Harvesting Smartphone Privacy Through Enhanced Juice Filming Charging Attacks

Weizhi Meng[1(✉)], Fei Fei[2], Wenjuan Li[1,2], and Man Ho Au[3]

[1] Department of Applied Mathematics and Computer Science, Technical University of Denmark, Kongens Lyngby, Denmark
weme@dtu.dk

[2] Department of Computer Science, City University of Hong Kong, Hong Kong, China

[3] Department of Computing, The Hong Kong Polytechnic University, Hong Kong, China
csallen@comp.polyu.edu.hk

Abstract. The increasingly high demand for smartphone charging in people's daily lives has apparently encouraged much more public charging stations to be deployed in various places (e.g., shopping malls, airports). However, these public charging facilities may open a hole for cyber-criminals to infer private information and data from smartphone users. Juice filming charging (JFC) attack is a particular type of charging attacks, which is capable of stealing users' sensitive information from both Android OS and iOS devices, through automatically monitoring and recording phone screen during the whole charging period. The rationale is that phone screen can be leaked through a standard micro USB connector, which adopts the Mobile High-Definition Link (MHL) standard. In practice, we identify that how to efficiently extract information from the captured videos remains a challenge for current JFC attack. To further investigate its practical influence, in this work, we focus on enhancing its performance in the aspects of extracting texts from images and correlating information, and then conducting a user study in a practical scenario. The obtained results demonstrate that our enhanced JFC attack can outperform the original one in collecting users' information at large and extracting sensitive data with a higher accuracy. Our work aims to complement existing results and stimulate more efforts in defending smartphones against charging threats.

Keywords: Mobile privacy and security · Android and iOS · Charging threat · OCR technology · Juice filming charging attack

1 Introduction

With the rapid development, smartphones have become one of the most commonly adopted devices for millions of people. International Data Corporation

W. Meng — The author was previously known as Yuxin Meng.

(IDC) reported that shipments grew 5.3% from 344.7 million in the second quarter of 2016, and that vendors shipped a total of 362.9 million smartphones worldwide in the third quarter of 2016 [9]. The number of phone users is predicted to increase from 1.5 billion in 2014 to around 2.5 billion in 2019. Nowadays, the majority of smartphones can act as a personal assistant, i.e., allowing to run various applications that help users view Office documents, access an address book or use GPS. Due to these capabilities, people are likely to store their personal and private data on the phones, and are often using smartphones in their daily lives (e.g., video-chatting with friends), which greatly increase the demand of recharging their mobile devices.

To meet this requirement, more and more public charging stations are being deployed for smartphone users. As an example, Singapore Power (SP) had planned to deploy up to 200 free mobile charging stations in various busy locations including hospitals, tertiary institutions, libraries and supermarkets [30]. Each charging station is expected to be equipped with 10 individual slots, including multiple charging connectors like mini or micro USBs, which can fit most mobile phones and tablets. Generally, these public charging facilitates can greatly benefit smartphone users in their daily lives; however, public stations may also expose a big threat on smartphone privacy and security, since we are not sure that these charging facilities are not maliciously controlled by cyber-criminals (e.g., station developers, Government agencies). As a result, there is a great need to pay more attention to charging threats [10,24].

In literature, Lau *et al.* [11] presented a malicious charging station named *Mactans*, which could harm a phone through injecting any malicious applications to collect users' secrets on iOS6 devices. Spolaor *et al.* [29] then designed a proof-of-concept application called *PowerSnitch*, and showed how an adversary could leverage a maliciously controlled charging station to exfiltrate data from Android smartphones via a USB charging cable by using power consumption in the form of power bursts. These two attacks are only effective on either iOS or Android devices. Meng *et al.* [20,21] developed a more scalable charging attack, called juice filming charging (JFC) attack, which can steal users' private information from both Android OS and iOS devices, through automatically recording phone screen (including users' inputs) during the charging period. All the interactions and screen information can be captured in the back-end as long as people keep charging their phones to the JFC charger. It is worth noting that JFC attack does not need to install any piece of applications or require any permission from users; thus, it may cause a large number of victims in practice.

Motivations and Contributions. JFC attack can work in an intelligent way by integrating Optical Character Recognition (OCR) technology in processing the recorded videos [21]. However, we perform a study and identify that how to efficiently extract information from the captured videos remains a challenge for current JFC attack. When a large amount of videos are recorded, it is very time-consuming and expensive for manual analysis. Therefore, it is very important for JFC attack to extract users' private information in an intelligent and accurate way. Motivated by this, in this work, we focus on JFC attack and attempt to enhance its performance in the aspects of extracting texts from images and correlating information. We further enable JFC chargers to upload recorded videos to a cloud environment and extract information from the captured videos in a centralized server. In the evaluation, we conduct a user study in a practical environment to investigate the practical influence of the improved attack. The contributions of our work can be summarized as below.

– We conduct a study to explore the practical performance of JFC attack and identify its limitation in accurately extracting users' private information from the collected videos. We then describe a technical way of improving the accuracy of information extraction from videos.
– In addition, our study identifies that information correlation is also a challenge for current JFC attack. To mitigate this issue, we present a compact but efficient approach for correlating information in terms of user credentials (e.g., unlock pattern, social networking account), as well as introduce how to launch JFC attack with a cloud environment.
– To investigate the practical influence of JFC attack on smartphone users' privacy, we further conduct a user study in a practical scenario. Experimental results demonstrate that the enhanced JFC attack can outperform the original one in both extracting and correlating users' private information. Our effort demonstrates the potential damage of charging attack and attempts to stimulate more research in this area.

Organization of the paper. Section 2 introduces the background of JFC attack including its features and setup details. Section 3 identifies the limitations of current JFC attack and describes how to enhance its performance in the aspects of information extraction and correlation. In Sect. 4, we collaborate with an IT center and perform a user study in a real scenario. We further discuss how to defend JFC attack in Sect. 5 and review related research studies in Sect. 6. Finally, we conclude this work in Sect. 7.

