A Large-scale Investigation into Geodifferences in Mobile Apps

Renuka Kumar, Apurva Virkud, Ram Sundara Raman, Atul Prakash, and Roya Ensafi

University of Michigan

Abstract

Recent studies on the web ecosystem have been raising alarms on the increasing geodifferences in access to Internet content and services due to Internet censorship and geoblocking. However, geodifferences in the mobile app ecosystem have received limited attention, even though apps are central to how mobile users communicate and consume Internet content. We present the first large-scale measurement study of geodifferences in the mobile app ecosystem. We design a semi-automatic, parallel measurement testbed that we use to collect 5,684 popular apps from Google Play in 26 countries. In all, we collected 117,233 `apk` files and 112,607 privacy policies for those apps. Our results show high amounts of geoblocking with 3,672 apps geoblocked in at least one of our countries. While our data corroborates anecdotal evidence of takedowns due to government requests, unlike common perception, we find that blocking by developers is significantly higher than takedowns in all our countries, and has the most influence on geoblocking in the mobile app ecosystem. We also find instances of developers releasing different app versions to different countries, some with weaker security settings or privacy disclosures that expose users to higher security and privacy risks. We provide recommendations for app market proprietors to address the issues discovered.

1 Introduction

We often view the Internet as a worldwide medium for communication without regard for geographic location [64]. However, studies have shown differences in Internet equity, for instance, in access to Internet content based on a user's geolocation [60, 72, 98, 124]. While censorship is a well-known enabler for such regional differences [2, 24, 90, 111, 128], there are emerging trends of geoblocking, a phenomenon where service providers or developers deny access to users in certain countries or regions. Recent studies on the web ecosystem show how service providers, given an option, lean towards indiscriminate blocking, effectively isolating certain countries (e.g., Cuba) and essential services (e.g., banking) [1, 80, 118]. Recognizing the gravity of this problem, in 2018, the EU passed regulations that ban unjustified geoblocking [34].

Mobile users worldwide access the Internet through apps downloaded from app markets like Google Play. Thus, any interference in app markets can result in different *app equity*, i.e., different access to apps or security and privacy offerings based on a user's geolocation. For example, Figure 1 shows different views of the LinkedIn app's homepage on Google Play for users in three countries. The app is available for install in the US and unavailable in Iran and Russia, though users perceive unavailability differently in the latter two countries. Although there are over 8.9 million apps and 3.5 billion smartphone users worldwide [89, 103], geodifferences in the mobile app ecosystem have received only limited attention.

In this paper, we present the first large-scale investigation into geodifferences in the mobile app ecosystem. Broadly, we are curious to know: (i) if we can download an app from, say, the US, Iran, and Russia at the same time; and (ii) whether the app, if available, has geodifferences. Given our goals for a wide geographic study, we select 5,684 globally popular apps from Google Play. Google Play is the largest and the most accessible app market, with over 2 billion active devices and 2 million apps that reach over 190 countries [14, 29], making it an obvious choice for our geographic study. We collect our measurements from 26 countries carefully chosen to have reliable direct vantage points while ensuring ample diversity in terms of geography, gross domestic product, and Internet freedom scores by Freedom House [39].

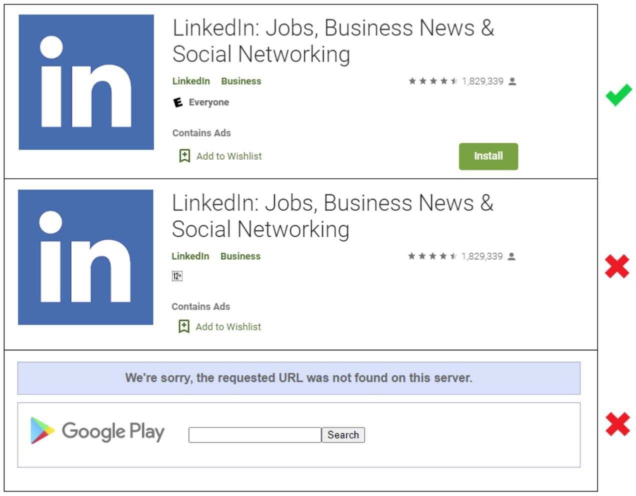
There are *significant* challenges in conducting large-scale measurements for thousands of apps from different locations. For instance, Google Play is a volatile app market with millions of apps, governed by opaque regulations [76, 95, 113]. Unlike web measurements, there are no known fully automated tools for collecting mobile measurement data. While web geoblocking has some clear signals (e.g., `403 Forbidden` HTTP response), the indicators for blocking in the mobile app ecosystem are unknown. Additionally, to measure geodifferences in apps, we need to download app binaries from many countries.

