot be used if a camera covers the entire detection area, which is common in small spaces. It does not handle a time delay, either. The distance error increases when the user's movement speed increases. It also requires users to follow a precise route, which is inconvenient and difficult for the users to do.

3 SYSTEM AND ADVERSARY MODEL

3.1 System Model

In the target scenario of wireless spy camera detection and localization, there are four major entities as shown in Fig. 1: spy camera(s), an installer/operator of spy camera(s), victims, and SCamF (Spy Camera Finder). The installer/operator installs one or more hidden cameras to spy on victims, collects and abuses the victims' video contents to get financial gain (e.g., online video streaming service) [6, 7, 39, 43]. We assume that spy cameras are Wi-Fi-based and

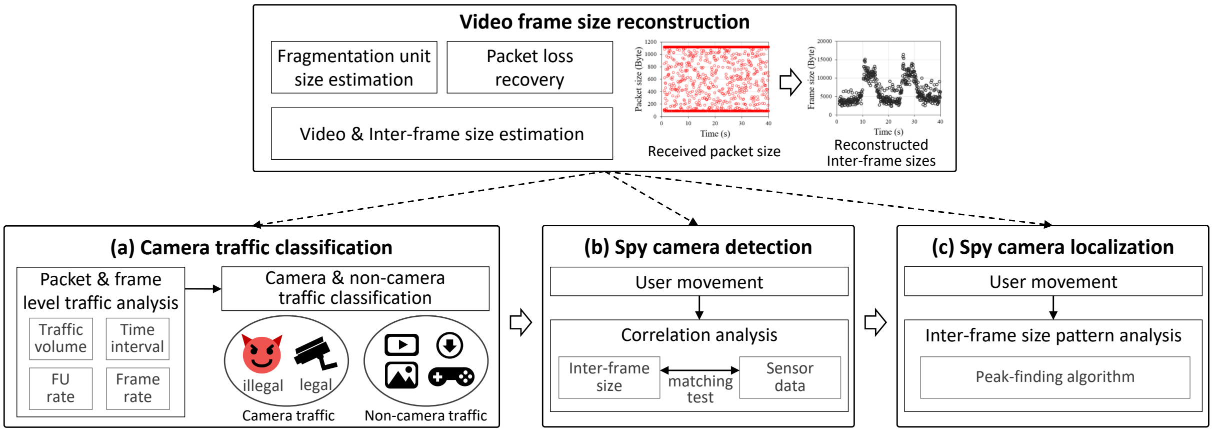


Figure 2: Overview of SCamF's operation and function.

hidden due to their easy installation (no wiring is needed) and easy Internet connection for collecting and monitoring victims' activities. The spy camera installer can connect hidden cameras to pervasively available Wi-Fi APs in sensitive spaces such as hotels, Airbnb with only a single visit. Victims are individuals visiting hotels or Airbnb for personal or business trips. A victim uses SCamF which collects Wi-Fi packets around him/her via the Wi-Fi monitor mode (Appendix §A) integrated with custom firmware on an Android smartphone that supports multiple bandwidths (20/40/80 MHz) for real-time packet capture across 2.4 & 5 GHz Wi-Fi channels. By using the monitor mode, SCamF can collect Wi-Fi traffic without connecting to the same Wi-Fi AP as the target spy cameras and determines whether suspicious camera traffic exists or not. The victim performs a simple motion to detect if the camera traffic is indeed spying on his/her activities and to localize the spy camera.

3.2 Adversary Model

SCamF is designed under the following three assumptions of adversaries.

A1. *Unrecognizable camera*: an adversary can hide a spy camera anywhere in the space of interest. Due to the advances in microelectronics technology, spy cameras have become small enough to be invisible to human eyes, where the diameter of the camera lens is less than 1 *mm* [7]. So, it is very difficult for ordinary users to find a spy camera [30].

A2. *Real-time streaming*: the target devices are wireless spy cameras which record and send/upload the victim's video in real time via a Wi-Fi network.

A3. *Standard Video Codec*: Wireless spy cameras commonly use standard video codecs such as H.264 [33][40] and H.265 [25] [23], and the codecs are usually integrated in a systems-on-chip (SoC) in the cameras [4]. The adversaries are assumed to use the integrated standard video codec without modifying it.

4 DESIGN OF SCAMF ALGORITHM

4.1 Design considerations

Any preventive solution must meet the following requirements and constraints.

R1. Access to any portion of network packet data aside from PHY/MAC headers is not guaranteed because of potential network encryption. Therefore, the information required to identify a wireless spy camera should be extracted solely from the Wi-Fi MAC headers.

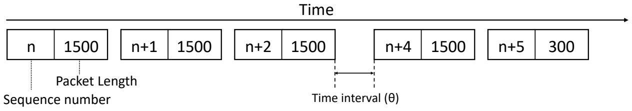


Figure 3: Reconstruction example from packet loss.

R2. Spy cameras are designed and manufactured by various vendors [8, 13]. From our experiments, we found different configurations and packet transmission patterns. Therefore, the solution should identify the common characteristics of wireless traffic from a variety of spy cameras.

R3. As the level of congestion on the target Wi-Fi network rises, packet loss increases [29], thus impeding the ability to analyze encrypted network traffic [9, 20, 22]. Therefore, the solution should be robust to loss of wireless packets.

R4. It should be easy to use. It should not require any special tools or prior actions, such as connecting to a Wi-Fi AP.

4.2 Workflow of SCamF

Fig. 2 illustrates SCamF's pipelined workflow. It operates in 3 steps: (a) Camera traffic classification (§4.4), (b) Spy camera detection (§4.5), and (c) Spy camera localization (§4.6). To achieve high accuracy in each step, SCamF reconstructs and utilizes the video frame sizes through network packet analysis from each encrypted Wi-Fi traffic as described in §4.3.

(a) Camera traffic classification: SCamF first classifies camera and non-camera traffic by using network packet level information such as traffic volume, inter-packet time interval, fragmentation unit (FU) rate, and frame per second (FPS).

(b) Spy camera detection: SCamF verifies if the thus-identified devices are indeed spy cameras recording the user's movements and live streaming the recorded videos by calculating the correlation between inter-frame size changes and the user movement. To reduce the false positive rate, SCamF utilizes smartphone sensors, such as gyroscope to track the user's movement pattern.

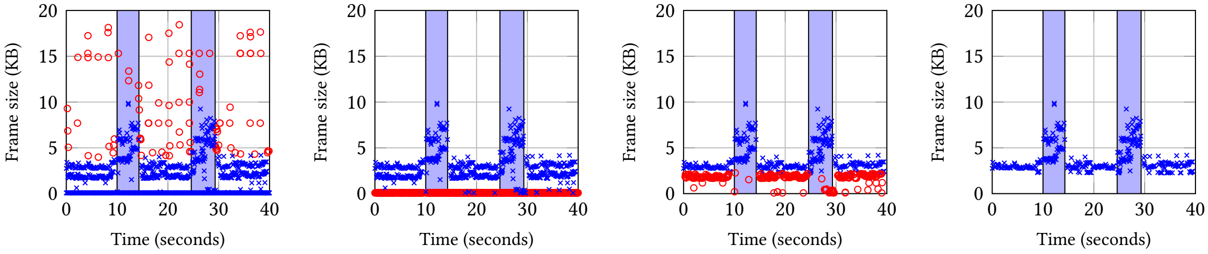
(c) Spy camera localization: SCamF localizes the detected spy cameras by observing the video frame size pattern according to the distance between a spy camera and a user.

4.3 Video frame size reconstruction

Due to the nature of video encoding and transmission, the wireless camera's traffic has unique characteristics. Most video compression

Table 1: Traffic features for classification between camera traffic and non-camera traffic.

Traffic type	Minimum traffic volume (kbps)			Minimum FPS			Average inter-packet time interval (msec)			FU rate		
	Min.	Max.	Avg.	Min.	Max.	Avg.	Min.	Max.	Avg.	Min.	Max.	Avg.
Camera	5.2	241.0	52.5	1	75.3	26.4	4.5	62.3	15.3	0.09	0.91	0.51
VOD	0	1585.8	175.4	0	2.33	0.2	0.3	41.2	2.1	0	1	0.95
Download	614.0	2043.9	1450.9	0	1	0.1	0.4	0.8	0.5	1	1	1
Picture	0	897.3	28.4	0	16.7	1.5	0.6	111.7	5.8	0	0.99	0.83
Game	0	4.0	1.3	0	12	4.3	8.1	73.0	50.2	0	0.62	0.05



(a) Removal of outliers

(b) Removal of noise frames

(c) Removal of remaining noise frames

(d) Final result

Figure 4: Inter-frame extraction procedures: ○ indicates data removed in each procedure.

standards, such as H.264 [33][40], and H.265 [25][23], use inter-frame prediction to achieve a high compression rate [31]. Inter-frames such as P-/B-frames encode the residuals after frame prediction using motion compensation referencing Intra-coded frame (I-frame). Inter-frame sizes increase according to the movement in a scene, which results in an increase of overall traffic. Most existing techniques for detecting spy cameras with network traffic analysis use network packet level information such as bitrates [4, 5, 21]. However, since a flow is mixed with video and other packets, and packet loss is caused by network congestion [26, 35, 44], it is difficult to accurately identify wireless camera transmission characteristics with raw packet information or basic statistics of packets.