2 Background of JFC Attack

JFC attack was developed to steal users' private data through automatically video-capturing smartphone screen when the phone is awake or users are interacting with their phones during the whole charging period [20]. This attack does not require to install any part of applications or ask for any permissions from smartphone users. By integrating with OCR technology [21], JFC attack can provide various features as below:

– *It is easy to implement but quite efficient.* After installing the driver, JFC attack can be launched by any computing devices even some small devices like RaspberryPi.
– *It does not need to install any additional applications or components on phones.* This attack does not require to install any pieces of applications on phone's side.

– *It does not need any permissions.* This attack does not need to request any permissions from smartphone users.
– *With less user awareness.* It is a kind of passive attacks, which causes less user attention in real scenarios.
– *It cannot be detected or notified by any current anti-malware software.* Existing anti-malware software are not aware of JFC attack; thus, they cannot detect or notify users about this threat.
– *It can be scalable and effective in both Android OS and iOS devices.* As compared to malicious applications (malware), JFC attack is more effective as it can work in both Android phones and iPhones.
– *It can automatically process collected videos and extract secrets using OCR technology.* After collecting the videos, JFC chargers can automatically extract text from the videos and store data into files.

Threat model. We adopt the basic assumptions that phone charging is a common requirement from smartphone users, and that most users would not treat public chargers as highly sensitive or dangerous. These assumptions have been approved in relevant studies (e.g., [21]) that most smartphone users would charge their phones in public places such as airports, subways, shopping malls and so on. Charging attacks can be classified as *public* and *private*. The former is mainly related to public charging facilities like the charging stations provided in an airport, while the latter is mostly relevant to private charging facilities like a private charger from friends.

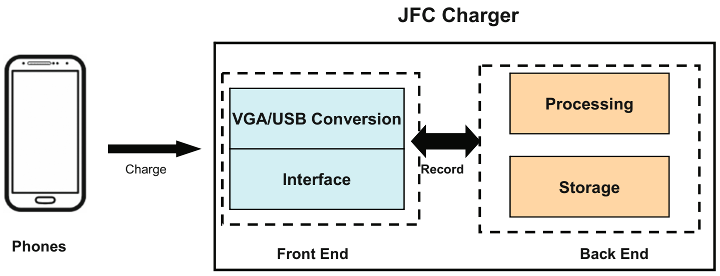


Fig. 1. The high-level implementation of JFC attack.

Basic idea. The original idea of JFC attack is from the observation that no permission would be asked when plugging phones to a projector, while the projector can display the phone screen. In addition, there are no compelling notifications on the screen when the device is being plugged, or the indicators are very small. Based on these observations, JFC attack is developed to automatically video-record phone screen by using a VGA/USB interface during the charging period. This attack reveals that the display of phone screen can be leaked through a standard micro USB connector, which adopts the Mobile

High-Definition Link (MHL) standard. For iPhones, the lighting connector is used. For Android phones, this is usually done through the micro-USB port. The micro-USB (available on most Android devices) can also involve connectors with MHL connectivity.

Real setup. The high-level implementation of JFC attack is depicted in Fig. 1. When users charge their phones to the JFC charger, their phone screens could be video-captured by the charger in the back-end and stored for later use, i.e., extracting private data from the recorded videos. To implement JFC attack, it is important to choose an appropriate VGA/USB interface, but actually there are many alternatives online. In the previous studies (e.g., [20,21]), a hardware interface named *VGA2USB* was adopted to implement JFC attack.[1] It is particularly a full-featured VGA/RGB frame grabber, which can send a digitized video signal from VGA to USB.

Figure 2 shows two examples of setting up JFC attack in practice, where Fig. 2(a) shows how to launch JFC attack on an Android phone and Fig. 2(b) describes the implementation of JFC attack on an iPhone. It is found that the connected iPhone screen can be displayed in the computer end. It is not hard to imagine that all screen information would be captured by JFC charger, including users' inputs such as typed passwords, PIN code, chatting history, etc.

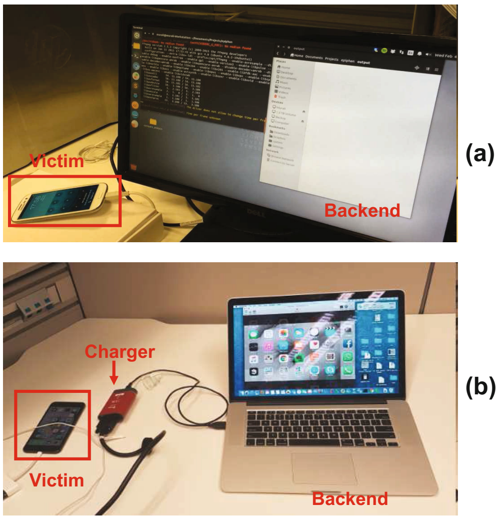


Fig. 2. Two examples of real setup for JFC attack using VGA2USB: (a) an Android phone, and (b) an iPhone.

Collected private information. The recorded videos contain the phone screen or the inputs from smartphone users; thus, various information can be extracted.

[1] http://www.epiphan.com/products/vga2usb/.

