Charging Me and I Know Your Secrets!  
Towards Juice Filming Attacks on Smartphones

Infocomm Security Department  
Institute for Infocomm Research, Singapore  
{mengw, whlee, muralism, krishnan}@i2r.a-star.edu.sg

ABSTRACT

Smartphones such as Android phones and iPhones are widely adopted worldwide and the number of smartphone users has increased dramatically. With the popularity of smartphones, the privacy of smartphone users are challenged by various malware and attacks. A large amount of malware has been developed and spread such as Soundcomber, Screenmiller, TouchLogger, and so on. In addition, some smartphone attacks are based on physical access to the phone like smudge attacks. All these have reminded current users to pay more attention for smartphone security such as installing anti-malware software to defend against malicious apps, which may reduce the effectiveness of most malware.

In this paper, we identify a vulnerability of smartphone charging and design a new type of charging attacks (called juice filming attacks) based on a standard USB connector and HDMI, which can steal users’ secrets through automatically video-capturing their inputs (e.g., unlock patterns, PIN code). The efficiency of our designed attacks relies on the observations that users are not aware of any risk when charging their phones in public places and that most users would interact with their phone during the charging. Different from other malware and attacks, our designed juice filming attacks possess six major features: 1) be simple but quite efficient; 2) user unawareness; 3) does not need to install any apps on phone’s side; 4) does not need to ask for any permissions; 5) cannot be detected by any current anti-malware software; 6) can be scalable and effective in both Android OS and iOS. To implement this attack, we employ a VGA /RGB frame grabber and further conduct several user studies to explore the feasibility and effectiveness of our attacks. Based on the understanding, we later propose several mitigation strategies to defend against this kind of attacks. We hope this effort to raise the awareness of charging attacks that may leak users’ private data.

Categories and Subject Descriptors  
C.2.0 [Computer-Communication Networks]: General—

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Keywords  
Mobile Security and Vulnerabilities, Android and iOS Security, Video Recording, Charging Attacks, Human Factors

1. INTRODUCTION  

The ubiquity of powerful personal computing devices has changed our lives, e.g., how we communicate with others and where to store private information, messages and data. Particularly, smartphones such as Android phones and iPhones have become omnipresent and are used widely by millions of people to send text messages and emails, surf the internet, and purchase online. According to the forecast from the International Data Corporation (IDC), worldwide smartphone shipments will reach a total of 1.2 billion units in 2014, resulting in a 23.1% increase from the 1.0 billion units shipped in 2013, where Android and iOS share a total of 95% of the mobile market [9].

1.1 Motivations  

Due to its popularity, smartphones have become an attractive target for hackers and malware [4]. In security community, a lot of malware has been studied to steal users’ private information and data. For example, Xu et al. [30] explored a vulnerability of 3G smartphones and proposed a video-based spyware, called Stealthy Video Capturer (SVC). They further implemented the spyware and conducted experiments on real world 3G smartphones. Their results showed that the spyware can capture private video information with unremarkable power consumption, CPU and memory occupancy. Schlegel et al. [26] presented Soundcomber, a Trojan with few and innocuous permissions, which is able to extract a small amount of targeted private information from the audio sensor of a smartphone. This Trojan can intelligently infer sensitive data such as credit card and PIN numbers from both tone- and speech-based interaction with phone menu systems. Meanwhile, Cai and Chen [3] developed TouchLogger, a smartphone application that extracts features from the device orientation data to infer the keys being typed. It is based on the observation that typing different keys may cause different motions of the smartphone, due to the locations of keys on a soft keyboard. Recently, Lin et al. [14]
designed Screenmilk, a malware application that can detect the right moment to monitor the screen and pick up a user's password when he or she is typing in real time, by exposing the ADB capability to any party with the INTERNET permission on the same device.

In addition, some smartphone attacks are based on the physical access to the devices. For instance, Aviv et al. [2] studied smudge attacks, which an attacker attempts to extract sensitive information about recent user input by inspection of smudges on touchscreens. They showed that partial or complete patterns are easily retrieved. Then, Zhang et al. [31] presented a fingerprint attack against touch-enabled devices. More specifically, an attacker first dusts the surface of devices with appropriate fingerprint powder to reveal the detail of the fingerprints on the touch screen and then uses camera to carefully take two photographs of fingerprints. Later, the attacker should sharpen the fingerprints in an image via various image processing techniques and map fingerprints to a keypad to infer tapped passwords.