For more accurate detection, we use not only network packet level information, but also information of the video frames being transmitted over the network. We propose a novel method to reconstruct video frame sizes from encrypted Wi-Fi packets and extract the sizes of inter-frames. Even though the payloads of packets are encrypted, SCamF reconstructs video frames by using a few pieces of available information, such as packet directions, packet length, and the sequence numbers obtained from the MAC headers.

Fragmentation unit size estimation: We separate the received packets into packet flows with the same transmitter/receiver pair, and infer the size of the fragmentation unit (FU) [42] of each packet flow since it depends on the type of spy camera. To transmit video frames over a Wi-Fi network, a frame greater than the FU size is divided into multiple FU-sized packets and then transmitted. The FU size can be found by analyzing the time interval between packets. Since wireless cameras transmit a video in frames, there is a difference in the time interval between different frames and within the same frame. Fig. 5 shows inter-packet intervals between consecutive packets within the same frame, and belonging to different frames of 534 packets from a wireless camera. Error bars represent 95% confidence intervals. The inter-packet interval in the same frame is much shorter than that in different frames. Thus, if the sizes of two consecutive packets are the same and the inter-packet interval is $\ll \theta$ ($\approx 10ms$), we determine it as the FU size.

Packet loss recovery: After finding the FU size of a packet flow, we can obtain the size of an original video frame by combining consecutive FU-sized packets and a subsequent small size packet.

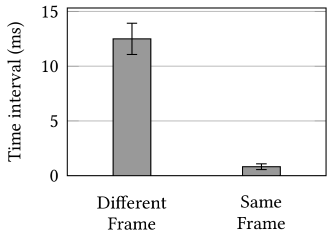


Figure 5: Comparison of inter-packet intervals between consecutive packets within the same frame, and belonging to different frames.

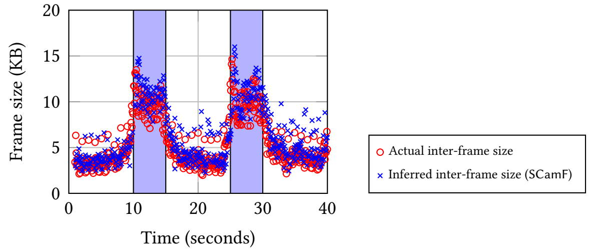


Figure 6: Comparison of inter-frame sizes: Actual vs. Inter-frame extraction result of SCamF.

Also, to compensate for the packet loss, we use sequence numbers of packets and the inter-packet time interval. For example, suppose we receive three FU-sized packets (FU size: 1500B) with sequence numbers n, $n+1$, and $n+2$, followed by an FU-sized packet and a 300B packet with sequence number $n+4$ and $n+5$, respectively, as shown in Fig. 3. If the time intervals before and after the lost packet with sequence number $n+3$ satisfy $\ll \theta$, then the packet is regarded as belonging to the same frame as other packets. Therefore, packets from n to $n+5$ are reconstructed into a 7.8 KB frame.

Inter-frame extraction: After reconstructing video frame sizes, we extract inter-frame sizes which reflect the change in the surrounding environment. Our goal is not to make an accurate inference of all video frame sizes, but to effectively observe the inter-frame size changes according to user movements. Therefore, in this section, we remove video frame sizes that affect the observation of the inter-frame size pattern. Fig. 4 summarizes the inter-frame extraction procedures. First, SCamF removes outliers that have large

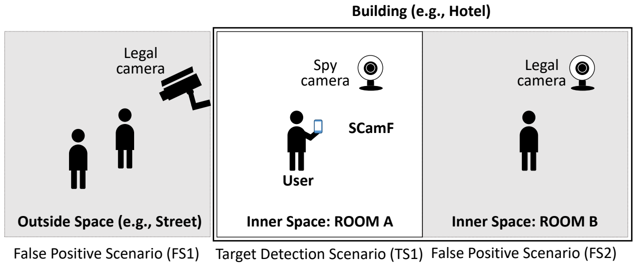

Building (e.g., Hotel)
Legal camera
Spy camera
SCamF
User
Legal camera
Outside Space (e.g., Street)
Inner Space: ROOM A
Inner Space: ROOM B
False Positive Scenario (FS1)
Target Detection Scenario (TS1)
False Positive Scenario (FS2)

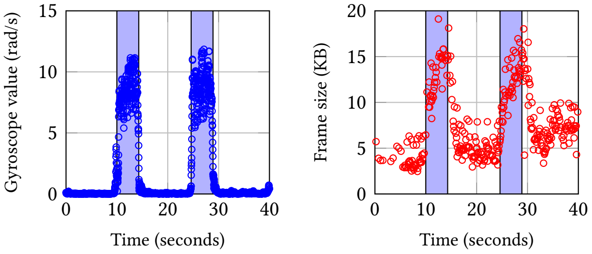
Figure 7: Target detection scenario (TS1) and potential false positive scenarios (FS1 and FS2).

differences in size from inter-frames as shown in Fig. 4a. In particular, since I-frames are much larger than inter-frames, SCamF efficiently eliminates I-frames by removing outliers. In order to remove the outliers, SCamF utilizes the interquartile range (IQR) which is usually used to find outliers. The IQR is the first quartile, Q_1 subtracted from the third quartile, Q_3 (i.e., $= Q_3 - Q_1$). Assuming that the frame size follows a normal distribution, about 95% of frame sizes fall within the range between $median - 1.5 \times IQR$ and $median + 1.5 \times IQR$. SCamF eliminates outliers by removing the data outside this range. Second, SCamF removes periodic frames (called *noise frames*) of the same size among the reconstructed frames (Fig. 4b), which are inferred to be non-video frames, such as audio or control frames. Third, a set of frames that is not related to the change of a scene remaining after the noise frame removal, is observed by several spy cameras. Since the remaining noise frames have a different pattern from video inter-frames, if there are remaining noise frames, the distribution of frame sizes appears to follow a bimodal distribution [19], which is a combination of an inter-frame set and the noise frame set. Fig. 4c shows the removal of the remaining noise frames. The final result is depicted in Fig. 4d. Compared to Fig. 4a, one can clearly observe the changes in data according to the user's movements after extracting the inter-frames (the user's movement periods are indicated by blue boxes). Fig. 6 shows that the size of the inter-frames reconstructed by SCamF and that extracted from the actual video is very similar, corroborating SCamF's ability to extract inter-frames.

4.4 Camera traffic classification

Since there are a number of connected devices on Wi-Fi networks [27], it is inefficient to investigate all connected devices nearby a smartphone user to detect spy cameras. Therefore, SCamF first identifies the traffic with wireless camera characteristics via a network traffic analysis. We call the traffic (SCamF's detection target) *camera traffic* and the other traffic *non-camera traffic*. From experimental observations, we find four features: traffic volume, inter-packet time interval, FU rate, and frame per second (FPS) that effectively differentiate between the camera and non-camera traffic. We have collected and analyzed a total of 1000 traffic traces from 20 cameras and a total of 450 traffic traces from 9 non-cameras to observe camera traffic features (detailed in §5.1).

Table 1 shows the measured values of the above four features across different traffic types. First, the periodicity of camera traffic ensures a certain amount of traffic volume all the time. We measured the traffic volumes per unit time (3 seconds in our measurements) and in each traffic we found the minimum traffic volume over the whole measurement time. We observe that camera traffic mostly shows a higher minimum traffic volume than non-camera traffic except for download traffic. Even though Video on demand (VOD) services have large traffic volumes, they exhibit fluctuations and low traffic periods because of the buffering [3].



(a) Sensor (Gyroscope) data **(b) Inter-frame size**

Figure 8: Changes of inter-frame size and sensor data according to the user's movements.

Second, like traffic volumes, most non-camera traffic traces show lower minimum FPS than the camera traffic due to the irregular packet transmission. SCamF calculates the average FPS per unit time with the video frame size reconstruction method (§4.3) by assuming each target traffic as camera traffic. When it comes to download traffic, it constantly generates a high volume of traffic, but the number of combined frames is small because the FU rate is high in download traffic. This results in low FPS.

In addition, camera traffic shows different inter-packet intervals from non-camera traffic. As shown in Fig. 5, the inter-packet interval between different frames is much larger than that within the same frame, which results in a high average inter-packet interval. Table 1 shows that VOD, download, and picture traffic traces have much smaller average inter-packet time intervals than camera traffic traces.

The FU rate, which indicates the ratio of FU-sized packets to the total received packets, is also useful for camera traffic classification. As shown in Table 1, VOD and download services have high FU rates since they usually generate burst traffic. Unlike burst traffic, the FU rate is relatively low in camera traffic because it is composed of not only video packets of various sizes but also other kinds of data such as audio and control which have lower data rates. Therefore, we can distinguish camera traffic by observing the FU rate per unit time.