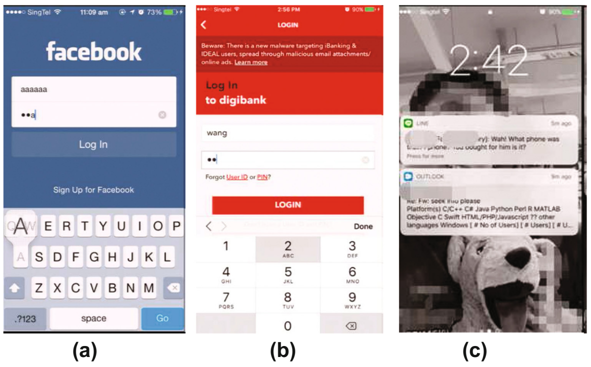
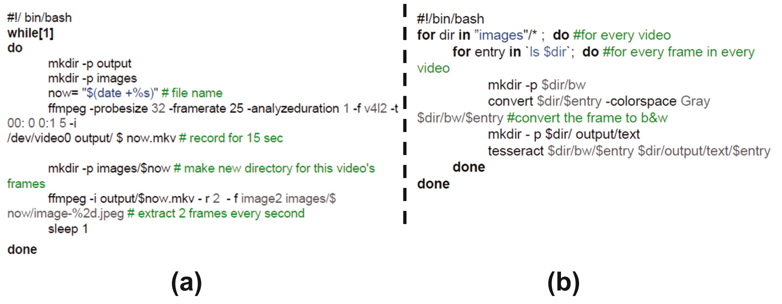

facebook
aaaaaa
Log In
Sign Up for Facebook
LOGIN
Beware: There is a new malware targeting iBanking & IDEAL users, spread through malicious email attachments/ online ads. Learn more
Log In
to digibank
wang
Forgot User ID or PIN?
LOGIN
Done
2:42
LINE
OUTLOOK
(a)
(b)
(c)

Fig. 3. Three examples of captured screen information: (a) Facebook account, (b) Bank account login, and (c) LINE and Outlook messages.

Figure 3 gives three examples of captured screen information via JFC attack, in which each of them is relevant to sensitive and private information of smartphone users. In particular, Fig. 3(a) shows the captured inputs for Facebook including the accounts and input passwords, Fig. 3(b) presents the captured screen for bank login including account name and input passwords, and Fig. 3(c) shows the captured messages from LINE chat and Outlook message. These examples demonstrate that JFC attack has a potential to become a big threat for smartphone privacy and security.

OCR technology. By integrating with OCR technology, JFC attack can process the collected videos automatically, e.g., video analysis and information extraction. There are two major phases [21]: *device checking* and *OCR processing*. The source code for Phase1 and Phase2 are presented in Fig. 4.



```
#!/ bin/bash
while[1]
do
        mkdir -p output
        mkdir -p images
        now= "$(date +%s)" # file name
        ffmpeg -probesize 32 -framerate 25 -analyzeduration 1 -f v4l2 -t
00: 0 0:1 5 -i
/dev/video0 output/ $ now.mkv # record for 15 sec

        mkdir -p images/$now # make new directory for this video's
frames
        ffmpeg -i output/$now.mkv - r 2  - f image2 images/$
now/image-%2d.jpeg # extract 2 frames every second
        sleep 1
done
```

(a)

```
#!/bin/bash
for dir in "images"/* ;  do #for every video
        for entry in `ls $dir`;  do #for every frame in every
video
                mkdir -p $dir/bw
                convert $dir/$entry -colorspace Gray
$dir/bw/$entry #convert the frame to b&w
                mkdir - p $dir/ output/text
                tesseract $dir/bw/$entry $dir/output/text/$entry
        done
done
```

(b)

Fig. 4. The code for (a) Phase1 and (b) Phase2.

– **Phase1 - Device Checking.** This phase allows the JFC charger to decide when to start the recording process. The Ephiphan frame grabber enables a

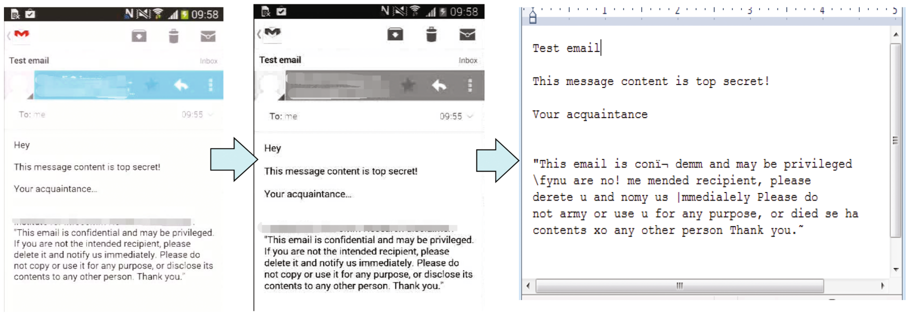
VGA device to send input through the USB. On Linux, Ephiphan devices expose the Video4Linux (V4L) API, hence the V4L API can provide a real time stream of the display of the device whenever a device is connected to the machine. The machine can periodically check the presence of a new V4L capable USB connected device. If such a device is found, the stream is piped into a file. After 15 s of device detection, the machine can automatically pause to evaluate whether the device is still connected. If connected, it continues streaming the screen contents into a file. Otherwise, it would go back into waiting mode. It is worth noting that the time of device detection can be tuned according to specific requirements.

– **Phase2 - Optical Character Recognition (OCR).** This phase allows the JFC charger to process the collected videos using OCR technology, i.e., extracting texts from the collected videos. Since OCR may take a while, it is not done synchronously with the first phase. While the machine is running at *Phase1*, the OCR code can simultaneously process the video frames and check for new frames. Then all the text files extracted from the same video can be merged into one (i.e., removing duplicate words and sorting the remaining words in alphabetical order).

3 Enhanced JFC Attack

3.1 Accuracy of Information Extraction

The current JFC attack utilizes tessract project[2] to construct an OCR engine, which can extract text from the images. However, it is found that direct processing of the image did not yield accurate results. For example, a wide variety of colours that the text is simply not recognizable by the OCR engine.



Hey

This message content is top secret!

Your acquaintance...

[redacted]
"This email is confidential and may be privileged. If you are not the intended recipient, please delete it and notify us immediately. Please do not copy or use it for any purpose, or disclose its contents to any other person. Thank you."

Hey

This message content is top secret!

Your acquaintance...