As the types of mobile malware are becoming large and diverse, many users are aware of this threat and would take countermeasures. For instance, users may install anti-malware software to protect their mobile phones. It is noted that installing such software cannot fully safeguard the phone, but can reduce the effectiveness of most malware through various analysis (i.e., analyzing permissions and battery usage). On the other hand, it is a bit difficult for launching smudge attacks, because an attacker should get the phone first. Most current studies of privacy leakage on smartphones are focusing on malware, accelerometer and physical side channel attacks, however, there are several limitations and open problems of these attacks as below:

- **Scalability.** Most malware is only workable on either Android or iOS.
- **Permission requirement.** Most malware needs to request more or less key permissions from smartphones, which may be detected by anti-malware techniques.
- **Complexity.** Some malware as well as accelerometer side channel attacks require more resources on phones' side (e.g., power), which may increase user awareness.
- **Deployment.** Malware and accelerometer side channel attacks need to be installed in smartphones, which may be detected by anti-malware software.

In this case, one question comes to our mind is whether there are other means to leak users privacy that can ease the limitations above. Motivated by [12], we identify that charging attacks are missed by literature and can be a big threat to smartphone privacy without user awareness. In this work, our motivation is thus to infer user private information by means of charging attacks.

### 1.2 Our Contributions

It is noted that user awareness is an important factor to affect the performance of malware on smartphones. Thus the successful rate of an attack would be greatly increased if users give less or no attention. In this paper, we design a new charging attack (called *juice filming attack*), which is able to steal users' information by automatically taking videos, when they are interacting with the phones during the charging process, without causing user awareness.

![Figure 1: The pop-up notification when connecting an iPhone to a computer.](image)

It is worth noting that our attack is different from the *juice jacking attacks* in Blackhat community, in which once an unlocked iOS device is plugged into a malicious device, it is presented with access to a significant amount of personal data and without the user’s permission and can install hidden malicious software on the device [10, 12]. The major difference here is that our attack does not need users to unlock the phone and ask for any permissions, whereas the *juice jacking attacks* require users to trust the computer at least (see Figure 1). To distinguish our work, we thus tune the name of our attack to *juice filming attack*.

Moreover, our attack is also different from Screenmilk in that *juice filming attack* does not need to install any app and ask for any permissions. Actually, these are two different categories of attacks where our charging attacks require more of a hardware-based setup while malware-based threats need more resources from software. Overall, our designed *juice filming attack* can offer six desirable features (see details in Section 2): 1) be easy to implement but quite efficient; 2) less user awareness; 3) does not need to install any additional apps or components on phones; 4) does not need to ask for any permissions; 5) cannot be detected by any current anti-malware software; 6) can be scalable and effective in both Android OS and iOS. To the best of our knowledge, our attack is the first one, which can provide the features above together. The contributions of this work can be summarized as below.

- We design a new charging attack, called *juice filming attack*, which can automatically video-capture user inputs during the charging. Based on the recorded video, it is easy to manually recover users’ secrets such as PIN code, unlock patterns, email accounts and passwords, etc. We then describe the threat model, adversary scenarios and attack design in detail.
- We further conduct several user studies to explore the feasibility and effectiveness of this attack. It is found that *juice filming attack* is effective in inferring users' private information and data in real scenarios. We later discuss some challenges and propose mitigation strategies regarding this attack.

The reminder of the paper is organized as follows. Section 2 introduces our proposed *juice filming attacks* in detail.
Section 3 describes several studies of user behaviors and user awareness when charging their phones and Section 4 discusses some challenges and mitigation strategies regarding our attack. Section 5 introduces prior related works about malware and attacks on mobile phones. Finally, Section 6 concludes our work with future directions.

2. CHARGING AND JUICE FILMING ATTACKS

In this section, we introduce the threat model, malicious charging scenarios and the design of juice filming attacks.

2.1 Threat Model

Mobile phones are very common nowadays and so are portable mobile chargers too. In fact they are used to charge the batteries inside the mobiles by providing them a low level DC voltage and current. According to the usage of smartphones, people usually charge their phones differently. For users who need to intensively use their phones (e.g., businessman, manager), they may need to charge their phones several times a day. In other words, phone charging is a basic and common demand for smartphone users.

Based on this demand, in this work, we assume that most users would not treat chargers as highly sensitive or dangerous. This assumption is not a wishful thinking since we have seen lots of people using public chargers in many places such as airports, subways, etc. The details can be seen in our study in Section 3. Furthermore, charging attacks can be divided into two types, public and private. A public charging attack is mainly based on a public charger such as chargers provided in airports, while a private charging attack utilizes a private charger from friends or others.