This paper makes several contributions that enable large-scale measurement studies of complex mobile app ecosystems. First, we identify the variables that impact measurements from Google Play through a set of preliminary experiments. Second, we design a semi-automated measurement technique that captures a reasonable snapshot of Google Play as seen by users in 26 countries. Using a parallel test-bed, we collect 117,233 app binaries and 112,607 privacy policies, which to the best of our knowledge, is the largest multi-country app dataset in the research community. Third, using control experiments, we extract the signals for geoblocking from Google's opaque server-side responses and deduce who is responsible for the blocking. Fourth, for apps that are not geoblocked, we investigate if users in certain regions are exposed to higher security and privacy risks from geodifferences in app features. Finally, we provide recommendations for app market proprietors like Google Play to address the issues we find.

Our results show high amounts of geoblocking with 3,672 globally popular apps geoblocked in at least one of 26 countries. We find that Iran and Tunisia have the highest geoblocking rates of 2,256 and 2,681 apps, respectively. In contrast, previous work on the web found up to 71 geoblocked domains from the Alexa top 10K list in the most affected countries [80]. While our data corroborates anecdotal evidence of takedowns due to government requests (e.g., recent ban of Chinese apps in India) [114], unlike common perception, we find that blocking by developers is significantly higher than takedowns in all our countries and app categories, and has the most influence on geoblocking in the mobile app ecosystem. Amongst the countries, Iran is the most blocked by developers and is the top outlier country in every app category. We believe this is because developers have unmoderated access to country targeting features on Google Play, which, as research has shown [80], could disproportionately isolate some regions.

While most developers release the same apps geographically, we find 596 apps with geodifferences, with confirmed instances of developers targeting different app versions to different countries, thus exposing users in certain countries to higher security and privacy risks. For instance, we find instances of the same apps requesting different permissions, using additional ad trackers, or selectively using unencrypted communication in different regions. While our data shows the positive influence of data protection laws (e.g., GDPR) in privacy policies in regions where such laws are enforced, we also find the same apps using outdated policies in countries with older legislation. Privacy policies of some apps in certain countries like Iran and Turkey could not be downloaded due to geoblocking of the websites hosting them.

We suggest several steps that Google and other app market proprietors could take to address some of the issues we find. For instance, app market proprietors could moderate their country targeting features, push for transparency from developers on their need for geodifferences in apps, and redress the blocking of privacy policies in certain countries by hosting the app's policy themselves to ensure its availability. We shared our work with Google and submitted a full disclosure on all the apps for which we observed geodifferences in security and privacy features. Google's privacy team has acknowledged our disclosures, and they are aware of the concerns raised through our research. To encourage further studies on the various aspects brought out by this work, we make our measurement data and code available at https://github.com/censoredplanet/geodiff-app.


LinkedIn: Jobs, Business News & Social Networking

LinkedIn: Jobs, Business News & Social Networking

We're sorry, the requested URL was not found on this server.

Figure 1: **User's View of Geoblocking:** *LinkedIn's* homepage on Google Play from the US, Iran, and Russia from top to bottom. Users in the US can download the app via the Install button, users in Iran see the homepage without the Install button, and users in Russia see a *URL not found* error on accessing the same page.

2 Background

Internet fragmentation (or "balkanization") is emerging as a growing concern given the wide disparity in business and government practices worldwide [62]. Web geoblocking studies have shown that developers on a content delivery network (CDN, e.g., Cloudflare), given an option, indiscriminately geoblock their content, furthering the fragmentation. However, not much is known about geoblocking in the mobile app ecosystem, where app market proprietors or developers may block access to mobile apps that users use to access content from their mobile devices. Furthermore, it is poorly understood to what extent developers distribute different versions of apps in different countries and whether those versions differ on security and privacy protections. This study focuses on geographical differences (geodifferences) from geoblocking or differences in app releases.