SCamF classifies traffic as camera traffic when it shows all of the four feature characteristics. Discarding non-camera traffic in the following steps greatly reduces the user effort and time overhead to detect spy cameras.

4.5 Spy camera detection

In §4.4, we detect devices generating camera traffic, but not all of them are spy cameras recording the users' movements. For example, wireless cameras outside of the user space can also generate camera traffic. As depicted in Fig. 7, our goal is to detect spy cameras in the same space as the user (TS1) and remove potential false positives like two FP scenarios (FS1 and FS2). To verify the presence of spy cameras within the detection area while minimizing false positives, SCamF observes the change of inter-frame sizes according to the user's movement which is measured by smartphone sensors. If the pattern of the user's movement and the change in inter-frame size coincide, the device is deemed to be recording and streaming the user's movement in real-time. The details of user movements are described in Appendix §B.

We define two modes of user movement — *active mode* in which the user takes movement, and *stationary mode* in which the user is

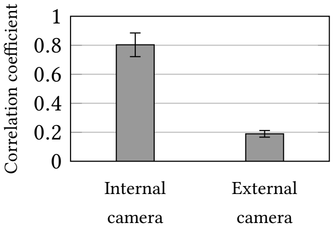


Figure 9: Correlation coefficient between movement mode indicating data and inter-frame sizes.

standing still — and calculate the correlation between the change of movement modes and the pattern of inter-frame size change. We call this process *FP prevention*. First, SCamF determines the stationary mode period and active mode period by measuring the smartphone sensor values as shown in Fig. 8a (The active mode periods are represented as blue boxes). The gyroscope sensor is used to measure the user's movement. After determining the movement mode period sequence, SCamF verifies whether the inter-frame size changes match with it. SCamF performs both a threshold-based matching algorithm and a Kendall rank correlation test [1].

In the threshold-based algorithm, SCamF measures the number of inter-frame of which size is higher than the threshold in each period. If more than 10% of inter-frames' sizes are higher than the threshold Th_D in an active mode period, SCamF regards the camera traffic data matches with the active mode period. Likewise, if less than 10% of inter-frames' sizes are higher than the threshold in a stationary mode period, SCamF regards the camera traffic data match with the stationary mode period. If the camera traffic data matches with all the sequences of the periods, SCamF concludes the video traffic is recording the user. The threshold Th_D is set using the stationary data. Since the user remains stationary during the detection of camera traffic in §4.4, we reuse the data collected in §4.4 for the calculation of the threshold. Since the threshold is calculated using the data collected from each device, SCamF provides adaptive detection for each device. Th_D is calculated as $X + 3.09\sigma$, indicating that 99.9% of inter-frame sizes are less than Th_D when the inter-frame size follows the normal distribution. X is the average and σ is the standard deviation of the collected inter-frame sizes.

From observation of many types of wireless cameras, we have found that in some cameras, the inter-frame sizes are not sufficiently reduced in a short time after the period change from active mode to stationary mode. The threshold-based detection fails to detect such camera devices. To detect such cameras, we propose a correlation test. We calculate the correlation between inter-frame sizes and a mode indicating data sequence which has a sequence of 1's during the active mode period and 0's for the stationary mode period. As described above, the mode period is determined by the sensor data. Since Kendall rank correlation [1] is suitable for comparing two data with different scales, SCamF uses it to measure the similarity between mode indicating data and inter-frame sizes. The Kendall correlation coefficient ranges between -1.0 and 1.0, and when the value is close to 1.0, the similarity between the two data is high. Across 20 types of wireless cameras, we measure correlation coefficients when cameras are in and outside the detection area. As shown in Fig. 9, when a camera is streaming a user's movement, the mode indicating data and the frame size are highly correlated, while external cameras show low correlation, thus corroborating the suitability of our correlation test for detecting the user's movement.

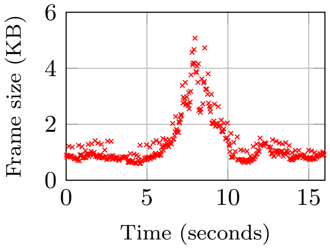
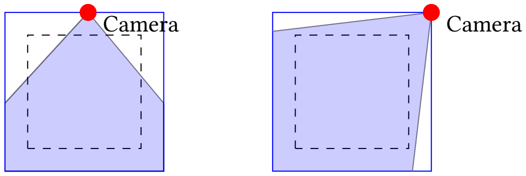


Figure 10: Inter-frame size pattern according to a user movement which causes a distance change between a spy camera and a user.



(a) Blind spot occurrence

(b) Camera's field of view covers the entire user path

Figure 11: Camera's field of view and blind spot example: The dotted line and blue colored part each represent the user's movement path and the camera's field of view.

In addition, we perform the threshold-based matching algorithm and a Kendall rank correlation test by applying a time shift of up to 3 seconds to the sensor data to compensate for the delay between the camera's recording and network transmission.

4.6 Spy camera localization

Lastly, we determine the location of the spy camera which is detected in §4.5, via inter-frame size patterns according to the distance between the spy camera and a user while the user walks along the walls in the room. The video frame reconstruction makes it possible to find the exact location of spy cameras.

4.6.1 Basic Idea. By exploiting the nature of video encoding where inter-frame sizes become larger as there are larger movements in the scene, we localize a spy camera recording a user's movements. The size of an object in a video scene varies with the distance between the object and a camera. Therefore, an object moving near a camera causes larger changes in the video scenes than when it moves from far away. That is, the encoded inter-frames have larger sizes when a user moves near the camera.

Assuming that a spy camera is installed on the walls or close to the walls, SCamF guides the user to walk along the wall in the room to find the location of the spy camera. Fig. 10 shows an example of reconstructed inter-frame sizes as a user passing by a spy camera. As the figure shows, the frame sizes increase as the user moves closer to the camera and the inter-frame size shows a distinct peak when the user passes through the position closest to a spy camera. Therefore, the localization of spy cameras can be simplified to the peak-finding problem.

4.6.2 Practical considerations. Some practical issues need to be addressed to maximize the localization accuracy. First, some wireless cameras transmit other frames of similar sizes to those of inter-frames and these may remain after inter-frame extraction. We refer to these frames as *noise values* which are assumed to be uniformly distributed. Because of the noise values, finding a peak by detecting the maximum value or observing the inter-frame size change ratio is not suitable. Second, obstacles such as furniture, and the

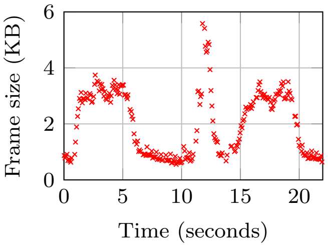
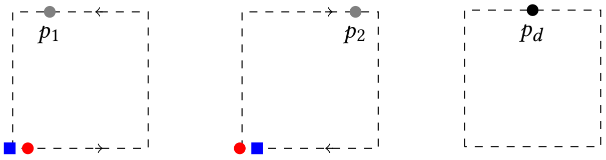


Figure 12: Inter-frame size pattern with blind spots.



(a) First round **(b) Second round** **(c) Final result**

Figure 13: Delay cancellation.

camera's field of view can affect the inter-frame sizes regardless of the distance between a user and a spy camera. Depending on the position and angle of a spy camera, the visibility of the camera covers a user's entire movement path (Fig. 11b), or blind spots exist (Fig. 11a). Obstacles can also make blind spots. If there are blind spots in the user's moving path, significant changes will occur in a scene when the user enters or exits the camera's field of view. Also, there is no change in a scene when the user is out of the camera's field of view. This causes sudden increases and decreases in inter-frame sizes and generates several local maxima of inter-frame sizes as shown in Fig. 12.

Considering these phenomena, we propose a peak-finding algorithm with an adaptive threshold, Th_{loc}. SCamF calculates Th_{loc} as $X + 1.28\sigma$ by default, where X is the average and σ is the standard deviation of the inter-frame sizes. $X + 1.28\sigma$ indicates that 10% of the inter-frame sizes exceed Th_{loc} when the inter-frame size follows a normal distribution. SCamF finds all positions with at least one inter-frame size exceeding Th_{loc} per unit time as local maxima and counts the number of inter-frames whose size exceeds Th_{loc} for each position. If there is more than one local maximum, SCamF increases Th_{loc} until only one peak remains. Conversely, if no peak is detected, SCamF lowers Th_{loc}. After adjusting Th_{loc} a predefined number of times, the local maximum with the largest number of inter-frames exceeding Th_{loc} is determined as a peak. Since we assume the uniform distribution of noise values, our peak-finding algorithm effectively detects the most distinct peak even in noisy data.

We also consider time delays. There are the measurement delay as well as the delay that changes in a scene are actually reflected in inter-frame sizes. The delay may vary with the type of camera or measurement device. In order to accurately localize a spy camera, we ask a user to walk in the opposite direction after s/he walked along the detection area once. For example, if the user walks around the detection area clockwise in the first round, SCamF asks the user to walk around counterclockwise in the second round. As illustrated in Fig. 13, a user starts at the red circle, walks along the dotted line, and arrives at the blue square. SCamF finds peaks at p_1 and p_2 in each round, respectively, and finally detects the midpoint of the peaks found in each round as the position of the spy camera, p_d. This way, the delay can be canceled out.