More specifically, the public charging attack usually has a wider impact as the charger would be deployed in public places, where many people have the need to charge their phones. In these cases, people are much likely to interact with their phones during the charging. In contrast, the private charging attack mainly utilizes a private charger, which focuses on a smaller range of targets. However, the private charger may obtain a higher credence from people than the public chargers, so it is possible to discover more sensitive information through social networking (i.e., asking people to show something). Both charging attacks can be effective to collect users’ sensitive and private information.

2.2 Common Adversary Scenarios

There is seldom work in literature researching into charging attacks, however, it is identified that these attacks are feasible in many real scenarios. In this section, we mainly describe three common scenarios in practice, which can be turned to adversary and be used to launch charging attacks, including public charging station, semi-public charging station and charger borrowed from others. We describe each scenario as below.

Public Charging Stations. With the increasing demand of phone charging, many public places like airports and subways starting to provide charging stations for individuals. As shown in Figure 2, we present two cases of public charging stations. The charger in Figure 2 (a) is located in airports while the other in Figure 2 (b) is deployed in subways.

It is seen that the charger in the airports (Figure 2 (a)) only provides an interface to users who want to charge their phones. This scenario opens a hole for attackers to explore. Imagining that if the charger can record the inputs when users are interacting with their phones, the users’ privacy would be challenged. In addition, it is hard to check whether the charger is safe since only an interface is provided.

Semi-Public Charging Stations. The term of ‘semi-public’ here mainly refers to a hotel or place that cannot be fully controlled by individuals. Taking hotels as an example, before moving in, it is impossible for users to control and inspect the room. Only after checking in, the room can be used during the stay and can be treated as a private place. In this case, we call this scenario as ‘semi-public’. If some ones (e.g., clean workers) prepare our attack in advance, all living guests will become victims.

Borrowed Charger. In this scenario, an individual may ask for charging their phones to others computers or PCs. The ‘others’ here may refer to a well-known friend, a known colleague or only a unfamiliar person. For example, when you are visiting your friends, you may charge your phone to his/her computer. Similarly, in a company environment, you may charge the phone to the computer of your colleagues or others. Because the phone will not work without charging, there is no much choices under this scenario.

This concrete scenario also opens a chance for attackers who can control the charger and lend to victims. As long as the victims use the phone during the charging, their personal information would be leaked out.

The scenarios above are three specific cases to demonstrate the potential threat of charging attacks, and it is easy to identify other similar scenarios. With the understanding of adversary model and scenarios, we introduce how to launch this kind of attacks in the next section.

2.3 Attack Design

We identify that most traditional charging attacks require the phone to be unlocked at least once [10, 12]. This leads us to consider whether it is possible to launch charging attacks without asking for any permission and phone unlock. In this section, we thus develop juice filming attacks, which are a type of charging attacks but without the need to request for phone unlock and any permissions. That is, whether users trust or distrust the computer does not affect our attacks.

Basic idea. Lau et al. [12] recently presented Mactans, a malicious charger using BeagleBoard to launch malware injection attacks during the charging. However, their attack requires the users to unlock the phone and install developer
licenses in advance.

Motivated by their work, our objective is to explore whether it is possible to launch charging attacks without requesting any permissions from the phones. In practice, we notice that no permissions will be asked when plugging the latest iOS or Android phones to a projector, while the projector can display the phone screen. Moreover, there are no compelling and visual indicators on the screen when the device (e.g., iOS) is being plugged. Based on these observations, we develop juice filming attacks, which can automatically video-record user inputs during charging by using VGA/USB interface. Our attack indicates that the display can be leaked through a standard micro USB connector through the Mobile High-Definition Link (MHL) standard. The high-level architecture of the juice filming attack is depicted in Figure 3. It is seen that a VGA/USB interface is connected to a computer forming a malicious charger, and the computer then can automatically record user inputs when connecting the phone to the charger.

**Set up.** To implement this attack, it is important to choose an appropriate VGA/USB interface. As a study, in this work, we employ a hardware interface called VGA2USB from Epiphan system Inc.\(^1\) The VGA2USB is a full-featured VGA/RGB frame grabber, which can send a digitized video signal from VGA to USB. In Figure 4 (a), we present a full and explicit implementation of our attack by means of VGA2USB. To hide the VGA2USB, it is easily cover it with other papers or in a sealed box. As shown in Figure 4 (b), we present an implicit implementation by hiding the VGA2USB with some papers so that users may be not aware of it.

