Proceedings of the APAN – Network Research Workshop 2013

Passive Smart Phone Identification and Tracking with Application Set Fingerprints

Norihiro Fukumoto [1],*, Shigehiro Ano [1] and Shigeki Goto [2]

1 Communication Network Planning Laboratory, KDDI R&D Laboratories Inc. / 2-1-15 Ohara, Fujimino-shi, Saitama 356-8502 JAPAN
2 School of Fundamental Science and Engineering, Waseda University / Room 16 and 17, 5th Floor, Building 63, School of Fundamental Science and Engineering, WASEDA University, 3-4-1 Ohkubo, Shinjuku-ku, Tokyo 169-8555 JAPAN

E-Mails: fukumoto@kddilabs.jp; ano@kddilabs.jp; goto@goto.info.waseda.ac.jp

* Tel.: +81-49-278-7390; Fax: +81-49-278-7510

Abstract: Current smart phone users can be identified and tracked by fingerprint techniques. Fingerprint identification techniques can be used for legitimate purposes such as network management and traffic control to avoid excessive congestion. This paper proposes a user identification and tracking technique specific to smart phone users for supervised network management. This paper proposes the application set fingerprint, which is a simple set of User-Agent request-header fields in HTTP sessions. The application set fingerprint has three advantages, fully-passive fingerprint generation, potential of user trackability, and fingerprinting considering the users' privacy. The results show that the application set fingerprint is practically effective and network operators can use it for tracking smart phone users with the purpose of efficient network management.

Keywords: Smart phone; Fingerprint; HTTP; Deep Packet Inspection; User Tracking; Network Management.

1. Introduction

Along with the spread of smart phones, most users are becoming more privacy-conscious on their smart phones than on their laptops [1]. Their concerns are not only imaginary fears but can be justified. As evidence of this, some studies point out that smart phone users can be tracked with a high degree of accuracy. Human mobility patterns gathered from applications that access users' geographic locations are sufficient to identify unique users [2]. This suggests that most application developers have a potential to violate the privacy of the users. The users are concerned with privacy not only with malicious application developers, but also with network service providers and Web site administrators. These network operators are technically capable of gathering traffic data implicitly, and there are several techniques of identifying the user from the traffic data. Mobile devices often have individual differences such as physical addresses, protocol signatures, Bluetooth signals, browser fingerprints, and email headers. They are easily collectable by network operators and sufficient to identify a unique user [3]. Even if the user pays close attention to snooping and encrypts the traffic, simple Web accesses may leave unique keys to identify the users on the access log. Modern Web browsers have various individual discriminations that could be collected by Web site administrators [4]. This means that current smart phone users could be identified and tracked. The identification and tracking techniques can also be used for legitimate purposes. One example is a network management and traffic control to avoid excessive congestion. If network operators could find heavy network users and identify them, detailed and individualized control could be provided.

This paper proposes a user identification and tracking technique specific to smart phone users for supervised network management. We focused on not only the foreground data but also the background data which mobile applications originate, and defined the application set fingerprint consisting of the set of User-Agent request-header fields in the HTTP protocol. Note that User-Agent request-header fields can be extracted without access to the HTTP message-body considering the privacy of the user. The new proposed method discovers that background data is a useful clue to identify smart phone users. The rest of this paper is organized as follows: Section 2 describes related work. Section 3 explains HTTP traffic. We propose a new method in Section 4. Section 5 discusses experimental results. Section 6 concludes this paper.

2. Related Work

In this section, we provide a brief overview of the fingerprinting techniques. Fingerprinting is a common technique of identifying equipment or acquiring information about the attributes of the equipment. Fingerprinting utilizes various information, physical features of the equipment, individual differences in the physical devices, users' habits, and communication patterns. These fingerprinting techniques are not exclusive and can coexist in harmony.

The motives of fingerprinting are also various, identification of the device, user tracking, and network management. However, in this paper we cover only studies related to the privacy risks caused by traffic analysis.

2.1. Device Fingerprint

There is often interindividual variability in equipment even in industrial products. Device fingerprinting utilizes the individual differences in equipment identification. For instance, one study pointed out that digital camera devices could be identified by sensor pattern noises [5]. Another study discussed the capability of device identification using an individual variety of quartz crystal clocks [6]. These device fingerprinting techniques could be applied for device identification, and thus user tracking.