[redacted]
"This email is confidential and may be privileged. If you are not the intended recipient, please delete it and notify us immediately. Please do not copy or use it for any purpose, or disclose its contents to any other person. Thank you."

```
Test email

This message content is top secret!

Vour acquaintance


"This email is conï¬ demm and may be privileged
\fynu are no! me mended recipient, please
derete u and nomy us |mmedialely Please do
not army or use u for any purpose, or died se ha
contents xo any other person Thank you.˜
```

Fig. 5. Information extraction by current JFC attack: converting an image to greyscale.

To mitigate this issue, Meng *et al.* [21] employed an approach of converting the images to greyscale before running it through the OCR engine, which yield a

[2] https://code.google.com/p/tesseract-ocr/.

better OCR output. An example is shown in Fig. 5. However, we identify that the resulting accuracy by this approach is still not high enough in practical usage. For example, the accuracy of OCR in Fig. 5 is about 89%, whereas some words cannot be recognized properly, i.e., "This email is confidential and may be privileged. If you are not the intended recipient, please delete it or notify us immediately" may be recognized as "This email is con..demm and may be privileged. fynu are no! me mended recipient, please derete u and nomy us mmedialely".

To investigate this issue, we conduct a study by validating the existing OCR technology using 10 videos collected from [21]. Each video is around 200 M and contains 700 - 900 frames. Figure 6 shows the extraction accuracy and the specific number of frames. It is found that the average accuracy is generally below 90% and would be not high enough in practice.

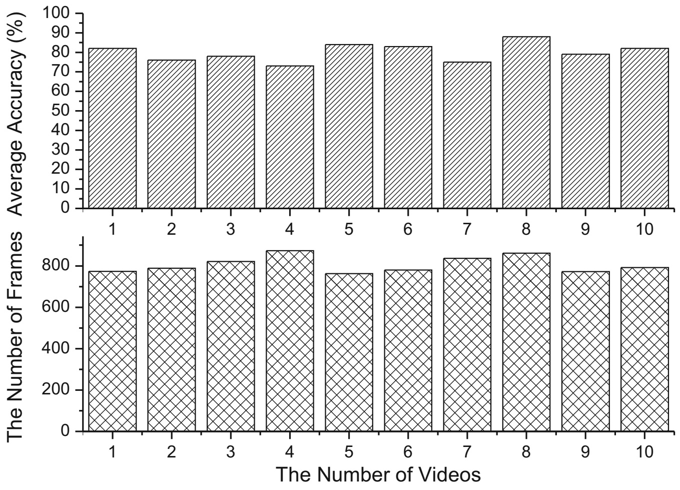


Fig. 6. The average accuracy and the number of frames.

3.2 Improvement for Information Extraction

Based on the results above, it is found how to accurately extract information from the collected videos remains a challenge for current JFC attack. To improve the attack effectiveness and the accuracy of information extraction, we adopt an engineering approach with three steps from image processing[3], including *image revision*, *image conversion* and *image clearance.*

- *Step1 - Image revision.* The main purpose of this step is to resize the image with variable height and width.

[3] https://github.com/tesseract-ocr/tesseract/wiki/ImproveQuality.

- *Step2 - Image conversion.* This step is similar to current JFC attack, which converts the image to grayscale format (black and white) before running it through the OCR engine.
- *Step3 - Image clearance.* This step aims to make an image clearer by removing the noise pixels.

```
Image Revision.
Input bitmap;
double nWidthFactor = (double) temp.Width / (double)newWidth;
double nHeightFactor = (double) temp.Height / (double)newHeight;
for (int x = 0; x < bmap.Width; ++x)
        for (int y = 0; y < bmap.Height; ++y)
do                  // Blue
                    bp1 = (byte)(nx * color1.B + fx * color2.B);
                    bp2 = (byte)(nx * color3.B + fx * color4.B);
                    nBlue = (byte)(ny * (double)(bp1) + fy * (double)(bp2));
                    // Green
                    bp1 = (byte)(nx * color1.G + fx * color2.G);
                    bp2 = (byte)(nx * color3.G + fx * color4.G);
                    nGreen = (byte)(ny * (double)(bp1) + fy * (double)(bp2));
                    // Red
                    bp1 = (byte)(nx * color1.R + fx * color2.R);
                    bp2 = (byte)(nx * color3.R + fx * color4.R);
                    nRed = (byte)(ny * (double)(bp1) + fy * (double)(bp2));

Image Clearance.
for (var x = 0; x < bmap.Width; x++)
         for (var y = 0; y < bmap.Height; y++)

do
                 var pixel = bmap.GetPixel(x, y);
                 if (pixel.R < 162 && pixel.G < 162 && pixel.B < 162)
                    bmap.SetPixel(x, y, Color.Black);
                 else if (pixel.R > 162 && pixel.G > 162 && pixel.B > 162)
                    bmap.SetPixel(x, y, Color.White);
```

Fig. 7. The code for image revision and clearance.

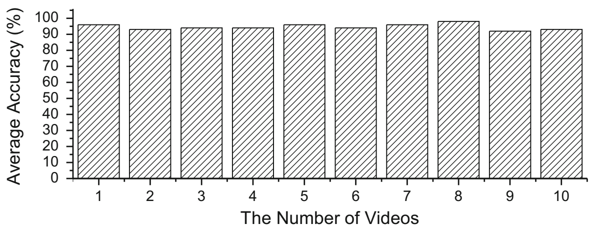


Fig. 8. The accuracy of extracting information using enhanced JFC attack.

The final goal of these steps is to better recognize textual information from an image. The code for image revision and image clearance can be referred to Fig. 7, while the code for image conversion is the same as exiting JFC attack (see details in [21]). To validate the performance, we process the same videos above by means of the three-step approach, and the results are depicted in Fig. 8. The experimental results show that the average accuracy can increase to above 90% (close to 95%) as compared to the results shown in Fig. 6, demonstrating that the three-step approach is effective to enhance JFC attack in extracting users' private information.

