Datasets of GridDroid

Jun Ma[1,2], *Member, CCF, ACM, IEEE*, Qing-Wei Sun[1,2,3], Chang Xu[1,2], *Senior Member, CCF, IEEE, Member, ACM*, and Xian-Ping Tao[1,2], *Senior Member, CCF*

[1] *State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China*
[2] *Department of Computer Science and Technology, Nanjing University, Nanjing 210023, China*
[3] *Huatai Securities, Nanjing 210019, China*

E-mail: majun@nju.edu.cn; sunqingwei@htsc.com; changxu@nju.edu.cn; txp@nju.edu.cn

Abstract Our evaluations of GridDroid involve three different datasets (i.e., S_1, S_2 and S_3). There were totally 527 apks (i.e., 256 in S_1, 114 in S_2 and 157 in S_3) and 138,601 app pairs, among which there are 233 repackaging pairs.

1 Datasets

We use datasets S_1 and S_2 from RepDroid [1] and S_3 from RegionDroid [2] as well as 52 repackaging pairs (104 apks) from [3] to evaluate the resistance and credibility of GridDroid.

S_1 consists of 256 apks and it is constructed to evaluate a birthmark's resistance to obfuscation and encryption. Specifically, 80 apks were obtained by applying three different obfuscation/encryption tools (i.e., the FakeActivity and NestedLayout obfuscation tools implemented by RepDroid and the encryption tool AndroCrypt provided by [4]) to 20 apks downloaded from F-Droid[1]. 72 apks in S_1 were obtained by applying the encryption provided by Ijiami[2] to 36 apps[3] downloaded from Wandoujia[4]. Finally, we also included in S_1 52 piggybacking pairs[5] (i.e., 104 apks) downloaded from the dataset provided by [3].

To evaluate a birthmark's capability of distinguishing different apps (potentially with similar GUIs), we constructed dataset S_2 and S_3. We assumed apps of the same category would likely share some common in their GUIs and we downloaded about 15 or 16 commercial apps from each of the 8 categories (i.e., Office, Communication, Finance, Education, News, Reading, Health, and Tool) of Wandoujia. Finally, we formed S_2 with 114 commercial apps[6]. All apps in S_2 were top ranked popular apps with at least 100,000 downloads.

Hybrid applications are becoming increasingly popular, and there are a number of hybrid application development frameworks available today. We believe apps using the same framework would share some common in their GUIs. Therefore, we downloaded 40 apps from the showcases[7] of seven different popular hybrid frame-

[1] https://f-droid.org/, Oct. 2021.

[2] https://www.ijiami.cn/, Oct. 2021.

[3] Originally, in our previous papers [1] and [2], there were 38 downloaded apps from Wandoujia; however, we found two of them were exactly the same (with the same MD5), and we only keep one of them in this paper. Besides, there was one app that could not run successfully in our experiment, so we removed it from the data set as well.

[4] https://www.wandoujia.com/, Oct. 2021.

[5] We downloaded 297 repackaging pairs from the dataset of [3]. We manually installed and tried to launch each apk, and removed pairs containing at least one apk that cannot be installed or launched. We also excluded apks implemented by Unity3D or other engines built on top of OpenGL, as they were out of the scope of this paper. Finally, we collected 52 piggybacking pairs involving 104 apks and included them in the dataset S_1.

[6] Originally, there were 125 apks in S_2; however, 11 apps could not be launched successfully during our experiment, so we removed it from the data set.

[7] Actually, 27 of the 40 apps are still available in Google Play, including ebay mobile (`com.ebay.mobile`), paypal mobile (`com.paypal.android.p2pmobile`).

works (i.e., PhoneGap[8], Appcelerator[9], Framework7[10], Ionic[11], Mobile Angular UI[12], Onsen UI[13], and Xamarin[14]. We also downloaded 117 apks that contain HTML, css files or layout xml files containing any 'webview' from Androzoo [5]. Finally, we formed S_3 with 157 commercial apps.

Table 1. Statistics of apks in Datasets S_1, S_2, S_3

Datasets		Size(KB) Min	Size(KB) Max	Size(KB) Avg	#Activity Min	#Activity Max	#Activity Avg
S_1 (256 apks)	F-Droid (20+60 apks)	41	2,186	551	2	23	6
	Wandoujia (36+36 apks)	382	39,151	8,219	2	131	47
	[30] (104apks)	53	30,784	3,855	1	137	15
S_2(114 apks)		382	76,516	18,658	3	715	130
S_3(157 apks)		447	62,663	9,042	1	165	16
Overall (527 apks)		41	76,516	9,448	1	715	45

* For more details about the three datasets, please refer to [1–3].

Statistics about the sizes and numbers of activities of the original apps contained in each datasets are shown in **Table** 1.