In addition to the variety in the hardware part of the equipment, the habit of user behavior and software could be fingerprinted. An example of a privacy risk caused by user behavior is the geographic location [2]. This problem is specific to smart phones kept by the users at all times, but could threaten the privacy of the smart phone users. The habit is not particular to humans, browsers have habits [4]. Combining Web browser measurements can realize global identifiers.

Our study investigates device fingerprinting as a novel technique to identify smart phone devices and thus users with the combination of installed applications with which they access the Web.

2.2. OS Fingerprint

The fingerprint technique is useful for classifying the traffic according to the Operating System (OS) of the traffic originators. Each TCP stack has OS-dependent characteristics in the header fields, default values, and enabled options [7]. The OS fingerprint is unsuitable for user identification, however, it can be helpful mainly for the purpose of statistical analysis of traffic data, and then it carries low privacy risk. However, the result of an OS fingerprint can be combined with other fingerprints to enhance the accuracy of identification.

2.3. Application Fingerprint

Another fingerprint technique is traffic classification in accordance with protocols. One study shows that the traffic behavior at the transport layer without knowledge of payload data is sufficient to identify the applications that generate traffic [9]. Although current applications do not always use static well-known ports, this application fingerprint approach provides accurate traffic classification considering the privacy of the traffic originators.

3. HTTP Traffic

The main idea of our study is to estimate the installed applications by inspecting the HTTP communications and generate a fingerprint that is sufficient to identify each smart phone.

Hypertext Transfer Protocol (HTTP) and HTTP Secure (HTTPS) are one of the most popular protocols on the Internet because of their simplicity and versatility [10]. Not only Web browsers but also many smart phone applications communicate through HTTP protocol on the back end, because HTTP protocol is a necessary and sufficient protocol for a number of objectives and most modern development environment supports HTTP protocol as a basic feature. We focused on the HTTP traffic originated by various applications installed in smart phones.

3.1. User-Agent Request-Header Field

HTTP protocol defines various header fields that give information about the client equipment and servers in plain text [11]. The User-Agent request-header field sent by client equipment contains multiple product tokens consisting of one or more product names and product versions such as:

```
  User-Agent: Mozilla/5.0 (Linux; U; Android 2.2; en-us; Nexus One
Build/FRF91) AppleWebKit/533.1 (KHTML, like Gecko) Version/4.0 Mobile
Safari/533.1
```

Each mobile application sets its own User-Agent request-header. Thus, service providers are capable of estimating which application on which equipment originates the HTTP communication precisely.

3.2. Background Data

Most smart phones support multitasking features that enable background data. Many mobile applications originate HTTP requests to update the data periodically or in accordance with push notifications. This specification means that network service providers have a chance to estimate which applications are installed in the smart phone equipment regardless of whether the application is active or inactive. Background data on smart phones potentially poses a privacy risk because it originates silently without users notice. This study analyzes all the traffic the smart phone generates including the background data with the purpose of passive estimation of the installed applications.

3.3. Deep Packet Inspection

Passive analysis of User-Agent request-header fields requires a Deep Packet Inspection (DPI) technique. Network service providers are able to extract User-Agent request-header fields from plain HTTP traffic in principle. In fact, several DPI products support the visibility of HTTP header fields including User-Agent.

4. Application Set Fingerprint

As described in the previous section, applications installed to the smart phones originate HTTP sessions interactively or non-interactively. Of course, installed applications of each smart phone are different. This means that the application set can be a fingerprint to identify the smart phone, and thus the user.

This paper proposes a novel fingerprint technique consisting of detected applications, called application set fingerprint. We describe the details of application set fingerprint in this section.

4.1. Definition of Application Set Fingerprint

Application set fingerprint can be defined as a simple set of User-Agent request-header fields in HTTP sessions. The fingerprint can also be expressed as a string of sorted and concatenated User-Agent request-header fields for convenience sake. As mentioned in Section 3, many applications for smart phones use HTTP protocol on the back end, then service providers are able to collect User-Agent request-header fields with a DPI technique.