3.3 Information Correlation

The current JFC attack can merge all the text files from the same video into one text file by removing duplicate words and sorting the remaining words in alphabetical order, whereas there is no information correlation process to link data from different videos. In such case, information could be burst when collecting videos after a period of time. In practice, JFC chargers can be deployed with a cloud as shown in Fig. 9, where a centralized server can help collect the videos recorded by each charger and extracts the information from videos. Under this architecture, JFC attack has a potential to collect a large amount of private information from smartphone users.

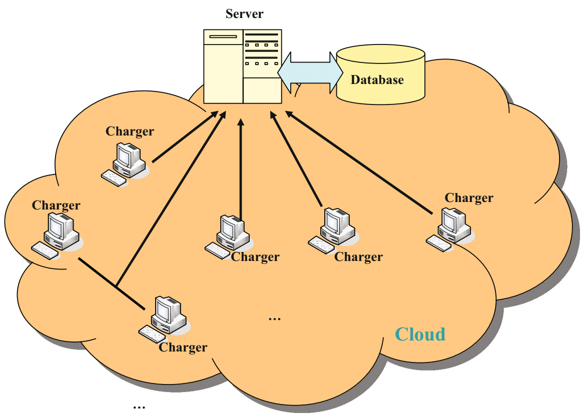


Fig. 9. Enhanced JFC attack with a cloud and centralized server.

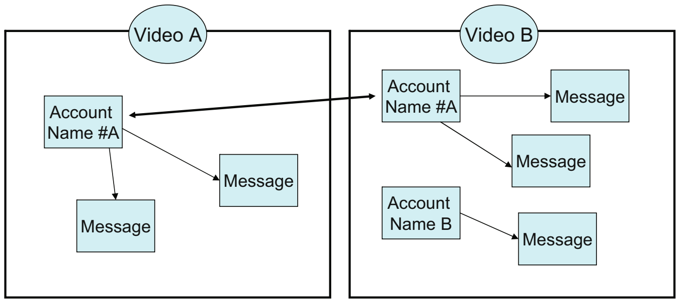


Fig. 10. An example of information correlation.

With the increasing number of videos, there is a need to correlate information from different videos and enhance the attack performance. To achieve this goal, we adopt a compact but efficient approach of indexing and linking information according to user's credentials, e.g., account names. An example is given in Fig. 10: if two identical account names are detected, then the corresponding information can be correlated (e.g., chatting history). The process of information correlation can be conducted in the back-end or in a cloud environment.

4 User Study

There are not many studies exploring the practical influence of JFC attacks in literature. In this work, we collaborated with an IT center (with over 250 personnel) in South China, and perform a user study to investigate the effectiveness of our advanced JFC attack in a practical environment.

Deployment. Before the study, we seek approval from the IT center to deploy up to 10 JFC chargers in one main dining & lobby room, where the JFC chargers can keep uploading the recorded videos to a cloud in the back-end. After uploading the videos, the chargers can delete the relevant videos to save disk space for new recorded videos and activate the attack for a long time. The centralized server has a maximum storage capacity of 100 T, where one-minute video usually requires 30 M space.

Table 1. Extracted user private information in the deployed environment.

User information	User information	User information
Android unlock pattern	PIN for iPhones	Gmail account and content
Other email account (e.g., 126, 163)	Other email content	Social networking account (e.g., Wechat, QQ)
Bank account	Social networking chat history (e.g., Wechat, QQ)	Visited website content
Email passwords (web-login)	Personal photos	Phone number list
Installed mobile applications	Settings	Bank message

Data Collection and Results. To protect users' privacy, we also seek users' approval and all data were handled by the IT center (we only collected statistics from the data). At least one IT administrator helped monitor the whole process and ensured all steps to comply with the relevant policies.

The JFC charger was implemented for two weeks excluding weekends (i.e., from Monday to Friday). The opening hours of the center are from 7am to 10pm; thus, we mainly recorded the information from 7am to 10pm. Figure 11

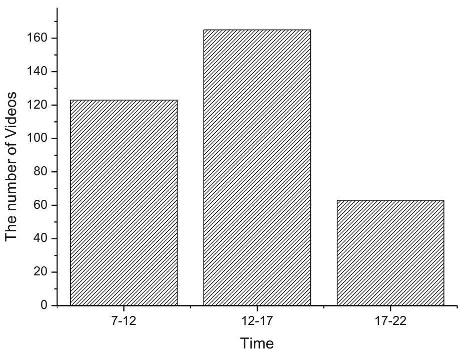


Fig. 11. The average number of collected videos for each day.

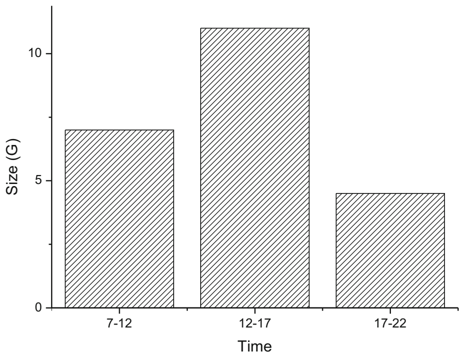


Fig. 12. The average size of collected videos for each day.

depicts the average number of recorded videos: the JFC chargers can collect more than 350 videos each day where the highest number of videos would be captured during the period of 12-17pm. This is because most people had their meals at that time and have the need to charge their phones in the room. From these collected videos, we can extract a large amount of private information about users, as summarized in Table 1, such as Android unlock pattern, PIN for iPhones, Email Account and content, social networking chat history, visited website, personal photos and so on.

Intuitively, each video has a different length and size, where a longer-time-frame video has a potential to leak more private information about a smartphone user. Figure 12 depicts the average size of collected videos for each environment. Generally, JFC chargers could collect around 22.5 G data each day. The accuracy of information extraction between the original and the advanced JFC attack is depicted in Fig. 13. It is found that the advanced attack could achieve an accuracy of 93% as compared to an accuracy of 82% achieved by the original attack setup. Overall, the advanced attack could provide a minimum accuracy over 88% while the original one could only reach a minimum accuracy of 75%.