1.1 RPs in Datasets

There were totally 527 apks (i.e., 256 in S_1, 114 in S_2 and 157 in S_3) and 138,601 app pairs. We manually checked these apks to find out the ground truth of possible RP candidates. Briefly, like [6], we first ran each app and assigned some flags (e.g., news, reading, education, financial, system, travel, tools, games) to the app based on the contents viewed during its execution. Then for each pair of apps which share at least one common flag, we manually compared the pair to see if it was an RP (candidate) or not. In our experiments, we try to show whether it is possible to identify and compare apps based on birthmarks built from their run-time GUI traces. Therefore, for all apps under test, we ignore their developer information as well as the corresponding certificates[15] used to sign them. Finally, we found 233 RPs as shown in **Table** 2.

Table 2. Number of RPs within/across S_1, S_2, S_3

	S_1	S_2	S_3	Total
S_1	212	14	-	226
S_2	-	5	-	5
S_3	-	-	2	2

RPs within the same dataset

Specifically, 212 RPs were found in S_1:

- The 52 repackaging pairs downloaded from [3].
- For each of the 20 apks downloaded from F-Droid, there were three obfuscated apks. Each pair of the four apks formed an RP[16], and there were totally 120 RPs of this kind.
- For each of the 36 apps downloaded from Wandoujia, there was one obfuscated apk (obtained by Ijiami) accordingly.
- Two medical consultation apks (`com.medapp v3.0.920.1` and `com.medapp.man v3.0.922.1`) downloaded from Wandoujia, as shown in Fig. 1, were developed by the same company and signed by the same certificate. The difference is that `com.medapp.man` is a dedicated version for men, while `com.medapp` is a dedicated version for

[8] http://phonegap.com/, Mar. 2018.
[9] https://www.appcelerator.com/, Mar. 2018.
[10] https://framework7.io/, Mar. 2018.
[11] https://ionicframework.com/, Mar. 2018.
[12] http://mobileangularui.com/, Mar. 2018.
[13] https://onsen.io/, Mar. 2018.
[14] https://www.xamarin.com/, Mar. 2018.
[15] Certificates can be easily added to tools to ignore pairs of apks developed by the same developer.
[16] In this paper, we considered one obfuscated/encrypted app and its original one as an RP; besides, we also considered a pair of apps obtained by obfuscating/encrypting the same app as an RP.

women. These two apks together with their encrypted apks formed four extra RPs. We refer the repacking pair of the two original apks as RP_{S_1}**-1** for simplicity.

We also found five RPs in S_2 and two RPs in S_3:

- RP_{S_2}**-1** consists of two different versions (i.e., `v5.7.9` and `v5.8.1`) of the same popular news app `com.ss.android.article.news`. They are signed by the same certificate of `ByteDance`.
- RP_{S_2}**-2** is exactly the same as RP_{S_1}**-1** in S_1. The two apks (`com.medapp v3.0.920.1` and `com.medapp.man v3.0.922.1`) are included in S_2 as well.
- RP_{S_2}**-3** consists of two un-signed apks (with different MD5 values) of the same news app `com.tencent.new v5.1.12`.
- RP_{S_2}**-4** consists a pair of apks `com.v.study v2.17` and `com.v.zy v2.43`, both of which bring together the answers behind tens of thousands of textbooks used in China. Although their package names are different, the two apks provide exactly the same GUIs. Besides, they are signed differently and have different MD5 values.
- RP_{S_2}**-5** consists of a pair of social marriage and dating apps `com.river.qiyuan v1.6.1` and `com.meiguihunlian v1.7.7` as shown in Fig. 2. Although signed differently, they are actually developed by the same company.
- RP_{S_3}**-1** consists of a pair of theme apps `com.touchpal.otheme_wise_leopard.Index v2.3 True Green` and `com.jb.gokeyboard.theme.cutekeyboards_future_light v5.0 Poison Green` as shown in Fig. 3. They are signed with different certificates.
- RP_{S_3}**-2** consists of two weather apps `com.yongmedia.wins v4.3.601` and `com.wpix.android.weather v4.3.500` as shown in Fig. 4. They are signed with different certificates.

RPs across different datasets

Three apks (including the two of RP_{S_2}-2 and another apk) in S_2 are included in S_1 as well. Specially, the three apks are download from Wandoujia and encrypted by IJiami in S_1. As a result, the three apks (together with their encrypted apks) in S_1 and their counter-parts in S_2 form 10 RPs across S_1 and S_2.