2.1 Google Play

Google Play is the largest and the most accessible app market, with 2 billion active devices and over 2 million apps reaching over 190 countries [14]. Global app markets like Google Play are not the only means for developers to distribute apps. There are also local app markets, for instance, such as those hosted by cellular network providers (e.g., Docomo in Japan [32]) or independent publishers (e.g., CafeBazaar in Iran [36]). However, given Google Play's reach, any geodifferences here will have the most impact on users and makes it an obvious choice for our geographic study.

Google Play allows developers to restrict app availability to users, for instance, based on their country (country targeting) [43]. We call such geoblocking due to developers country targeting their apps as *developer-blocking*. Microsoft's Skype Lite, which is available only in India [82], and Hulu, which is available only in the US [61], are examples of developer-blocked apps. In a special case of country targeting, restrictions are imposed on access to paid apps or in-app purchases, for instance, due to embargo rules or Google's policy in a country (e.g., Iran) [16, 48]. Other forms of targeting are device targeting, where apps are released for specific devices, or carrier targeting, where apps are released for specific cellular service providers [10, 51].

At a high level, Google's transparency report shows content removal because of government requests and violations of Google's policies [53]. Real-world reports corroborate app removal requests (*takedowns*) specifically from Google Play for the same reasons. For instance, Google taking down LinkedIn in Russia [37] and the Indian government's ban of several Chinese apps [114] are examples of takedowns from government requests. On the other hand, Google's removal of Fortnite [19] is an example of a takedown due to a violation of Google policy. Broadly, we call the former *government-requested takedown* and the latter a *non-compliance takedown* and aim to distinguish the two in our work. While Google's non-compliance takedowns tend to be global, takedowns from government requests are regional.

Given how apps on Google Play are subject to developer-blocking and takedowns, a user's view of an app may vary across regions. Figure 1 shows an example of how users in the US, Iran, and Russia see the homepage of the LinkedIn app (as of June 2020). A user can download the app using the "Install" button on its homepage in the US. A user in Iran, in contrast, sees the same homepage without the Install button, indicating that they cannot download the app. Finally, in Russia, accessing the app's URL returns an error. Given how a user's snapshot of Google Play may vary geographically, our goal is to characterize who is responsible for the differences we observe.

A user can access an app on Google Play directly via a URL with a fixed structure that points to its homepage. For the LinkedIn app, the URL to its homepage is `https://play.google.com/store/apps/details?id=com.linkedin.android`, where `com.linkedin.android` is the app's unique ID. The app's homepage contains its *metadata* such as the app's category, the number of installs, the app version, and the URL to download the `apk` upon clicking the Install button.

2.2 Android Security and Privacy

Android *apps* are programs written to operate on Android devices. An Android app is packaged into an installable archive called the Android Package (`apk`). An `apk` contains compiled classes, resources (e.g., images), assets (e.g., media files), configuration files, and a manifest file (`AndroidManifest.xml`). The Android manifest contains app settings used by both Google Play and the Android device, such as an app's unique app ID, its user-facing version name, internal version code for developers, permissions, and third-party libraries [11, 17]. This work studies app features that may have security and privacy risks to users.

Android has three notable security features that protect users—app permissions, signatures, and settings for encrypted communication. App permissions control access to system features (e.g., GPS) and is classified into one of four protection levels [8]: (i) *Normal*, which carries the least risk; (ii) *Dangerous*, which is high-risk and requires explicit user consent; (iii) *SignatureOrSystem*, which is privileged and used by system apps (e.g., apps by device vendors); and (iii) *Signature*, which is used to share data between apps.

Android apps are signed using a digital signature that serves as a bridge of trust between a user, developer, and Google Play. Android provides three signing algorithms: versions V1 (for Android versions < 7.0), V2 (for Android 7.0+), and V3 (for Android 9+). Ideally, a developer has to sign with *all* three schemes for maximum security [12]. Note that there is also a V4 signing algorithm that was released September 2020 for Android 11 [28]. However, we exclude the V4 signature algorithm from this work since our `apks` predate Android 11.

Android allows developers to set their communication preferences via a *Network Security Policy* file (`network_security_config.xml`) [15] or an app's manifest file [18]. In the absence of this config file, prior to Android 9, an app's communications with servers by default use the unencrypted HTTP protocol unless disabled in the app's manifest file. However, with Android 9, the default is encrypted (HTTPS) communication. Given this nuance, not setting communication preferences is potentially dangerous since the system may default to unencrypted communication based on Android version.