To better explain the attack, we present an example in Figure 5. It is seen that the iPhone screen can be video-captured by the computer (note that we should first install the VGA2USB drivers in the computer). In this case, all the users’ inputs such as typed passwords, PIN code, email address and used application types can be recorded and extracted by manually analyzing the captured video. In addition, in real deployment, the computer can be replaced by a small hardware such as Raspberry Pi\(^2\) to further reduce user awareness and increase the flexibility of deployment.

**Features.** As compared to the most existing smartphone malware and attacks, our proposed juice filming attack has six features (or advantages).

- **Be easy to implement but quite efficient.** By deploying the hardware and its driver with a computer, our attack can be launched without other requirements on the phone’s side. In addition, our attack can capture all the users’ inputs, while Screenmilker [14] can only capture part of the inputs. Therefore, our attack is much stronger than most other malware in the aspects of implementation and secret capturing.

- **Does not need to install any additional apps or components on phones.** Different from other malware such as Screenmilker [14], SVC [30], TouchLogger [3], our attack does not need to install any additional applications or components on smartphones in advance, which can greatly reduce the risk of being detected.

- **Does not need to ask for any permissions.** (1) As our attack does not install any apps on phones, it does not need to ask for any permissions from users. As for the other malware like Soundcomber [26], permissions (or innocuous permissions) are still required, which can be detected by well-configured anti-malware software. (2) Moreover, for other charging attacks like Mactans [12], they still need the phone to be unlocked during the attack (see Figure 1, users need to trust the computer). In contrast, our attack does not require the phone unlock, that is, our attack is effective regardless whether users trust the computer or not. As a result, our attack is considered to be more powerful in stealing users’ sensitive information.

- **Less user awareness.** Since our attack does not need to install any additional apps on phones and request any permissions from users, our attack receive much less awareness than other smartphone malware. The

---

\(^1\)http://www.epiphan.com/products/vga2usb/.

Figure 5: The captured inputs from an iPhone.

Less user awareness makes our attack be much more effective in real scenarios, and we conduct several user studies regarding this issue in Section 3.

• Cannot be detected by any current anti-malware software. Due to the features above, it is extremely hard, or even impossible, for current anti-malware software to detect our attack. In contrast, for malware such as Screenmilker [14], Soundcomber [26], and TouchLogger [3], a well-configured anti-malware engine can identify them through various analysis techniques.

• Can be scalable and effective in both Android OS and iOS. It is noticed that most current smartphone malware and attacks are only applicable on one of the major operating systems. For example, Screenmilker [14] and Soundcomber [26] are only feasible on Android phones. Differently, our attack can be effective in both Android phones and iPhones, which is more powerful in real-world implementation.

3. STUDY AND EVALUATION

In this section, we present several studies regarding the interaction between users and phones in public places, user awareness on charging attacks, and the effect of our attack on compromising users’ privacy.

3.1 User Interaction

In this study, we aim to investigate the behaviors of different users on how they interact with their phones in an airport (public place) through direct observation.

Study details. This study was taken in an airport (one terminal), where multi-media carts are provided to passengers. These carts can provide multiple functions such as phone charging, flight information checking, catering recommendation, moving watching, etc.

For the one-week study, we randomly select 30 passengers each day who are managing a cart and all the information is collected through direct observation. For privacy reasons, we keep 6-7 meters away from the target passenger so that we only record following information: gender, phone charging duration and interaction duration. After collection, we will confirm with the passengers and ensure that they approve us to keep the data for research.

Table 1: Study results of charging duration (CD) and interaction duration (ID).

<table>
<thead>
<tr>
<th>Gender</th>
<th>No Charging</th>
<th>CD: [1 min, 3 min]</th>
<th>CD &gt; 3 min</th>
<th>No Interaction</th>
<th>ID: [1 min, 3 min]</th>
<th>ID &gt; 3 min</th>
</tr>
</thead>
<tbody>
<tr>
<td>Male #</td>
<td>14</td>
<td>21</td>
<td>70</td>
<td>9</td>
<td>20</td>
<td>90</td>
</tr>
<tr>
<td>Female #</td>
<td>5</td>
<td>12</td>
<td>88</td>
<td>1</td>
<td>20</td>
<td>90</td>
</tr>
</tbody>
</table>

Table 2: Information of participants in the study on user awareness.