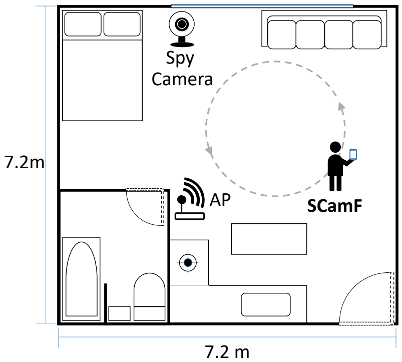


Figure 14: Testbed setup (space size: $7.2m \times 7.2m$**).**

The location of the detected spy camera on the user's movement path can be displayed as the UI example in Fig. 24b (Appendix §B). Since SCamF automatically tracks a user's movement with sensor data, SCamF localizes a spy camera regardless of the shape of a room or a user's moving path. Furthermore, SCamF can simultaneously detect and localize multiple cameras on the same channel since it separates the received packets into each packet flow through MAC addresses as described in §4.3.

5 PERFORMANCE EVALUATION

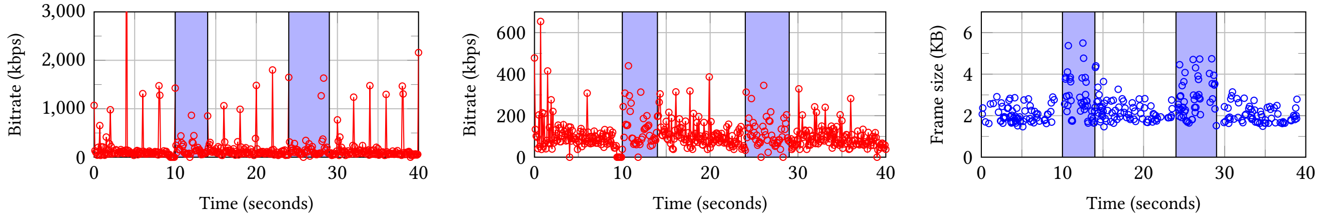
5.1 Experimental setup

To detect and localize spy cameras conveniently with mobile devices, such as smartphones and tablets, which are ubiquitous in modern life, we have implemented SCamF on commodity Android smartphones. The details of our implementation are described in Appendix §A. Also, We set up a testbed to mimic a real hotel or Airbnb room as shown in Fig. 14 where spy cameras and a Wi-Fi AP to which they are connected are installed. In the testbed, we conduct experiments across 20 different types of wireless cameras and 9 types of non-camera network traffic. We selected 14 wireless cameras from best-selling cameras on Amazon and Spy Gadget [16], and 6 from DeWiCam [4, 5]. There are 31 other APs and 18 devices installed in the testbed. All of these devices are connected to the network through APs. The bandwidth of the AP to which the experimental cameras are connected is 20 *Mbps*. The average uplink bandwidth utilization is 74.58%, which is considered a lightly congested network. Table 4 (Appendix §C) lists 20 wireless cameras and their specifications used for experiments in our testbed. Nine types of non-camera traffic are generated and transmitted through the AP in the testbed room. Non-camera traffic is categorized into four types of traffic: VOD, download, picture, and game. VOD traffic includes YouTube, Netflix, and Naver TV which transmits stored videos. Download traffic is collected from downloading apps (e.g., Call of Duty [2]), and picture traffic is collected from news, blog, webtoon, and a shopping mall app that mainly consist of pictures. Game traffic is collected while running the mobile game Fortress [24]. We collect the traffic by capturing wireless packets sent and received by mobile phones or laptops.

Five volunteers conduct experiments in spy camera detection and localization that require user interaction with SCamF. We measure and/or calculate true-positive (TP), false-positive (FP), false-negative (FN), and true-negative (TN) rates, precision, recall, accuracy as well as F1 score.

5.2 Effectiveness of video frame level analysis

To the best of our knowledge, all previous works for detecting spy/hidden cameras by analyzing wireless traffic use only network packet level information such as bitrates. The previous spy camera



(a) Network packet level analysis **(b) Network packet level analysis with removal of peaks** **(c) Video frame level analysis**

Figure 15: Comparison of traffic analysis techniques: network packet level vs. video frame level analysis (SCamF).

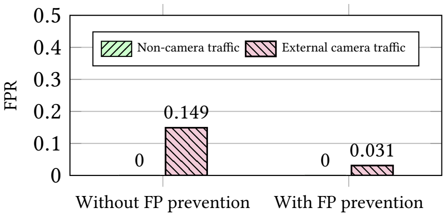


Figure 16: Detection accuracy with/without FP prevention.

detection approaches [4, 5, 21, 37] detect a spy camera streaming a user by observing bitrate changes according to the user's movement or changes in the lighting condition. As a result of analyzing the traffic of various types of cameras, we found that this method works only for a few types of cameras. Fig. 15 shows an example of bitrate changes based on network packet level analysis, which are used in previous spy camera detection approaches, and the inter-frame size change obtained by our frame reconstruction according to a user's movement. As shown in Fig. 15a, it is difficult to infer the user's movement from the bitrate pattern. Even after the periodic peaks are removed to reduce the influence of I-frames [21], the bitrate changes are still not noticeable due to the noise frames as shown in Fig. 15b. On the other hand, the reconstructed frame size change is clearly observed according to the user's movement as shown in Fig. 15c. Therefore, video frame size reconstruction of SCamF enables effective detection and localization of spy cameras.

5.3 Performance of camera traffic classification

As described in §4.4, we use four features to differentiate camera traffic from non-camera traffic: traffic volume, inter-packet time interval, FU rate, and FPS. Based on the analysis results in §4.4, we set the threshold for each feature. We classify traffic as non-camera traffic when the minimum traffic volume is less than 5000 *bps*, the average inter-packet interval is less than 2 *ms*, the FU rate is greater than 0.98, or the minimum FPS is less than 0.5. SCamF observes each traffic for 10 seconds to detect camera traffic. Tests are repeated 50 times for each traffic. Table 2 shows the performance of distinguishing camera traffic from non-camera traffic. SCamF is shown to differentiate the camera traffic from non-camera traffic with a precision of 0.94 and recall of 1.0.

In addition, we investigate the performance of SCamF to classify the network traffic of multiple simultaneous cameras. We test 10 times across 20 types of wireless cameras. When the number of cameras is one or two, SCamF perfectly classify the camera traffic in all tests with the performance of TPR = 1, and when the number of cameras is three, the three cameras are classified with the simultaneous detection probability of 0.99. The decrease in camera traffic classification accuracy as the number of cameras increases is conjectured to be caused by network congestion due to an increase of traffic. In order to analyze this result in detail, we evaluate SCamF further in §5.6 while accounting for network congestion.

Table 2: Camera traffic classification performance.

Precision	Recall	Accuracy	F1 score
0.935	1	0.978	0.966

5.4 Performance of spy camera detection

We study the performance of detecting spy cameras with user movements. The distance between a camera and a user, and the direction of the user affect the detection performance of SCamF. In order to investigate these effects, we measure the TPRs of spy camera detection according to the distance between a camera and a user across three types of motions: waving a hand facing a camera, with the user's back toward a camera, or while turning at the same spot. Cameras 3, 8, and 9 are used for the test. We test 5 times for each position and motion. As shown in Fig. 19, as the distance between a camera and a user increases, TPRs decrease because the size of the user on the camera screen becomes smaller. If the user has his/her back to the camera, the TPR is the lowest because the user's body blocks the movement from the camera. In the case of hand waving while turning in place, it shows the highest TPR because the user's motion size is the largest and the user's direction becomes irrelevant to the detection. Therefore, in subsequent experiments, waving a hand while turning in place is used as a user motion. Since SCamF asks a user to take a motion in the center of the detection area, this user motion is valid even in a space of 15 meters wide.

The overall spy camera detection performance is measured with 20 cameras. For each experiment, we repeat tests 10 times and average the results. Fig. 17 shows the TPRs for each camera and the average TPR which is 0.97. To investigate the performance of SCamF's FP prevention, we analyze the FPRs for non-camera and external camera traffic. External camera traffic comes from wireless cameras outside the detection area as shown in FS1 and FS2 in Fig. 7. To mimic external camera traffic, we deploy cameras in a real office where 5 engineers work. They move around the office as usual, and do not make any artificial movements during the recording.

We compare the performance of our detection (with FP prevention) with that of a detection method without FP prevention which determines the target device as a spy camera when the inter-frame sizes are higher than the threshold when a user moves once.

In our testing, each experiment starts in the active mode and changes the mode twice. Nine types of non-camera traffic and external camera traffic from 20 types of wireless cameras are collected. For each test, traffic data is collected for 1 hour and tested 50 times. Fig. 16 plots SCamF's performance with and without FP prevention. For non-camera traffic, SCamF achieves an FPR of 0 by eliminating even a few false positives from the camera traffic classification. The FPR for external camera traffic is 0.149 without FP prevention. With FP prevention, the FPR for external camera traffic decreases to 0.031, indicating that FP prevention effectively minimizes potential false positives.