In the study, we employ *hit rate* and *error rate* to measure the performance of JFC attack, which can be defined as below.

$$Hit\ rate = \frac{The\ number\ of\ correct\ correlation}{The\ total\ number\ of\ correlation} \tag{1}$$

$$Error\ rate = \frac{The\ number\ of\ incorrect\ correlation}{The\ total\ number\ of\ correlation} \tag{2}$$

It is revealed that the advanced JFC attack could achieve an average hit rate of 92% and an error rate of 3.3%. These errors are mainly caused by unclear texts due to inaccurate extraction. On the whole, our study validates that JFC attack can make a large impact on smartphone privacy and security. A large amount of private information could be identified through further enhancing JFC attack in the aspects of information extraction and correlation.

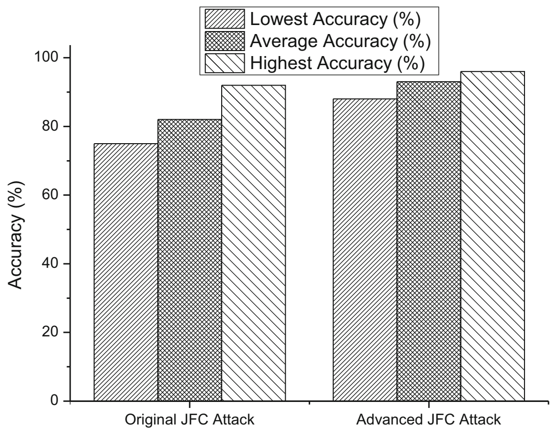


Fig. 13. The accuracy between the original and the advanced JFC attack.

5 Further Discussion

This work presents an advanced JFC attack by improving its performance in accurately extracting and correlating users' private information. In the study, we show that the JFC attack can capture phone screen in the back-end and threaten smartphone privacy. After launching JFC attack after a period of time, a large amount of data can be recorded to dig out users' private information. Our user study in a practical environment demonstrates that JFC attack has a potential to cause a large number of victims. It is worth noting that this attack can be further enhanced by integrating more advanced techniques in the aspects of both extracting and correlating secrets from videos.

Defence. The root cause of JFC attack is that Android OS and iOS devices allow screen mirroring without granting explicit permissions. To defend against this kind of attacks, there is a need to deploy additional security mechanisms and increase user awareness.

– *Disabling automate screening.* This is the most direct and easiest way to defend against JFC attack, by disabling automatic screening function. For example, the phones can disable automatically outputting screen information, and ask users to choose such function when they have the need. However, disabling screen output by default is effective, but may decrease usability, i.e., when there is a need to display phone screen for many times, users have to click the display button again and again. In this case, an alternative is to make notifications and ask for permission. Recall that when connecting iPhones to a computer, the phone will prompt a notification asking whether the user trusts the computer or not. The smartphone could do the same action to warn users before the display of phone screen. This strategy can increase user awareness especially for novice, but can provide much flexibility for advanced users.

– *Securing Charger.* To protect data leakage against JFC attack, one potential solution is to use a safe charger such as USB Condom [31]. This USB is able to prevent accidental data exchange when the device is plugged into another device with a USB cable, through cutting off the data pins in the USB cable and only allowing the power pins to connect in practice. However, this solution does not work for particular charging attacks like PowerSnitch [29], which can leak information via analyzing power consumption.
– *Employing biometrics.* It is feasible to reduce the impact of JFC attack by integrating biometrics, since JFC attack is unable to capture these secrets without a screen-input. For example, behavioral biometrics can be added to the process of inputting PIN code and unlock patterns (i.e., building a fingerprint-based unlocking mechanism). Several behavioral-based authentication methods can be referred to [5–7,12,16,17,19,27].
– *Educating users.* Until particular patches or control policies are updated by vendors, JFC attacks are difficult to defend by current security mechanisms (e.g., [18]). Therefore, similar to other area like spam detection [13], user education is a necessary action to raise users awareness and attention.

6 Related Work

As smartphones have become a major target for cyber-criminals, privacy leakage is a big concern for smartphone users. There is a line of research and practical studies on how to infer mobile users' private information and data through malware, side channels and physical access attacks.

Smartphone malware. Malicious applications are a big threat on smartphones [4]. Lin *et al.* [14] found that the ADB capability could be exposed to any party with the INTERNET permission on the same device. They then built *Screenmilker*, an application that can monitor the screen and pick up a user's password when the user is typing in real-time. Xing *et al.* [32] evaluated the Android updating mechanism and found Pileup flaws, through which a malicious application could strategically declare a set of privileges and attributes on a low-version operating system, and wait until it is able to escalate its privileges on the new system. By exploiting the Pileup vulnerabilities, their application can not only acquire a set of newly added system and signature permissions, but also determine their settings. Andriesse and Bos [1] introduced a code hiding approach for trigger-based malware, which can conceal malicious code inside spurious code fragments. A summary of malware research can be referred to a survey [25].

Accelerometer side channel. Most popular malware utilized side channel to steal information on mobile devices. Cai and Chen [3] presented a side channel on touchscreen smartphones with only soft keyboards. They identified that when users clicked on the soft keyboard, especially when he/she holds the phone by hand rather than placing it on a fixed surface, the phone vibration on touchscreens are highly correlated to the keys being typed. They conducted a study and showed that they were able to infer correctly more than 70% of the keys typed on a number-only soft keyboard on a phone. Marquardt *et al.* [15] also demonstrated that an application with access to accelerometer readings on a modern mobile phone can use side channel to recover text entered on a nearby keyboard. They showed that by characterizing consecutive pairs of keypress events, up to 80% of typed content can be recovered. Schlegel *et al.* [28] designed *Soundcomber*, a stealthy Trojan with innocuous permissions that can sense the context of its audible surroundings to automatically extract a small amount of targeted private information such as credit card and PIN numbers from both tone- and speech-based interaction with phone menu systems.