<table>
<thead>
<tr>
<th>Age Range</th>
<th>Male</th>
<th>Female</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt; 25</td>
<td>46</td>
<td>40</td>
</tr>
<tr>
<td>25-35</td>
<td>33</td>
<td>35</td>
</tr>
<tr>
<td>35-45</td>
<td>14</td>
<td>16</td>
</tr>
<tr>
<td>&gt; 45</td>
<td>10</td>
<td>6</td>
</tr>
</tbody>
</table>

Study results. We totally observe up to 210 passengers during a week. In Table 1, we present the results regarding how the passengers may interact with their phones in a public place with a public charging station.

It is noticeable that only 14 males and 5 females are not charging their phones after getting a multi-media cart, while the other 191 passengers choose to charge their phones using the carts. It is also found that 33 passengers charged their phones in a time duration between 1 minute and 3 minutes. The remaining passengers charged their phones more than 3 minutes. For the interaction duration (with 191 passengers), it is found that only 10 males and 1 female did not use the phone during the charging, but up to 133 passengers were interacting with their phones for more than 3 minutes.

The results on charging duration and interaction duration are very encouraging since most people are willing to charge their phones using a public charger and will interact with their phones during the charging process. In addition, we find that more female passengers would like to play with the phones during the charging than the male passengers. The study presents that user behavior in real scenarios opens a hole for our attack. For example, under our attack, if a user logs in to email account or bank account during the charging, we can collect all these sensitive information.

3.2 User Awareness on Charging Attacks

In this study, our goal is to investigate user awareness and attitude on charging attacks and public charging stations. In total, we collected the feedback from 200 participants and the information of participants is shown in Table 2. All participants are volunteers and aged from 17 to 48. In addition,
126 of them are students (not from security related majors), while the others are researchers, businessmen, officers and senior people.

For each participant, we present them a form with a set of questions. Some interesting questions and the corresponding answers are shown in Table 3. It is seen that 132 participants have the need to charge their phones outside and 145 of them are willing to use a public charging station. There are up to 156 participants who mention that they would interact with their phones during the charging. Most participants indicate that they will surf Internet and check social networking sites. In the study, it is found that 179 participants do not have any security concerns regarding the phone chargers in public places, while only 9 participants know or hear charging attacks before. On a more positive note, there are 132 participants out of 145 who are willing to use a public charging station. There are up to 156 participants who mention that they would interact with their phones during the charging. Most participants indicate that they will surf Internet and check social networking sites. In the study, it is found that 179 participants do not have any security concerns regarding the phone chargers in public places, while only 9 participants know or hear charging attacks before. On a more positive note, there are 132 participants out of 145 who are willing to use a public charging station. However, most of them do trust the public charging stations and will interact with their phones. Therefore, juice filming attacks have large potential victims in public places, so that they will install anti-malware software on smartphones, so that they will install anti-malware software on their phones.

On the other hand, based on the feedback, it is found that not much users are aware of the risk of charging their phones in public places. Even worse, most of them do trust the public charging stations and will interact with their phones. Therefore, juice filming attacks have large potential victims and will be more effective than common malware. To protect users' privacy, it is essential to educate people and let them pay more attention to the charging attacks.

### 3.3 Performance of Juice Filming Attacks

In this section, we attempt to explore the performance of juice filming attack in a real scenario.

**Attack set up.** We use a Linux system, where Ephiphan devices expose the Video4Linux (V4L) API. Whenever a device is connected to this Linux machine, V4L API provides a real-time stream of the display of the device. ffmpeg has been used to demonstrate this vulnerability (see our code), which has been configured to read the V4L stream and writes it to a file. A script then periodically checks for the presence of a new V4L-capable and USB connected device. If such a device is found, the stream is piped into a file. After 15 seconds of device detection, the script automatically pauses to evaluate whether the device is still connected. If connected, it continues streaming the screen contents into a file. Otherwise, it goes back into waiting mode - periodically polling for the presence of new V4L devices.

The workflow of the entire process can be described in Figure 6 and the code of video capture can be seen in Figure 7. As the script checks for update periodically, our juice filming attack can be launched automatically when detecting new phones. Then, all recorded videos (in MP4 format) will be stored in a folder. It is worth noting that the time for checking new devices can be tuned according to specific requirements. A demo of our attack can be seen in YouTube.3

**Study scenario.** In this study, we invite 10 volunteers to a lab room, who will interact with their phones during charging. Before the study, we did not inform them that our computer can record their information, but require them to use their phone as they would use as usual (during charging). We hope to get the real information as they would leak out in common lives. Each participant can use their phone no more than 10 minutes. After the study, we will notify them our attack, confirm the leaked information with them and delete all recorded videos for privacy reasons.