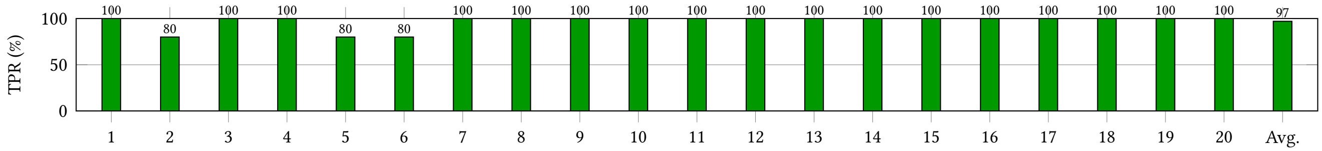


Figure 17: SCamF's spy camera detection performance for each wireless camera.

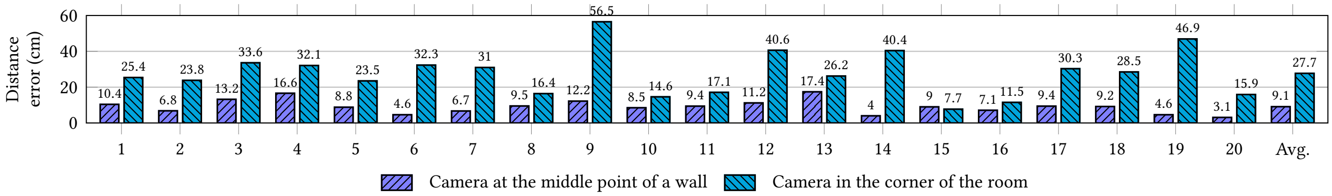


Figure 18: SCamF's spy camera localization performance for each wireless camera.

5.5 Performance of spy camera localization

In order to evaluate SCamF's localization performance, we measure the distance error. We install a camera at two positions, the middle point of a wall and a corner of the room, as depicted in Fig. 11, and measure SCamF's performance with each camera installation position. The moving path of an experimenter is approximately 30 *cm* away from the walls and the experimenter moves at about 1.4 m/s, which is the average walking speed of people. Twenty cameras are tested 5 times for each installation location. After finding the location of a spy camera, SCamF displays the detected location on the experimenter's moving path. The distance error is calculated as a difference between the detected position and the projection of a camera's location on the user's moving path. Fig. 18 shows the distance error in detecting each camera according to each installation location. The average distance error is 9.1 *cm* when a camera is at the center of a wall and is 27.7 *cm* when a camera is at the corner of the room. As described in §4.6, blind spots may occur depending on the installation position and angle of a camera, and a camera's angle of view. When the camera is at the center of a wall, blind spots occurred as shown in Fig. 11a in most cameras. On the other hand, when a camera is installed in a corner of the room, most cameras' field of view covers the experimenter's entire moving path. Due to blind spots, a distinct peak is observed for a short time, resulting in a smaller distance error when a camera is installed in the middle of a wall. For all cameras and installation positions, SCamF accurately localizes spy cameras with only centimeter-level distance errors.

5.6 Impact of network congestion

To evaluate the robustness of SCamF under network congestion, we evaluate SCamF's camera traffic classification and spy camera detection while changing network congestion level. Since it is difficult to actually generate a harsh network environment, we adjust the network congestion level by artificially applying sequence number drops to the packets captured through the monitor mode. As described in §4.3, SCamF reconstructs video frames based on sequence numbers. Therefore, the performance of SCamF is analyzed according to the sequence number loss rate, not the packet loss rate. Since the network retransmits packets to recover a lost packet, a sequence number loss implies more packet losses. The sequence number loss rate is calculated as (the number of lost sequence numbers) / (the last received sequence number − the first received sequence number).

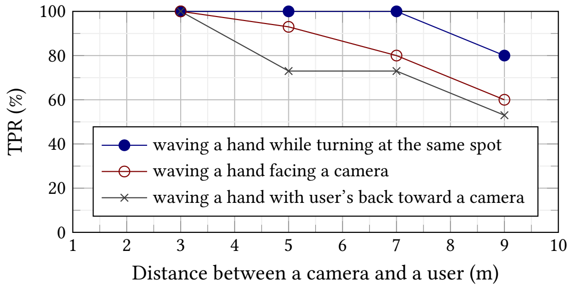


Figure 19: Spy camera detection performance of SCamF according to the distance between a camera and a user, and different motions.

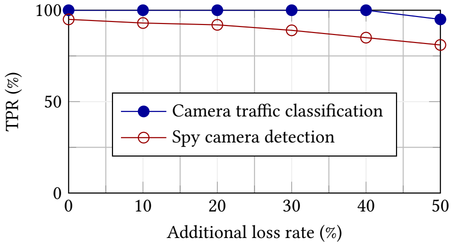


Figure 20: Performance impact of network congestion.

Fig. 20 shows the average TPR performance of SCamF when additional sequence number losses are added to the natural losses. The natural sequence number loss rate in our experimental environment is about 5%. All 20 cameras are tested and 5 random seeds are used to generate sequence number drops. In the case of camera traffic classification, all TPRs are 100% until the additional sequence number drop rate is 40%, and even at a 50% drop rate, it shows 95% TPR. This result shows that our features used for camera traffic classification are robust against network congestion. For spy camera detection, performance slightly degrades with higher network congestion, but achieves a TPR of over 80% even at an additional loss rate of 50%.

5.7 Comparison with prior work

Table 3 shows a comparison of SCamF's overall functionality with prior work. Since the terminologies used in each work are different, we classify the functionalities according to terminologies defined in §4. For example, camera localization in DeWiCam [4, 5] is the same as spy camera detection of SCamF.

As shown in Table 3, only SCamF and MotionCompass [18] provide all functions of camera traffic classification, spy camera detection, and localization. However, MotionCompass targets to detect and localize only wireless cameras equipped with motion

Table 3: Functionality comparison with prior work.

	Packet loss recovery	Camera traffic classification	Spy camera detection	Spy camera localization	Method of encrypted traffic analysis
SCamF	✓	✓	✓	✓	Network packet & video frame level
SNOOPDOG [37]	✗	✗	✓	✓	Network packet level
MotionCompass [18]	✗	✓	✓†	✓†	Network packet level
DeWiCam [4, 5]	✗	✓	✓	✗	Network packet level
Blink [21]	✓	✓	✓	✗	Network packet level

† It only detects and localizes wireless spy cameras with motion sensors.

sensors and localizes a wireless camera only in the situation where users can cross the video recording boundaries as described in §2. The main strength of SCamF is that SCamF detects and localizes wireless spy cameras by inferring video frame information from encrypted network traffic, whereas prior work uses only network packet level information which can be obtained directly from encrypted network traffic. This allows SCamF to accurately observe changes in spy camera traffic according to user movements.

To validate the effectiveness of video frame level analysis, we compare the spy camera detection performance of SCamF with two state-of-the-art wireless camera detection methods that utilize network packet level analysis, DeWiCam [4, 5] and SNOOPDOG [37]. In order to detect a spy camera according to a user's motion, both methods observe changes in the entire size of a traffic flow while SCamF extracts and observes only inter-frame sizes. Since the performance of each method can be affected by the surrounding environment, camera type, user movement, etc., the spy camera detection methods of DeWiCam and SNOOPDOG were implemented and tested under the same conditions as SCamF as described in §5.1 and §5.4. The motion duration of DeWiCam is set to 15 seconds. To evaluate spy camera detection performance, DeWiCam authors conducted experiments with 3 out of 20 cameras, and SNOOPDOG tested TPR with 7 cameras and FPR with 1 camera. In our experiment, we used all 20 cameras to test the spy camera detection performance of DeWiCam, SNOOPDOG, and SCamF as described in §5.4. Fig. 21 shows the average TPRs and FPRs of SCamF, DeWiCam, and SNOOPDOG. The error bars indicate the minimum and maximum TPRs and FPRs. As shown in Fig. 21, SCamF achieves the highest TPR and significantly lowers the FPR compared to the prior work. In addition, SCamF shows stable TPR and FPR performance regardless of the type of camera, whereas the TPRs and FPRs of the prior work vary greatly with the type of camera.

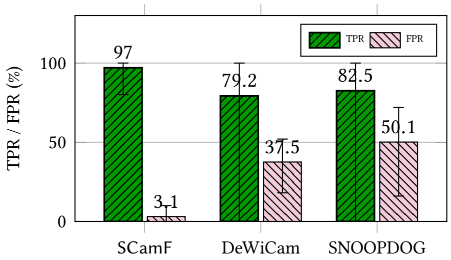


Figure 21: Performance comparison of spy camera detection.