Android developers may use third-party libraries for convenience or to provide services such as in-app advertisements or billing [27]. However, prior research has shown that third-party libraries may compromise a user's privacy by leaking their sensitive data [27, 57, 73]. To protect a user's privacy, Google requires developers to disclose collection of, access to, or use of sensitive data (e.g., personally identifiable information) [50], via a privacy policy that developers must host on an active URL. The policy must be listed on the app's homepage on Google Play [45] and must follow data protection laws when appropriate, including the US-based California Consumer Privacy Act (CCPA) [108] and the European Union's General Data Protection Regulation (GDPR) [63].

3 Measurement Design & Data Collection

Given that our goal is to study geodifferences, we want to capture a snapshot of apps from Google Play as seen by users

Country	Code	Region	FHIF	Country	Code	Region	FHIF
Canada	CA	NA	F(87)	Ukraine	UA	Europe	PF(56)
Germany	DE	Europe	F(80)	India	IN	Asia	PF(55)
USA	US	NA	F(77)	Zimbabwe	ZW	Africa	PF(42)
UK	UK	Europe	F(77)	Turkey	TR	Europe	NF(37)
Australia	AU	Oceania	F(77)	Russia	RU	Europe	NF(31)
Japan	JP	Asia	F(73)	Venezuela	VE	SA	NF(30)
Hungary	HU	Europe	F(72)	Bahrain	BH	Asia	NF(29)
Kenya	KE	Africa	PF(68)	UAE	AE	Asia	NF(28)
Colombia	CO	SA	PF(67)	Egypt	EG	Africa	NF(26)
South Korea	KR	Asia	PF(64)	Iran	IR	Asia	NF(15)
Tunisia	TN	Africa	PF(64)	Hong Kong	HK	Asia	-
Mexico	MX	NA	PF(60)	Ireland	IE	Europe	-
Singapore	SG	Asia	PF(56)	Israel	IL	Asia	-

Table 1: **Country List.** 26 countries with their ISO Code, region, and Freedom House Internet Freedom score, sorted by the freedom score (unavailable for three countries). Abbrv: NA= North America, SA = South America, F= *Free*, PF= *Partly Free*, NF= *Not Free*.

in many countries. Measurements of such kind pose several challenges as noted in prior research on the web ecosystem [2, 80, 111], such as finding a vantage point closest to a user. However, in the mobile app ecosystem, we have the additional burden of downloading thousands of apps and their metadata from Google Play in different countries. These downloads may take weeks to complete, with app updates in between that complicate comparative analysis. We thus have to find vantage points suitable for long-running downloads, account for app updates, and factor in network errors and latency.

Country Selection. For this study, we carefully choose an initial list of 30 countries with ample diversity in geography, gross domestic product (GDP), and Internet freedom scores as measured by Freedom House [39]. We first considered choosing countries based on their GDP alone as in prior work on Internet geoblocking [80]; however, countries like India that are recently seeing a *crisis in expression* [23] have higher GDP than countries like Canada that are relatively free. Hence, we rely on *Freedom House's Internet Freedom* (FHIF) score, which numerous studies have used for country selection and analysis [2, 25, 92, 99], as an approximate indicator of Internet freedom. Since the study could potentially span several months, acquiring cost-effective and reliable vantage points was a significant challenge; we thus limit ourselves to 30 countries initially.

Testing Vantage Points. Prior research on Internet measurements has shown that data collected from different vantage points may have different properties [100]. Hence, we first conduct a preliminary study to confirm that a user's view of the app market from a non-residential vantage point is the same as a residential vantage point. To confirm this, we collect app metadata by proxying HTTPS requests to Google Play's home pages for a manually curated set of low-risk apps (e.g., Google Chrome, Netflix) from different vantage points and compare the data collected. We use Luminati, a commercial platform that sells access to residential proxy servers [74] and purchased numerous VPS/VPN servers for non-residential vantage points. We manually verified that app metadata collected using these vantage points was the same. We further confirmed this by collecting metadata from Google Play in select countries via a test Android app we developed. Given that these vantage points provided the same view of app metadata, we chose reliable non-residential vantage points whose advertised geolocation matched Google's detected location.