User-Agent request-header fields should be collected for a definite period of time, and then collected User-Agent request-header fields should be sorted by identifiers at the network layer, usually IP addresses. In this way, an application set fingerprint is generated for each network layer identifier. Note that network layer identifiers are used just for temporary aggregation of User-Agent request-header fields. Thus, even if the smart phone renews the network layer identifier, the smart phone can be identified by recollection of the User-Agent request-header fields for a period and regeneration of the application set fingerprint.

4.2. Advantages of Application Set Fingerprint

There are three advantages to the proposed application set fingerprint. The first advantage is that the application set fingerprint is generated only from traffic data passively collected. Some fingerprint techniques require active measurements to collect the behavior of software. On the other hand, the User-Agent request-header fields can be collected completely passively. The second advantage is the potential of user trackability. Although most fingerprint techniques target identifying each item of equipment or device, sometimes the application set fingerprint can identify the smart phones of a particular user, because of synchronization between smart phones. For instance, an application installed in a smart phone can be automatically synchronized to a tablet owned by the same user. This means the smart phone and the tablet can be detected as devices owned by the same user. The third advantage is privacy. The application set fingerprint refers only to HTTP header fields, which means that private data in the HTTP message-body is not disclosed.

4.3. Entropy of Application Set Fingerprint

There are many applications that originate HTTP traffic for smart phones, and each application is updated sometimes. The User-Agent request-header field includes product version information which changes on each update. Thus, the different versions of the same application send the different User-Agent request-header fields they provide a clue to identify the smart phone. For instance, the following User-Agent request-header field of Mozilla, a popular Web browser, includes not only the product name and environment variables but also product versions. Of course, the product versions change on each update and cause variations.

```
  User-Agent: Mozilla/5.0 (Linux; U; Android 2.2; en-us; Nexus One
Build/FRF91) AppleWebKit/533.1 (KHTML, like Gecko) Version/4.0 Mobile
Safari/533.1
```

We have HTTP-enabled application set A, and each application $a \in A$ has version variation V_a. The distribution of the application set fingerprint can be calculated as an infinite product of the variations of each application. It is notable that undetected application can be treated as a kind of product version which should be added to V_a. For example, when an application has three different versions such as `Dalvik/1.2.0`, `Dalvik/1.4.0`, and `Dalvik/1.5.0`, the version variation V_a comes out to three. In addition, a null version is added to describe the case when that application is not installed. Here, the variation of application set fingerprint N is given by:

$$N = \prod_{a \in A} (V_a + 1) \tag{1}$$

Note that the real variation of application set fingerprints is not obvious because each software developer names User-Agent request-header fields by their own. Thus, we suppose that the size is finite for convenience sake, and estimate it from collected traffic data in practice.

When new application set fingerprint n is received, the surprisal I can be calculated. When the variation of application set fingerprints N and the fingerprinting algorithm f_n are given, we have a discrete probability density function $P(f_n), n \in [0, 1, \cdots, N]$. The surprisal I and the entropy of the distribution $P(f_n)$ are given by:

$$I(f_n) = -\log_2\big(P(f_n)\big) \tag{2}$$

$$H(f_n) = \sum_{n=0}^{N} P(f_n) \log_2\big(P(f_n)\big) \tag{3}$$

Here, the surprisal is a measure of the information content given by the new fingerprint. The equations simply mean that a rare application set gives more information on the identification.

5. Experiments

We analyzed traffic from smart phones to determine the efficiency of application set fingerprints. We collected 44,248 fingerprints at a smart phone network by applying the definition of the application set fingerprint shown in Section 4.

Figure 1(a) shows the frequency distribution of application set fingerprints in the double logarithmic plot. The result shows that the distribution has an extremely long tail and 92.2% of the smart phone users have a unique application set fingerprint. In other words, network operators can easily make a shortlist of target users.

Another question is how many User-Agent request-header fields are required to identify a unique user. Figure 1(b) depicts the frequency distribution of the number of User-Agents in a unique user application set fingerprint group. The result shows that a specific user can be identified with as few as 20 applications.