6 DISCUSSION

6.1 Deployment Considerations

One can consider two factors for SCamF's wide deployments. First, SCamF uses the MAC address to separate different network traffic flows. Though mobile devices have recently adopted MAC address randomization [41], it does not affect SCamF's detection/localization because even if MAC address randomization is applied, devices use consistent MAC addresses after establishing a network flow.

Second, SCamF utilizes the Wi-Fi monitor mode to sniff Wi-Fi traffic across 2.4 & 5 GHz Wi-Fi channels. Though this enables SCamF's users to detect/localize wireless spy devices without any technical background, which SCamF does not require users to be associated with the same Wi-Fi AP as the target spy cameras, a limitation is that the mobile device during SCamF's operation cannot perform normal Wi-Fi communication under the Wi-Fi monitor mode. This inconvenience can be minimized via automatically switching from the legacy mode to the monitor mode during SCamF's operation only, vice versa.

6.2 Limitations

By the nature of the underlying detection logic, SCamF can only detect live streaming spy cameras on Wi-Fi networks. Other types of cameras that store the recorded video in the local storage or transmit the stored video later are not in the scope of SCamF. However, recently, most crime cases [6, 7, 39] have occurred by wireless spy cameras because they are easy to deploy and manage, and their proportion is rapidly increasing in the commercial market [16]. Therefore, SCamF is applicable to most prevalent scenarios. To cover a wider spectrum of spy cameras, it might be helpful to utilize other types of solutions using RF signals [15] or visual properties of the camera lens [10] in addition to SCamF.

SCamF does not consider highly-skilled attackers who figure out the underlying detection algorithms and actively modify traffic (traffic padding, traffic injection, traffic delay, etc. [11, 12, 28, 45], which need firmware modification of a spy camera) to evade our detection. Countermeasures against such active attacks are part of our future inquiry.

7 CONCLUSION

We proposed SCamF, a fine-grained encrypted traffic analysis approach to detect and localize wireless spy cameras by using both network packet and video frame level information inferred from the encrypted Wi-Fi traffic, which is robust to network congestion. As a result, SCamF classifies camera traffic and non-camera traffic, determines if a suspicious device is actually recording a user's movements, and localizes it. To further reduce the false positive rate, i.e., ignoring legal cameras outside the user space, SCamF utilizes smartphone sensors to track the user's activity pattern and exploits the sensor readings to calculate the exact correlation between a video frame size change and the user's movement. Moreover, we implemented SCamF as an Android app in commodity smartphones. Using the extensive experimentation on a real testbed across 20 types of wireless cameras, we showed SCamF to detect wireless spy cameras with an average TPR of 0.97 and FPR of 0.031, and localize spy cameras with only centimeter-level distance errors.

REFERENCES

[1] Hervé Abdi. 2007. The Kendall rank correlation coefficient. *Encyclopedia of Measurement and Statistics. Sage, Thousand Oaks, CA* (2007), 508–510.

[2] Inc. Activision Publishing. 2021. Call of Duty®: Mobile - Season 5: In Deep Water. https://play.google.com/store/apps/details?id=com.activision.callofduty.shooter&hl=en_US&gl=ES.

[3] Pablo Ameigeiras, Juan J Ramos-Munoz, Jorge Navarro-Ortiz, and Juan M Lopez-Soler. 2012. Analysis and modelling of YouTube traffic. *Transactions on Emerging Telecommunications Technologies* 23, 4 (2012), 360–377.

[4] Yushi Cheng, Xiaoyu Ji, Tianyang Lu, and Wenyuan Xu. 2018. DeWiCam: Detecting Hidden Wireless Cameras via Smartphones. In *Proceedings of the 2018 on Asia Conference on Computer and Communications Security* (Incheon, Republic of Korea) *(ASIACCS '18)*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3196494.3196509

[5] Yushi Cheng, Xiaoyu Ji, Tianyang Lu, and Wenyuan Xu. 2020. On Detecting Hidden Wireless Cameras: A Traffic Pattern-based Approach. *IEEE Transactions on Mobile Computing* 19, 4 (2020), 907–921. https://doi.org/10.1109/TMC.2019.2900919

[6] CNN. 2019. Family finds hidden camera livestreaming from their Airbnb in Ireland. https://edition.cnn.com/2019/04/05/europe/ireland-airbnb-hidden-camera-scli-intl/index.html.

[7] CNN. 2019. Hundreds of motel guests were secretly filmed and live-streamed online. https://edition.cnn.com/2019/03/20/asia/south-korea-hotel-spy-cam-intl/index.html.

[8] Thomas Publishing Company. 2022. Spy Cameras Suppliers. https://www.thomasnet.com/products/spy-cameras-95965083-1.html.

[9] Mauro Conti, Qian Qian Li, Alberto Maragno, and Riccardo Spolaor. 2018. The Dark Side(-Channel) of Mobile Devices: A Survey on Network Traffic Analysis. *IEEE Communications Surveys & Tutorials* 20, 4 (2018), 2658–2713. https://doi.org/10.1109/COMST.2018.2843533

[10] ALP Digitech. 2019. Spy hidden camera Detector. https://apps.apple.com/us/app/spy-hidden-camera-detector/id925967783#?platform=iphone.

[11] Lucas Dixon, Thomas Ristenpart, and Thomas Shrimpton. 2016. Network Traffic Obfuscation and Automated Internet Censorship. *IEEE Security & Privacy* 14, 6 (2016), 43–53. https://doi.org/10.1109/MSP.2016.121

[12] Kevin P. Dyer, Scott E. Coull, Thomas Ristenpart, and Thomas Shrimpton. 2012. Peek-a-Boo, I Still See You: Why Efficient Traffic Analysis Countermeasures Fail. In *2012 IEEE Symposium on Security and Privacy*. Institute of Electrical and Electronics Engineers, New York, NY, USA, 332–346. https://doi.org/10.1109/SP.2012.28

[13] europages. 2022. Manufacturer producer - spy cameras. https://www.europages.co.uk/companies/Manufacturer%20producer/spy%20cameras.html.

[14] Raspberry Pi Foundation. 2022. Raspberry PI. https://www.raspberrypi.org/.

[15] futureapps. 2022. Hidden Camera Detector. https://play.google.com/store/apps/details?id=hiddencamdetector.futureapps.com.hiddencamdetector&hl=en_US&gl=US.

[16] Spy Gadgets. 2021. Spy Cameras. https://www.spygadgets.com/collections/spy-cameras.

[17] Ltd. Hardkernel co. 2022. ODROID. https://www.hardkernel.com/.

[18] Yan He, Qiuye He, Song Fang, and Yao Liu. 2021. MotionCompass: Pinpointing Wireless Camera via Motion-Activated Traffic. In *Proceedings of the 19th Annual International Conference on Mobile Systems, Applications, and Services* (Virtual Event, Wisconsin) *(MobiSys '21)*. Association for Computing Machinery, New York, NY, USA, 215–227. https://doi.org/10.1145/3458864.3467683

[19] Hajo Holzmann and Sebastian Vollmer. 2008. A likelihood ratio test for bimodality in two-component mixtures with application to regional income distribution in the EU. *AStA Advances in Statistical Analysis* 92, 1 (2008), 57–69.

[20] Stephan Kleber, Lisa Maile, and Frank Kargl. 2019. Survey of Protocol Reverse Engineering Algorithms: Decomposition of Tools for Static Traffic Analysis. *IEEE Communications Surveys & Tutorials* 21, 1 (2019), 526–561. https://doi.org/10.1109/COMST.2018.2867544

[21] Tian Liu, Ziyu Liu, Jun Huang, Rui Tan, and Zhen Tan. 2018. Detecting Wireless Spy Cameras Via Stimulating and Probing. In *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services* (Munich, Germany) *(MobiSys '18)*. Association for Computing Machinery, New York, NY, USA, 243–255. https://doi.org/10.1145/3210240.3210332

[22] Fannia Pacheco, Ernesto Exposito, Mathieu Gineste, Cedric Baudoin, and Jose Aguilar. 2019. Towards the Deployment of Machine Learning Solutions in Network Traffic Classification: A Systematic Survey. *IEEE Communications Surveys & Tutorials* 21, 2 (2019), 1988–2014. https://doi.org/10.1109/COMST.2018.2883147

[23] Zhaoqing Pan, Jianjun Lei, Yun Zhang, Xingming Sun, and Sam Kwong. 2016. Fast Motion Estimation Based on Content Property for Low-Complexity H.265/HEVC Encoder. *IEEE Transactions on Broadcasting* 62, 3 (2016), 675–684. https://doi.org/10.1109/TBC.2016.2580920

[24] PANGSKY. 2021. Battle Royale Fortress. https://play.google.com/store/apps/details?id=com.pang.ftrs.google.