**Manual analysis.** After collecting the videos, we mainly conduct a manual analysis to extract users sensitive information. In Figure 8 and Figure 9, we give examples of how to infer information from Android devices and iOS devices, respectively.

- **Android OS.** In Figure 8 (a), we show how to record the unlock patterns and it is very easy to achieve. To infer the information such as user account and passwords, we can record the typed characters. For example, Figure 8 (b) shows that it is easily to observe the current typed characters, while Figure 8 (c) confirms that our record is correct.

3http://youtu.be/spK0Ay_YxxU.
Figure 8: Manual analysis on Android platform.

Figure 9: Manual analysis on iOS platform.

• iOS. The analysis is similar on iOS devices. Figure 9 (a) presents how we observe the typed PIN code. When a number is pressed, its color will be changed. Figure 9 (b) and Figure 9 (c) give an example of inferring user accounts and passwords from Facebook website. It is noticeable that the typed characters can be easily observed and recorded.

In this case, all input information can be observed and recorded by manual analysis. The major goal of the current work is to illustrate the feasibility of juice filming attacks, so that we do not automate the analysis process. However, the automation is definitely one of our future work, which can reduce time consumption and increase efficiency.

Study results. The study intends to check what can be obtained during our attack. In terms of the manual analysis, the detailed results are presented in Table 4. It is visible that a lot of private information has been leaked under our attack. More specifically, PIN code and unlock patterns can be obtained from every participant, since all persons need to unlock their phones and play. In addition, email accounts and social networking accounts such as Facebook, WeChat and QQ can be obtained. Even worse, some passwords of the social networking accounts are also available, as these participants re-entered the password for verification. During the manual analysis, it is found that all inputs are clearly recorded (i.e., we can extract and confirm user secrets by re-playing the videos more than once). On the whole, the study describes the feasibility and effectiveness of juice filming attacks in real scenarios.

4. DISCUSSIONS

The above studies present the feasibility and effectiveness of our attack in real scenarios, while there are still some challenges in real-world implementation. In this section, we discuss the requirements, present some issues of implementation and describe mitigation strategies to defend against our attack.

4.1 Requirements

In common cases, smartphone malware needs to install additional software on phone’s side such as java scripts or other code, which can leak target information or offer necessary functionality to attackers. This is very essential for any software-based threats. After installing the malicious modules, malware can steal users information with or without users interaction, depending on what information is focused. For example, if the stored images are the target information, then the malware can steal and transfer them to attackers without users interaction. On the other hand, if the target information is user’s PIN code, then the malware often needs to wait for users inputs.

For launching juice filming attacks, a major requirement is an additional hardware of VGA/USB interface that can be used to build a malicious charger. This is a big difference between our hardware-based threats and the software-based threats. Actually, the VGA/USB interface is not hard to get from Internet with different prices such as eBay. Thus, our attack is highly deployable in real scenarios. Another requirement of our attack is that users have to interact with their phones during the charging, so that our attack can record all inputs.4 However, as our attack does not install any software on phone’s side, it can retrieve sensitive information without users awareness. Due to the power of our attack, we consider these requirements are reasonable and affordable.

4.2 Challenges

Storage of videos. To launch our attack, storage is an issue for some resource limited hosts. In our study settings, a one-minute video costs 70 to 80 Megabytes, so that it is

<table>
<thead>
<tr>
<th>User ID</th>
<th>Leaked Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>User 1</td>
<td>PIN code, Facebook account, QQ account</td>
</tr>
<tr>
<td>User 2</td>
<td>PIN code, Email account/passwords</td>
</tr>
<tr>
<td>User 3</td>
<td>Unlock Patterns, QQ account</td>
</tr>
<tr>
<td>User 4</td>
<td>PIN code, Weibo account, Email accounts</td>
</tr>
<tr>
<td>User 5</td>
<td>Unlock Patterns, Email accounts/passwords</td>
</tr>
<tr>
<td>User 6</td>
<td>Unlock Patterns, WeChat account</td>
</tr>
<tr>
<td>User 7</td>
<td>Unlock Patterns, Email accounts</td>
</tr>
<tr>
<td>User 8</td>
<td>Unlock Patterns, Facebook account/passwords</td>
</tr>
<tr>
<td>User 9</td>
<td>PIN code, Email accounts, WeChat account</td>
</tr>
<tr>
<td>User 10</td>
<td>PIN code, Email accounts, QQ account</td>
</tr>
</tbody>
</table>

4It is worth noting that some wired attack surfaces [22] do not need user interaction, however, the are more device specific and not suitable for our case.
a burden if the system captures a very long video. Luckily, the video resolution can be tuned according to different scenarios. Generally, a higher resolution costs larger storage. With the advent of cloud computing, another way is to upload the video to the cloud so that even a resource limited device can keep launching our attack for a long time.