Besides, S_1 and S_2 share two pairs of apks with the same package name but different versions. The first pair consists of two versions (`v1.2.1` in S_1 and `v2.4.4`) of `com.tencent.reading`. The second pair consists of two versions (`v5.1.2` in S_1 and `v7.1.0`) of `com.baidu.homework`. These apks (together with the two corresponding encrypted apks) finally form 4 RPs across S_1 and S_2. For the sake of simplicity, we refer the two pair as $RP_{S_1S_2}$-1 and $RP_{S_1S_2}$-2 accordingly, and we use $RP_{S_1S_2}$-1' (and $RP_{S_1S_2}$-2') to denote the RP obtained by replacing the original apk in $RP_{S_1S_2}$-1 with its encrypted version.

(a) (b) (c)

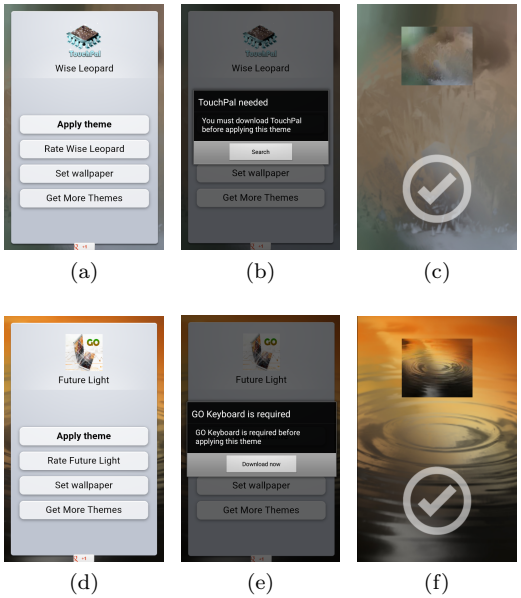
(d) (e) (f)

Fig. 1. The repackaging pair RP_{S_1}-1 in S_1. (a) - (c) `com.medapp`. (d) - (f) `com.medapp.man`.

(a) (b) (c)

(d) (e) (f)

Fig. 2. The repackaging pair RP_{S_2}-5 in S_2. (a) - (c) `com.river.qiyuan`. (d) - (f) `com.meiguihunlian`.

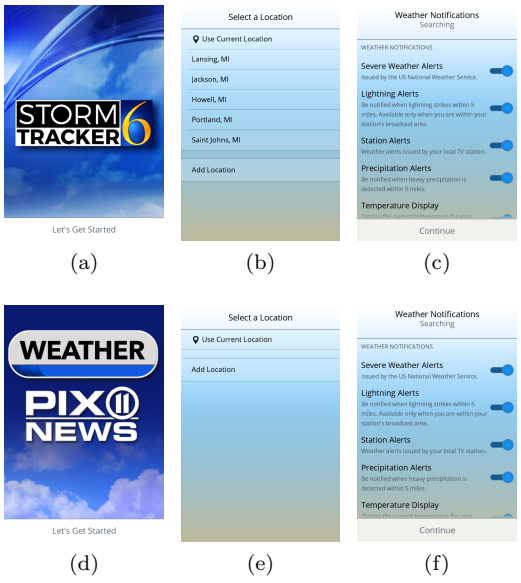


(a) (b) (c)

(d) (e) (f)

Fig. 3. The repackaging pair RP_{S_3}-1 in S_3. (a) - (c) `com.touchpal.otheme_wise_leopard`. (d) - (f) `com.jb.gokeyboard.theme.cutekeyboards_future_light`.

.



(a) (b) (c)

(d) (e) (f)

Fig. 4. The repackaging pair RP_{S_3}-2 in S_3. (a)-(c) `com.yongmedia.wins`. (d)-(f) `com.wpix.android.weather`.

References

[1] Yue S, Feng W, Ma J, Jiang Y, Tao X, Xu C, Lu J. Repdroid: An automated tool for android application repackaging detection. In *Proceedings of the 25th International Conference on Program Comprehension*, ICPC '17, 2017, pp. 132–142.

[2] Yue S, Sun Q, Ma J, Tao X, Xu C, Lu J. Regiondroid: A tool for detecting android application repackaging based on runtime ui region features. In *2018 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, 2018, pp. 323–333.

[3] Li L, Bissyandé T F, Klein J. Rebooting research on detecting repackaged android apps: Literature review and benchmark. *IEEE Transactions on Software Engineering*, 2021, 47(4):676–693.

[4] Kim D, Gokhale A, Ganapathy V, Srivastava A. Detecting plagiarized mobile apps using api birthmarks. *Automated Software Engineering*, 2015, pp. 1–28.

[5] Allix K, Bissyandé T F, Klein J, Le Traon Y. Androzoo: Collecting millions of android apps for the research community. In *Proceedings of the 13th International Conference on Mining Software Repositories*, MSR '16, 2016, pp. 468–471.

[6] Soh C, Tan H B K, Arnatovich Y L, Wang L. Detecting clones in android applications through analyzing user interfaces. In *Proceedings of the 2015 IEEE 23rd International Conference on Program Comprehension*, ICPC '15, 2015, pp. 163–173.