[25] Grzegorz Pastuszak and Andrzej Abramowski. 2016. Algorithm and Architecture Design of the H.265/HEVC Intra Encoder. *IEEE Transactions on Circuits and Systems for Video Technology* 26, 1 (2016), 210–222. https://doi.org/10.1109/TCSVT.2015.2428571

[26] Ashish Patro, Srinivas Govindan, and Suman Banerjee. 2013. Observing Home Wireless Experience through WiFi APs. In *Proceedings of the 19th Annual International Conference on Mobile Computing & Networking* (Miami, Florida, USA) *(MobiCom '13)*. Association for Computing Machinery, New York, NY, USA, 339–350. https://doi.org/10.1145/2500423.2500448

[27] Changhua Pei, Zhi Wang, Youjian Zhao, Zihan Wang, Yuan Meng, Dan Pei, Yuanquan Peng, Wenliang Tang, and Xiaodong Qu. 2017. Why It Takes So Long to Connect to a WiFi Access Point. In *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*. Institute of Electrical and Electronics Engineers, New York, NY, USA, 1–9. https://doi.org/10.1109/INFOCOM.2017.8057164

[28] Antônio J. Pinheiro, Paulo Freitas de Araujo-Filho, Jeandro de M. Bezerra, and Divanilson R. Campelo. 2021. Adaptive Packet Padding Approach for Smart Home Networks: A Tradeoff Between Privacy and Performance. *IEEE Internet of Things Journal* 8, 5 (2021), 3930–3938. https://doi.org/10.1109/JIOT.2020.3025988

[29] Shiva Raj Pokhrel and Carey Williamson. 2018. Modeling Compound TCP Over WiFi for IoT. *IEEE/ACM Transactions on Networking* 26, 2 (2018), 864–878. https://doi.org/10.1109/TNET.2018.2806352

[30] The Washington Post. 2021. Does your Airbnb or hotel have a hidden camera? Experts share tips for protecting yourself. https://www.washingtonpost.com/travel/tips/airbnb-hidden-camera-tiktok/.

[31] Atul Puri, Xuemin Chen, and Ajay Luthra. 2004. Video coding using the H.264/MPEG-4 AVC compression standard. *Signal Processing: Image Communication* 19, 9 (2004), 793–849. https://doi.org/10.1016/j.image.2004.06.003 Technologies enabling Movies on Internet, HD DVD, and DCinema.

[32] Sriram Sami, Sean Rui Xiang Tan, Bangjie Sun, and Jun Han. 2021. LAPD: Hidden Spy Camera Detection Using Smartphone Time-of-Flight Sensors. In *Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems* (Coimbra, Portugal) *(SenSys '21)*. Association for Computing Machinery, New York, NY, USA, 288–301. https://doi.org/10.1145/3485730.3485941

[33] Heiko Schwarz, Detlev Marpe, and Thomas Wiegand. 2007. Overview of the Scalable Video Coding Extension of the H.264/AVC Standard. *IEEE Transactions on Circuits and Systems for Video Technology* 17, 9 (2007), 1103–1120. https://doi.org/10.1109/TCSVT.2007.905532

[34] SpyAssociates.com Security. 2018. SPYFINDER PRO Hidden Spy Camera Detector. https://www.amazon.com/SPYFINDER-PRO-Hidden-Camera-Finder/dp/B07HVJ8VZR.

[35] Charalambos Sergiou, Pavlos Antoniou, and Vasos Vassiliou. 2014. A Comprehensive Survey of Congestion Control Protocols in Wireless Sensor Networks. *IEEE Communications Surveys & Tutorials* 16, 4 (2014), 1839–1859. https://doi.org/10.1109/COMST.2014.2320071

[36] sherry. 2019. RF Signal Detection. https://amzn.to/2TQs1A1.

[37] Akash Deep Singh, Luis Garcia, Joseph Noor, and Mani Srivastava. 2021. I Always Feel Like Somebody's Sensing Me! A Framework to Detect, Identify, and Localize Clandestine Wireless Sensors. In *30th USENIX Security Symposium (USENIX Security 21)*. USENIX Association, Berkeley, CA, USA, 1829–1846. https://www.usenix.org/conference/usenixsecurity21/presentation/singh

[38] Katty so. 2019. RF Spy Camera Detector. https://www.amazon.com/Camera-Detector-Hidden-Detecting-Tracker/dp/B07TKHYL16.

[39] Birru Dereje Teshome. 2019. Spy Camera Epidemic in Korea: A Situational Analysis. *Asian Journal of Sociological Research* 2, 1 (2019), 1–13.

[40] Geert Van der Auwera, Prasanth T. David, and Martin Reisslein. 2008. Traffic Characteristics of H.264/AVC Variable Bit Rate Video. *IEEE Communications Magazine* 46, 11 (2008), 164–174. https://doi.org/10.1109/MCOM.2008.4689260

[41] Mathy Vanhoef, Célestin Matte, Mathieu Cunche, Leonardo S. Cardoso, and Frank Piessens. 2016. Why MAC Address Randomization is Not Enough: An Analysis of Wi-Fi Network Discovery Mechanisms. In *Proceedings of the 11th on ACM Asia Conference on Computer and Communications Security* (Xi'an, China) *(ASIA CCS '16)*. Association for Computing Machinery, New York, NY, USA, 413–424. https://doi.org/10.1145/2897845.2897883

[42] Y.-K. Wang, R. Even, T. Kristensen, and R. Jesup. 2011. RTP Payload Format for H.264 Video. In *RFC 6184, Internet Engineering Task Force (IETF)*. RFC Editor, Wilmington, DE, USA, 1–101.

[43] WPTV. 2022. Martin County mother says hidden cameras caused 'path of destruction' for family. https://www.wptv.com/news/treasure-coast/region-martin-county/martin-county-mother-says-hidden-cameras-caused-path-of-destruction-for-family.

[44] Jin Xie, Wei Hu, and Zhenghao Zhang. 2011. Revisiting Partial Packet Recovery in 802.11 Wireless LANs. In *Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services* (Bethesda, Maryland, USA) *(MobiSys '11)*. Association for Computing Machinery, New York, NY, USA, 281–292. https://doi.org/10.1145/1999995.2000022

[45] Keyang Yu, Qi Li, Dong Chen, Mohammad Rahman, and Shiqiang Wang. 2021. PrivacyGuard: Enhancing Smart Home User Privacy. In *Proceedings of the 20th International Conference on Information Processing in Sensor Networks (Co-Located with CPS-IoT Week 2021)* (Nashville, TN, USA) *(IPSN '21)*. Association for Computing Machinery, New York, NY, USA, 62–76. https://doi.org/10.1145/3412382.3458257

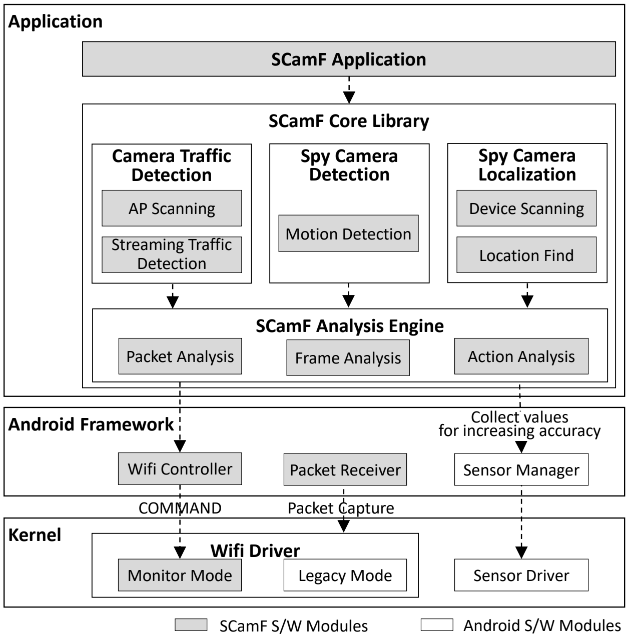


Figure 22: Software architecture of SCamF.

APPENDIX

A IMPLEMENTATION

We have implemented SCamF algorithms (§4) on an Android smartphone (Galaxy S21+). Fig. 22 shows the software architecture that consists of three independent subsystems each of which operates as a *process*: (1) SCamF application and SCamF core library in the application layer, (2) *Wi-Fi Controller* and *Packet Receiver* in Android framework layer, and (3) *Wi-Fi Monitor Mode* in kernel layer. Our SCamF application is written in JAVA. To facilitate reproducibility of our results and to make our solutions portable to other platforms such as Raspberry PI [14] and Odroid [17], SCamF core library is written in C++.

Application Layer. This layer contains SCamF application and SCamF core library. SCamF core library includes the core algorithms described in §4. SCamF core library needs to communicate with SCamF application and *Packet Receiver* to analyze received packets and notify the user of the result. In particular, while the SCamF core library analyzes packets, *Packet Receiver* continuously captures packets without delay to communicate asynchronously with each other. SCamF core library analyzes raw packets received from *Packet Receiver*, internally goes through *Camera Traffic Detection, Spy Camera Detection*, and *Spy Camera Localization* modules to structure a spy camera and send it to SCamF application. In addition, for increasing the accuracy of detection, *Action Analysis* analyzes user's activities using values collected by *Sensor Manager*.

Android Framework Layer. This layer contains *Wi-Fi Controller* and *Packet Receiver*. First, SCamF core library activates *Wi-Fi Controller* and *Packet Receiver* and requests *Wi-Fi Controller* to enable monitor mode for a specific channel. In order to detect a spy camera, each Wi-Fi channel must be traversed. Therefore, *Wi-Fi Controller* enables monitor mode and change channels at run-time.