**Postprocess of videos.** In this work, we only process the captured videos manually, but it is very time-consuming when the number of available videos becomes large. To automate the postprocess of analyzing videos, it is promising to develop some intelligent algorithms (e.g., video processing). We leave this as one of our future directions.

**Capturing devices.** In this work, we mainly use either a PC or a notebook to capture the videos. In real-world implementation, it is more helpful to launch this attack using a smaller device, which can further degrade user awareness/concerns and increase the successful rate. One of the possible solutions is to use Raspberry Pi, which is a programmable hardware and is smaller than computers. This is one of our future directions as well.

### 4.3 Mitigation Strategies

As Android OS and iOS allow screen mirroring without explicit permission granting, this is the root cause of our attack. Actually, our attack can be feasible and effective in any iOS devices including iPhones and iPads (see Figure 10), so it is very necessary to protect users’ privacy by taking proper countermeasures to defend against our attacks. There are several potential mitigation strategies.

- **Like prompting the user whether to trust the computer that the smartphone is connected to (see Figure 1), the smartphone operating system should also warn and make notifications before output of the display. This strategy aims to increase user awareness on this type of attacks when a notification is shown.**

- **One of the potential defense solutions is to use a safe USB to present data leakage such as USB Condom [27]. This USB aims to prevent accidental data exchange when the device is plugged into another device with a USB cable. It achieves this by cutting off the data pins in the USB cable and allowing only the power pins to connect through. As such, these USB Condoms can prevent attacks like "juice jacking".**

- **For some specific secrets like PIN code and unlock patterns, if we use a fingerprint-based unlocking mechanism instead (e.g., the iPhone 5s), then our attack cannot easily capture these secrets. The key point here is not to input secrets from the touchscreen. For example, it is possible to use voice and other biometrics to interact with the phone.**

- **Even if some notifications can be shown to warn users when connecting the phone to the charging stations, it does not work for a tampered projector. In this case, it is better for people to take their own charging devices to defend against charging attacks and guarantee their information privacy.**

### 5. RELATED WORK

As smartphones become a major target for attackers, information leakage has been a big challenge for smartphone users. There is a line of research and practical works on how to infer users’ private information and data through malware, various sensors, and physical access attacks.

**Smartphone malware.** A lot of attacks on smartphones are based on malware. Lin et al. [14] found that the ADB capability can be exposed to any party with the INTERNET permission on the same device and built Screenmilk, an app that can detect the right moment to monitor the screen and pick up a user’s password when the user is typing in real time. Xing et al. [29] recently conducted research on the Android updating mechanism and found Pileup flaws, through which a malicious app can strategically declare a set of privileges and attributes on a low-version operating system (OS) and wait until it is upgraded to escalate its privileges on the new system. By exploiting the Pileup vulnerabilities, the app can not only acquire a set of newly added system and signature permissions, but also determine their settings and it can further substitute for new system apps. Some other malware can be referred to a recent survey [23]. In addition, most popular malware in literature is to steal information through side channels as below.

**Accelerometer side channel.** Cai and Chen [3] presented a side channel, motion, on touch screen smartphones with only soft keyboards. They identified that when users type on the soft keyboard on the smartphone, 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. In their evaluation, they showed that they were able to infer correctly more than 70% of the keys typed on a number-only soft keyboard on a smartphone. Later, Marquardt et al. [16] also demonstrated that an application with access to accelerometer readings on a modern mobile phone can use such information to recover text entered on a nearby keyboard. They
showed that through characterizing consecutive pairs of key-
press events, as much as 80% of typed content can be recov-
ered. Schlegel et al. [26] designed Soundcomber, a stealthy
Trojan with innocuous permissions that can sense the con-
text of its audible surroundings to target and 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 individu-
also who is driving, and pointed out that current smart-
phone operating systems allow any application to observe
accelerometer readings without requiring special privileges.
They further demonstrated that accelerometers can be used
to locate a device owner to within a 200 meter radius of
the true location. Then, Owusu et al. [21] described that
accelerometer readings are very powerful which can be used
to extract entire sequences of entered text on a smartphone.
They showed how a background application can use the ac-
celerometer as a side channel to spy on keystroke information
during sensitive operations such as account login. They pre-
sented that they could successfully break 59 out of 99 pass-
words using only accelerometer measurements logged during
text entry. Miluzzo et al. [20] presented TapPrints, a frame-
work for inferring the location of taps on touchscreens using
motion sensor data and showed that inferring English letters
could be done with up to 90% and 80% accuracy. Several
previous work can be referred to [1, 11, 15, 28, 32].