Han *et al.* [8] presented that accelerometer readings can be used to infer the trajectory and starting point of an individual who is driving, and pointed out that current smartphone operating systems allow any application to observe accelerometer readings without requiring special privileges. Thus, accelerometers can be used to locate a device owner within a 200 m radius of the true location. Owusu *et al.* [23] described how a background application can use the accelerometer as a side channel to spy on keystroke information during sensitive operations, e.g., account login. They could successfully break 59 out of 99 passwords by using only accelerometer measurements logged during text entry. Miluzzo *et al.* [22] presented *TapPrints*, a framework for inferring the location of taps on touchscreens using motion sensor data with up to 90% and 80% accuracy.

Physical side channel. These attacks are mainly based on physical objects, like oily residues left on the touchscreen or the screen reflection from nearby objects. Aviv *et al.* [2] first explored the feasibility of smudge attacks on touchscreens with different lighting angles and light sources. They indicated that the pattern could be partially identifiable in 92% and fully in 68% of the tested lighting and camera setups. Zhang *et al.* [33] presented a fingerprint attack against tapped passwords via a keypad, which could reveal more than 50% of the passwords. For the screen reflection, Raguram *et al.* [26] showed that automatic reconstruction of text typed on a mobile device's virtual keyboard is feasible via compromising reflections, i.e., those of the phone in the user's sunglasses. By means of the footage captured in realistic environments (e.g., on a bus), their approach could reconstruct fluent translations of recorded data in almost all of the test cases.

Charging attacks. To our knowledge, Lau *et al.* [11] designed an early charging attack named *Mactans*. They particularly deployed a malicious charger using BeagleBoard to conduct malware injection on smartphones during the charging period. However, their attacks require users to unlock the phone screen and install developer licenses in advance. Spolaor *et al.* [29] described how an adversary could leverage a malicious charging station to exfiltrate smartphone data via a USB charging cable using power consumption. They designed *PowerSnitch*, an application that could send out bits of data in the form of power bursts by manipulating the power consumption of the device's CPU. One limitation of this attack is that users have to pre-install a small application on their phones.

Meng *et al.* [20,21] developed JFC attacks, which can record screen information during the whole charing period, without the need to ask for any permission or phone unlock action. It is worth noting that any current anti-malware software is not aware of such threat. To launch this attack, an additional hardware of VGA/USB interface is needed, which is not hard to obtain online. Thus, charging attacks are highly deployable in real scenarios.

7 Conclusion

As compared with malicious applications (malware), charging threats are often ignored by the literature. With the increasing adoption of public charging stations, we argue that charging attacks may become a big concern for users' privacy. Juice filming charging (JFC) attack is one specific kind of charging attacks, which can steal users' private data from both Android OS and iOS devices, through automatically monitoring and recording screen information during the charging period.

In real-world deployment, we identify that information extraction and correlation are still challenges for current JFC attack. To investigate the practical influence of this attack, in this work, we focus on JFC attack and try to enhance its performance in the aspects of extracting and correlating textual information from the captured videos. In the evaluation, we conduct a user study with an IT center. The results demonstrate that the enhanced JFC attack can collect users' information at large and extract private data with a higher accuracy (i.e., over 90%) than the original one. Our work validates that JFC attack may cause a large number of victims, which should be given more attention.

Acknowledgments. We would like to thank all anonymous reviewers for their helpful comments in improving the paper. This work was partially supported by National Natural Science Foundation of China (61602396).

References

1. Andriesse, D., Bos, H.: Instruction-level steganography for covert trigger-based malware. In: Dietrich, S. (ed.) DIMVA 2014. LNCS, vol. 8550, pp. 41–50. Springer, Cham (2014). doi:10.1007/978-3-319-08509-8_3
2. Aviv, A.J., Gibson, K., Mossop, E., Blaze, M., Smith, J.M.: Smudge attacks on smartphone touch screens. In: Proceedings of the 4th USENIX Conference on Offensive Technologies (WOOT), pp. 1–7 (2010)
3. Cai, L., Chen, H.: TouchLogger: inferring keystrokes on touch screen from smartphone motion. In: Proceedings of the 6th USENIX Conference on Hot Topics in Security (HotSec), pp. 1–6 (2011)
4. Dagon, D., Martin, T., Starner, T.: Mobile phones as computing devices: the viruses are coming! IEEE Pervasive Comput. **3**(4), 11–15 (2004)
5. De Luca, A., Hang, A., Brudy, F., Lindner, C., Hussmann, H.: Touch me once and i know its you!: implicit authentication based on touch screen patterns. In: Proceedings of the 2012 ACM Annual Conference on Human Factors in Computing Systems (CHI), pp. 987–996 (2012)