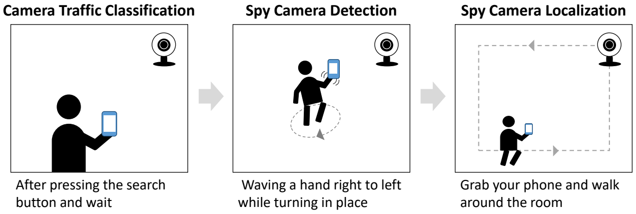
Kernel Layer. As described in §4, SCamF requires the Wi-Fi device driver and firmware to monitor Wi-Fi packets. Unfortunately, no device drivers and firmware support Wi-Fi traffic monitoring in commodity smartphones. We have customized the Wi-Fi traffic monitor mode, where the Wi-Fi driver provides a control interface to switch between monitor mode and legacy mode. When SCamF application is launched, *Wi-Fi Controller* tries to switch to monitor mode through the interface. After completing all activities, when our application is terminated, *Wi-Fi Controller* returns to legacy mode through the interface as well.

B USER INTERFACE AND OPERATION

This section describes the user interface (UI) of SCamF and the operations that SCamF requires from a user to detect spy cameras. Fig. 23 shows the user operation flow in each step of SCamF.

First, when a user launches SCamF APP and presses the search button, SCamF guides the user to wait without moving while it detects camera traffic. If SCamF detects camera traffic in the detection environment, the list of detected devices generating camera traffic is displayed. The user can exclude the known devices and select the devices he/she wants to detect if the devices are recording and streaming the user, which means if they are spy cameras. The detected devices on the same wireless channel can be selected and detected at the same time.

After the user selects the devices and presses the detection button, SCamF confirms that the selected devices are streaming the user as in §4.5. SCamF guides the user to move to the center of the detection area and wave a hand right to left while turning in place.


Camera Traffic Classification

After pressing the search button and wait

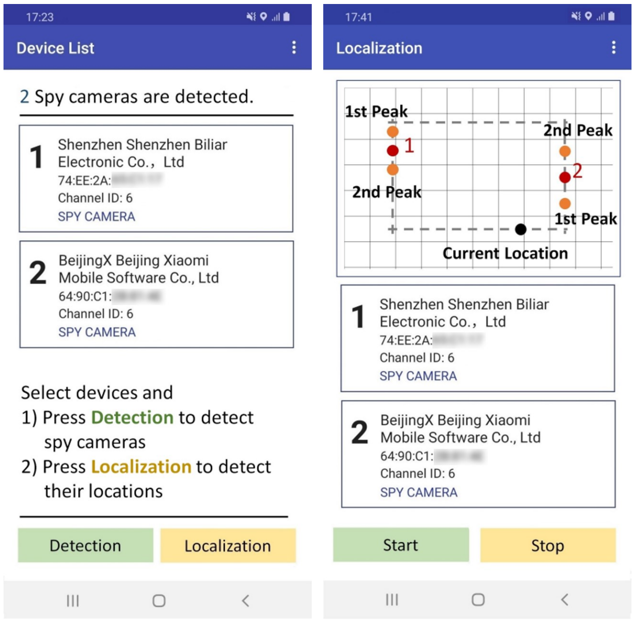
Spy Camera Detection

Waving a hand right to left while turning in place

Spy Camera Localization

Grab your phone and walk around the room

Figure 23: User operation flow.



2 Spy cameras are detected.

1 Shenzhen Shenzhen Biliar Electronic Co.，Ltd
74:EE:2A:[redacted]
Channel ID: 6
SPY CAMERA

2 BeijingX Beijing Xiaomi Mobile Software Co., Ltd
64:90:C1:[redacted]
Channel ID: 6
SPY CAMERA

Select devices and
1) Press Detection to detect spy cameras
2) Press Localization to detect their locations

1 Shenzhen Shenzhen Biliar Electronic Co.，Ltd
74:EE:2A:[redacted]
Channel ID: 6
SPY CAMERA

2 BeijingX Beijing Xiaomi Mobile Software Co., Ltd
64:90:C1:[redacted]
Channel ID: 6
SPY CAMERA

(a) Result of spy camera detection **(b) Result of spy camera localization**

Figure 24: An example of spy camera detection and localization results in SCamF Application (Sample UI).

Table 4: List and functions of wireless cameras for performance evaluation on our testbed.

No.	Brand	Model	Video/Audio Encoding	Resolution				Other Supported Functions		
				Default	High	Middle	Low	Sound	Motion	Night Vision
1	Yi	YYS.2016	H.264, AAC	1080p	✓	-	✓	✓	-	✓
2	Yi	YHS.2116	H.264, AAC	1080p	✓	-	✓	✓	✓	✓
3	Xiaomi	SXJ01ZM	H.264, AAC	1080p	✓	-	✓	✓	✓	✓
4	360	D606	H.264, AAC	1080p	✓	-	✓	✓	-	✓
5	TP-Link	TL-IPC20-2.8	H.264	720p	-	✓	-	-	-	✓
6	Amcrest	IP2M-841B-V3	H.264	1080p	✓	✓	✓	-	✓	✓
7	goospy	S64	H.264, AAC	720p	✓	✓	✓	✓	✓	✓
8	ieGeek	2.0 megapixels ip camera	H.264, AAC	720p	-	✓	✓	✓	✓	✓
9	Xiaomi	MJSXJ05CM	H.265, AAC	1080p	✓	-	-	✓	-	✓
10	Egloo	TSC-221A	H.264, AAC	1080p	✓	✓	-	✓	✓	✓
11	Hej	GKW-IC052	H.264, AAC	1080p	✓	✓	-	✓	✓	✓
12	Green	PE204	H.264, AAC	480p	-	-	✓	✓	✓	✓
13	JWC	JCURI-HOME2	H.264, AAC	1080p	✓	-	-	✓	✓	✓
14	Wisenet	SNH-P6410BN	MPEG-4, AAC	480p	✓	✓	✓	✓	-	✓
15	Relohas	S93	H.264, AAC	720p	✓	✓	✓	✓	✓	✓
16	luhoe	C-TOP	H.264	720p	✓	✓	-	-	✓	-
17	YINEW	U21	H.264	720p	-	✓	✓	-	-	✓
18	Geagle	Wi-Fi mini camera	H.264, AAC	720p	-	✓	✓	✓	-	✓
19	Xiaomi	MJSXJ09CM	H.265, AAC	1080p	✓	-	✓	✓	✓	✓
20	Wisenet	HNO-E60	H.264, AAC	1080p	✓	-	✓	✓	✓	✓

SCamF guides the user to repeat the movement (5 seconds) and then the stop (10 seconds) twice. SCamF notifies the user of the beginning and end of the movement with vibrations and sounds. Through the spy camera detection step (§4.5), SCamF detects the spy cameras and shows the list of them as shown in Fig. 24a.

Finally, the user can find the location of each spy camera via SCamF. The user holds the phone and walks around the detection area along the wall, and then he/she walks around once again in the opposite direction. As explained in §4.6, SCamF detects the frame size peak in each round and determines the middle position of the peaks as the final spy camera position. SCamF uses smartphone sensors to track the user's walking path and displays the position of detected peaks and the final location of the detected spy cameras on the user's walking path as shown in Fig. 24b.

C OPERATION TIME OF SCAMF

We have evaluated the time required at each step of SCamF. For the camera traffic classification, SCamF first detects c_1 channels through which APs communicate by collecting packets for 0.3 seconds per channel. Among the c_1 channels, SCamF detects channels with a data rate of more than 5000 bps by collecting packets in c_1 channels for 1.5 seconds per channel. If c_2 channels are detected, SCamF observes each channel for 10 seconds to detect channels in which camera traffic exists.

The spy camera detection requires a total of 30 seconds to detect cameras per channel because there are two 10 second stationary mode periods and two 5 second active mode periods for user movements as described in §4.5. Because spy cameras on the same channel can be detected at the same time, the time to detect all cameras in the detection step is $30 \times c_d$ seconds, where c_d is the number of channels detected in camera traffic classification. In the case of localization, the required time varies depending on the size of the detection area and the walking speed of a user. Similar to the spy camera detection, SCamF can simultaneously detect the location of spy cameras in the same network channel, and sequentially localize spy cameras in different channels. Therefore, the time required for localization is $\frac{l_{path} \times 2}{v_{user}} \times c_d$, where l_{path} is the length of the walking path of a user to walk around the detection area once, and v_{user} is the walking speed of the user.

For example, if SCamF scans all channels in IEEE 802.11n 2.4GHz band, and there are APs on 10 channels and 2 spy cameras on one channel, The camera traffic classification requires 28.9 seconds $(13 \times 0.3 + 10 \times 1.5 + 10)$ and the spy camera detection requires 30 seconds. If the user walks at 1.4 m/s in a square room of $3m \times 3m$, the localization requires 17 seconds.