Physical side channel. Most of these attacks are based
on oily residues left on the touchscreen and the screen re-
fection from nearby objects. For example, Aviv et al. [2]
explored the feasibility of smudge attacks on touch screens
for smartphones. They considered different lighting angles
and light sources and the results indicated that the pattern
could be partially identifiable in 92% and fully in 68% of the
tested lighting and camera setups. Later, Zhang et al. [31]
presented a fingerprint attack against tapped passwords via
a keypad instead of graphical passwords. Their experiments
on iPad, iPhone and Android phone showed that in most
scenarios, the attack can reveal more than 50% of the pass-
words. For the screen reflection, Raguram et al. [24] showed
that automated reconstruction of text typed on a mobile
device’s virtual keyboard is possible via compromising re-
fections such as those of the phone in the user’s sunglasses.
By means of the footage captured in realistic environments (e.g.,
on a bus), they showed that their approach was able to
reconstruct fluent translations of recorded data in almost
all of the test cases.

Behavioral biometrics. With the spread of smartphone
malware, it is critical for users to take proper countermea-
sures to protect their privacy such as installing anti-malware
software and applying continuous authentication on smart-
phones. The use of biometrics is a potential solution based
on the measurements from human physiological or behav-
ioral characteristics, which can reduce the impact of the
juice filming attacks. Physiological biometrics usually use
measurements from the human body such as fingerprints,
iris scans and facial scans. For example, you can use your
fingerprint to unlock the iPhone 5s. Behavioral biometric
methods, which are a kind of continuous authentication, use
measurements from human actions such as keystroke dy-
namics, mouse dynamics and touch dynamics. On smart-
phones, touch dynamics has become a hot topic.

Feng et al. [6] presented FAST (Fingergestures Authen-
tication System using Touchscreen), which is a touchscreen
based authentication approach on mobile devices. It com-
plements and validates this data using a digital sensor glove.
Their study with 40 users indicated that FAST can achieve
a False Accept Rate (FAR) of 4.66% and False Reject Rate
of 0.13% for the continuous post-login user authentication.
At the same time, Meng et al. [17] proposed a user authen-
tication scheme based on touch dynamics, which uses 21
behavioral features related to touch dynamics for authenti-
cating users. In the evaluation, they collected and analyzed
touch gesture data of 20 phone users by comparing several
known machine learning classifiers. Experimental results
indicated that an average error rate of 3% can be achieved by
means of a combined PSO-RBFN classifier.

Later, Frank et al. [7] investigated whether a classifier
can continuously authenticate users on smartphones. They
employed a set of 30 behavioral touch features that can be
extracted from raw touchscreen logs and demonstrated that
different users populate distinct subspaces of this feature
space. They then proposed a framework to learn and clas-
sify the touch behavior of a user during an enrollment phase,
which is able to accept or reject the current user through
monitoring interaction. Their classifier achieves a median
equal error rate of 0% for intrasession authentication, 2%-,
3% for intersession authentication, and below 4% when the
authentication test was carried out one week after the en-
rollment phase. Several other related work can be referred to
[13, 18, 19, 25].

6. CONCLUSION AND FUTURE WORK

In this paper, we develop a novel charging attack, called
juice filming attack, which can automatically record user in-
puts when they are interacting with their phones during the
charging process. This attack is very effective in realistic scenar-
ios as it does not need to install any apps and ask for any
permissions, which also distinguish our work from others.
We further conduct several studies regarding users behav-
iors during charging and user awareness on charging attack.
It is found that users lack of awareness on this attack and
our attack can record all users inputs. Through manually
analyzing the recorded videos, we can extract users sensitive
information such as email accounts and social networking ac-
counts, even the relevant passwords. We later discuss some
implementation challenges and present several strategies to
defend against this attack.

Future work including automate the postprocess of video
analysis and use Raspberry Pi to construct a malicious charger
station. We hope this work can raise the awareness of users
that charging phones to public places may leak users’ data.

7. ACKNOWLEDGMENTS

We would like to thank all participants for their hard work
in the user study and all anonymous reviewers for their help-
ful comments in improving this paper.

8. REFERENCES

Emanations,” Proceedings of IEEE Symposium on

J.M. Smith, “Smudge attacks on smartphone touch