6. Feng, T., Liu, Z., Kwon, K.-A., Shi, W., Carbunary, B., Jiang, Y., Nguyen, N.: Continuous mobile authentication using touchscreen gestures. In: Proceedings the 2012 IEEE Conference on Technologies for Homeland Security (HST), pp. 451–456 (2012)
7. Frank, M., Biedert, R., Ma, E., Martinovic, I., Song, D.: Touchalytics: on the applicability of touchscreen input as a behavioral biometric for continuous authentication. IEEE Trans. Inf. Forensics Secur. **8**(1), 136–148 (2013)
8. Han, J., Owusu, E., Nguyen, L., Perrig, A., Zhang, J.: ACComplice: location inference using accelerometers on smartphones. In: Proceedings of the 4th International Conference on Communication Systems and Networks (COMSNETS), pp. 1–9, New York, NY, USA (2012)
9. IDC, Smartphone Momentum Still Evident with Shipments Expected to Reach 1.2 Billion in 2014 and Growing 23.1% Over (2013), http://www.idc.com/getdoc.jsp?containerId=prUS24857114
10. Juice Jacking Vulnerability for iOS, https://www.infotransec.com/news/juice-jacking-vulnerability-ios
11. Lau, B., Jang, Y., Song, C.: Mactans: Injecting malware into iOS devices via malicious chargers. Blackhat, USA (2013)
12. Li, L., Zhao, X., Xue, G.: Unobservable Re-authentication for Smartphones. In: Proceedings of the 20th Annual Network and Distributed System Security Symposium (NDSS), pp. 1–16 (2013)
13. Li, W., Meng, W.: An empirical study on email classification using supervised machine learning in real environments. In: Proceddings of the 2015 IEEE International Conference on Communications (ICC), pp. 7438–7443. IEEE (2015)
14. Lin, C.-C., Li, H., Zhou, X., Wang, X.: Screenmilker: how to milk your android screen for secrets. In: Proceedings of Annual Network and Distributed System Security Symposium (NDSS), pp. 1–10 (2014)
15. Marquardt, P., Verma, A., Carter, H., Traynor, P.: (sp)iPhone: decoding vibrations from nearby keyboards using mobile phone accelerometers. In: Proceedings of ACM Conference on Computer and Communications Security (CCS), pp. 551–562. ACM, New York (2011)
16. Meng, Y., Wong, D.S., Schlegel, R., Kwok, L.: Touch gestures based biometric authentication scheme for touchscreen mobile phones. In: Kutyłowski, M., Yung, M. (eds.) Inscrypt 2012. LNCS, vol. 7763, pp. 331–350. Springer, Heidelberg (2013). doi:10.1007/978-3-642-38519-3_21
17. Meng, Y., Li, W., Kwok, L.-F.: Enhancing click-draw based graphical passwords using multi-touch on mobile phones. In: Janczewski, L.J., Wolfe, H.B., Shenoi, S. (eds.) SEC 2013. IAICT, vol. 405, pp. 55–68. Springer, Heidelberg (2013). doi:10.1007/978-3-642-39218-4_5
18. Meng, W., Li, W., Kwok, L.F.: EFM: enhancing the performance of signature-based network intrusion detection systems using enhanced filter mechanism. Comput. Secur. **43**, 189–204 (2014)
19. Meng, W., Wong, D.S., Furnell, S., Zhou, J.: Surveying the development of biometric user authentication on mobile phones. IEEE Commun. Surv. Tutorials **17**, 1268–1293 (2015)
20. Meng, W., Lee, W.H., Murali, S.R., Krishnan, S.P.T.: Charging me and i know your secrets! towards juice filming attacks on smartphones. In: Proceedings of the Cyber-Physical System Security Workshop (CPSS), in Conjunction with AsiaCCS 2015. ACM (2015)

21. Meng, W., Lee, W.H., Murali, S.R., Krishnan, S.P.T.: JuiceCaster: towards automatic juice filming attacks on smartphones. J. Netw. Comput. Appl. **68**, 201–212 (2016)
22. Miluzzo, E., Varshavsky, A., Balakrishnan, S., Choudhury, R.R.: TapPrints: your finger taps have fingerprints. In: Proceedings of MobiSys, New York, NY, USA , pp. 323–336 (2012)
23. Owusu, E., Han, J., Das, S., Perrig, A., Zhang, J.: ACCessory: password inference using accelerometers on smartphones. In: Proceedings of the 12th Workshop on Mobile Computing Systems & Applications (HotMobile), pp. 1–6. ACM, New York (2012)
24. Ossmann, M., Osborn, K.: Multiplexed Wired Attack Surfaces. Black Hat USA (2013), https://media.blackhat.com/us-13/US-13-Ossmann-Multiplexed-Wired-Attack-Surfaces-WP.pdf
25. Peng, S., Yu, S., Yang, A.: Smartphone malware and its propagation modeling: a survey. IEEE Commun. Surv. Tutorials **16**(2), 925–941 (2014)
26. Raguram, R., White, A.M., Goswami, D., Monrose, F., Frahm, J.-M.: iSpy: automatic reconstruction of typed input from compromising reflections. In: Proceedings of the 18th ACM Conference on Computer and Communications Security (CCS), pp. 527–536, ACM, New York (2011)
27. Sae-Bae, N., Memon, N., Isbister, K., Ahmed, K.: Multitouch gesture-based authentication. IEEE Trans. Inf. Forensics Secur. **9**(4), 568–582 (2014)
28. Schlegel, R., Zhang, K., Zhou, X., Intwala, M., Kapadia, A., Wang, X.: Soundcomber: a stealthy and context-aware sound Trojan for smartphones. In: Proceedings of the 18th Annual Network and Distributed System Security Symposium (NDSS), San Diego, CA, USA, pp. 17–33 (2011)
29. Spolaor, R., Abudahi, L., Moonsamy, V., Conti, M., Poovendran, R.: No free charge theorem: a covert channel viaUSB charging cable on mobile devices. In: Proceedings of the 15th International Conference on Applied Cryptography and Network Security (ACNS), pp. 84–102 (2017)
30. Singapore Power to provide 200 free mobile phone charging stations for SG50 (2015), http://www.straitstimes.com/singapore/singapore-power-to-provide-200-free-mobile-phone-charging-stations-for-sg50
31. The Original USB Condom, http://int3.cc/products/usbcondoms
32. Xing, L., Pan, X., Wang, R., Yuan, K., Wang, X.: Upgrading your android, elevating my malware: privilege escalation through mobile OS updating. In: Proceedings of the 2014 IEEE Symposium on Security and Privacy, Berkeley, CA, USA, pp. 393–408 (2014)
33. Zhang, Y., Xia, P., Luo, J., Ling, Z., Liu, B., Fu, X.: Fingerprint attack against touch-enabled devices. In: Proceedings of the 2nd ACM Workshop on Security and Privacy in Smartphones and Mobile Devices (SPSM), pp. 57–68. ACM, New York (2012)