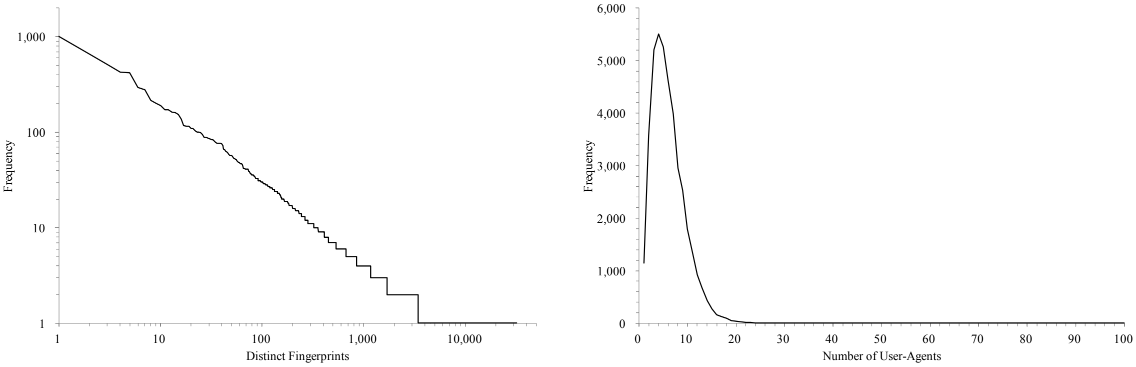

Figure 1. (a) Frequency distribution of application set fingerprint. **(b)** Frequency distribution of number of User-Agents in unique application set fingerprint group.

6. Conclusions

This paper proposes a new method for passive smart phone identification and tracking with application set fingerprints. Our study suggests that network operators have a potential to identify current smart phone users for the purpose of smart network management.

The primary contribution of the paper is the application set fingerprint which utilizes not only the foreground data but also the background data originating from mobile applications. An application set fingerprint is a simple set of User-Agent request-header fields in HTTP sessions. Most current applications for smart phones use HTTP protocol on the back end. Then, service providers are able to collect User-Agent request-header fields by DPI technique. The application set fingerprint has three advantages: it can be generated only from traffic data passively collected, it has potential user trackability, and it can track each user considering privacy.

The experiment results show that the application set fingerprint is practically effective, which means that smart phone users can be tracked by network operators.

References

1. Chin, E.; Felt, A. P.; Sekary, V.; Wagner, D. Measuring User Confidence in Smartphone Security and Privacy. *Proceedings of the Eighth Symposium on Usable Privacy and Security (SOUPS '12)*, 2012, pp.1-16.
2. Montjoye, Y.A. De; Hidalgo, C. A.; Verleysen, M.; Blondel, V. D. Unique in the Crowd: The privacy bounds of human mobility. *Scientific Reports*, 2013, Vol. 3, No. 1376.
3. Takeda, K. User Identification and Tracking with online device fingerprints fusion. *IEEE International Carnahan Conference on Security Technology (ICCST)*, 2012, pp.163-167.
4. Eckersley, P. How unique is your web browser? *Proceedings of the 10th International Conference on Privacy Enhancing Technologies*, 2010, pp.1-18.
5. Lukas, J.; Fridrich, J.; Goljan, M. Digital camera identification from sensor pattern noise. *IEEE Transactions on Information Forensics and Security*, 2006, Vol.1, Issue 2, pp.205-214.
6. Kohno, T.; Broido, A.; Claffy, K. C.; Remote physical device fingerprinting. *IEEE Transactions on Dependable and Secure Computing*, 2005, Vol.2, No.2, pp.93-108.
7. Zalewski, M. p0f. http://lcamtuf.coredump.cx/p0f3/.
8. Krishnamurthy, B. Generating a privacy footprint on the Internet. *Proceedings of the ACM SIGCOMM Internet Measurement Conference*, 2006, pp.65-70.
9. Karagiannis, T.; Papagiannaki, K.; Faloutsos, M. BLINC: Multilevel Trafic Classifcation in the Dark. *Proceedings of the ACM SIGCOMM Internet Measurement Conference*, 2005, pp.229-240.
10. Falaki, H.; Lymberopoulos, D.; Mahajan, R.; Kandula, S.; Estrin, D. A First Look at Traffic on Smartphones. *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*, 2010, pp.281-287.
11. Fielding, R.; Gettys, J.; Mogul, J.; Frystyk, H.; Masinter, L.; Leach, P.; Berners-Lee, T. Hypertext Transfer Protocol -- HTTP/1.1. *Internet Engineering Task Force*, 1999, Request for Comments 2616, http://www.ietf.org/rfc/rfc2616.txt.