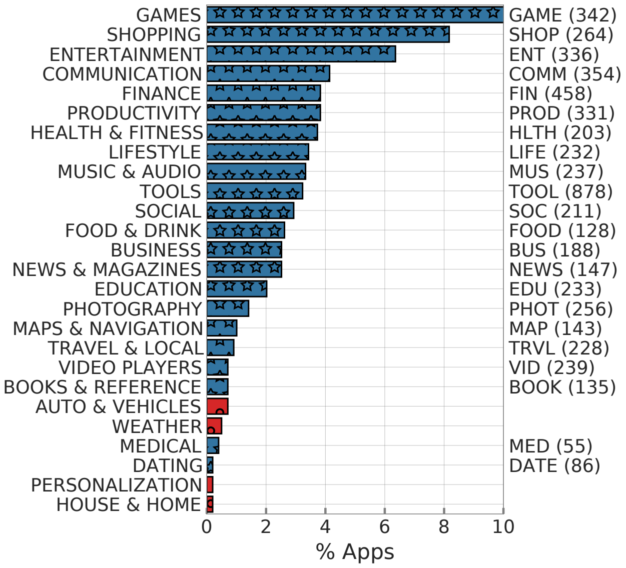


Figure 2: **Category Breakdown.** We compute the category breakdown of *top charts* of Google Play in 5 free countries. We study the top 20 categories and two additional categories MED and DATE (colored blue, full name on the left and abbreviation on the right with the numbers of apps per category).

Final Country List. Table 1 shows our final list of 26 countries, their geographic region, and FHIF scores. We dropped four of our initial 30 countries (Brazil, Ethiopia, Cuba, and Saudi Arabia) because of unreliable vantage points. We exclude China because Google Play is unavailable there.

Choosing App Categories. Prior research has shown that Google is a "superstar" market dominated by a few most downloaded apps [127]. Given this and the wide scale of our comparative study, we select only the top apps in the most popular categories. Google Play has 32 top-level categories, and for Games, 17 sub-categories [47]. We first compute a category-wise breakdown of the top charts (i.e., top 200 apps) on Google Play in the top five FHIF *Free* countries in our list (Australia, Canada, Germany, UK, and USA). Then, we pick the top 20 categories and two others—MED, DATE—which contain apps we expect to collect sensitive information. In all, we study 22 categories as shown in Figure 2.

App Selection. To curate our app list, we collect the metadata of the top 200 apps in each of our 22 categories. Combining the apps thus collected for the same five FHIF *Free* countries, we get 17,351 unique apps. From this, we choose

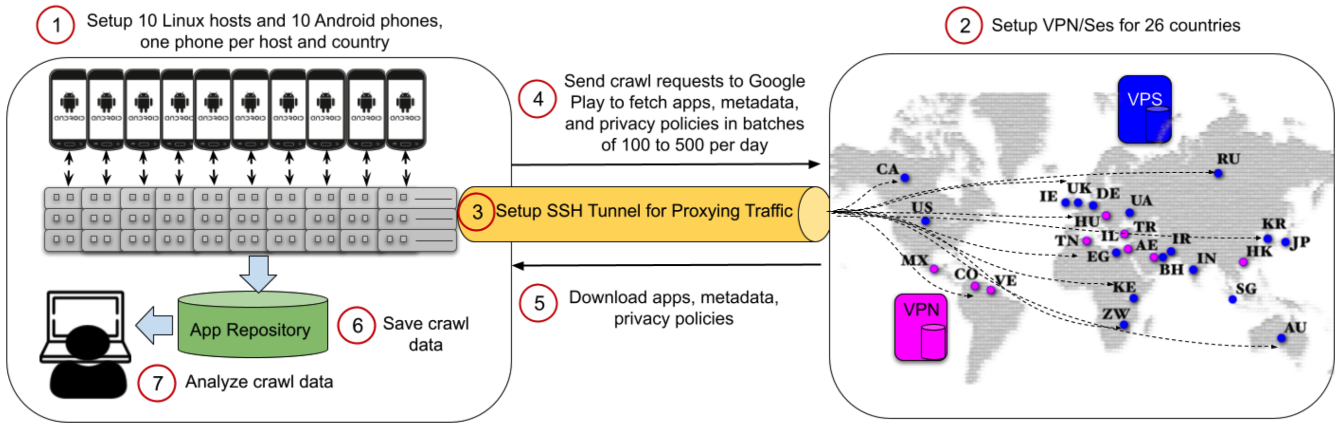

1 Setup 10 Linux hosts and 10 Android phones, one phone per host and country
2 Setup VPN/Ses for 26 countries
4 Send crawl requests to Google Play to fetch apps, metadata, and privacy policies in batches of 100 to 500 per day
3 Setup SSH Tunnel for Proxying Traffic
5 Download apps, metadata, privacy policies
App Repository
6 Save crawl data
7 Analyze crawl data
VPS
VPN
CA
RU
IE
UK
DE
UA
US
HU
TR
TN
IL
IR
KR
JP
EG
AE
MX
BH
IN
HK
CO
VE
KE
SG
ZW
AU

Figure 3: **Measurement Design and Setup.** Steps: (1) Set up ten Linux hosts and ten Google Pixel 2 phones, with one Google account for each country. (2) Set up VPN/Ses and SSH tunnels for downloads via VPSes. (3) Send crawl requests to Google Play to download in batches of 100–500 apps from all 26 countries, with each batch downloaded on the same day. (4) Save data to centralized repository for analysis.

popular apps based on their number of installs, and use 1 million installs as the popularity bar for all categories except Games. For Games, we use 50 million installs as the popularity bar, as 47% of the apps with at least 1 million installs are Games. This trimmed our app list down to 4,465 apps. Finally, based on researcher interest in security and privacy, we add 1,219 apps from a keyword search for *security*, *privacy*, *vpn*, *ad blocker*, and *crypto wallet* that belong to our selected 22 categories. Our final app list has 5,684 apps.

Measurement Design. Given the scale of our study, we want our measurement design to enable parallel downloads of apps from multiple countries. To eliminate inconsistencies from app updates during downloads, we conduct a preliminary longitudinal daily crawl of app metadata in 20 randomly chosen countries over 31 days and compute daily app updates. We find that only 1.1% of the apps have either a major or minor update per day on average and approximate the distribution of the percentage of apps updated on a given day to be N(1.1, 0.5) by the central limit theorem [85]. The median number of apps that have an update per day is 0. Based on this observation, we divide the 5,684 apps into batches of 100-500 apps and download each batch from every country within a 24-hour window. Doing so gets us a reasonable snapshot of an app within a batch in all 26 countries.

Data Collection. We download from ten countries simultaneously. For parallel downloads, we set up ten Linux measurement machines and ten Google Pixel 2 phones running Android 10, as shown in Figure 3. We set up these phones with one Google account per country. The Linux servers are set up with a *multipass* virtual machine [83], one per country, based on the vantage point's bandwidth and constraints of the host. To proxy download traffic, we set up an SSH tunnel to each VPS and use *proxychains4* as a SOCKS proxy [96]).

We download 5,684 apps and their metadata in batches varying from 100–500 apps depending on the network conditions from all 26 countries on the same day. On average, we download a batch within a 9.5-hour window between the earliest and latest country. We download all batches in 15 days (June 2020).

Checks & Precautions. We deployed several mechanisms to mitigate transient download errors. We randomize app downloads and use a pre-calibrated rate-limit for each country. We retry every failed download until two consecutive retry attempts for the same app return the same error. Using a separate Google account for each country eliminates the possibility of fetching cached content. We also clear the Play Store cache on the phones between downloads. For countries that use a VPN vantage point, we also check for run-time changes to geolocation every 50 app downloads. We manually verify download error logs each day. For network errors that are not transient, we additionally re-crawl a random set of apps each day to confirm failures. We discard the day's download for all countries if a country is unreachable on a given day.

Scraper Implementation. We customize two scrapers to download an apk and its metadata from Google Play. The Google Play scrapers are Python modules—*PlaystoreDownloader* and *google-play-scraper*—that send HTTPS requests to a URL containing an app's ID (as described in section 2) [54, 94]. The apk scraper requires the device's ID and Google account credentials. To scrape privacy policy, we first get an app's privacy policy link from its metadata and use a Selenium crawler to download the policies [104].

Data Extraction. Considering our goal is a large-scale geographic study of apps, we only examine an app's static features. We use *aapt* and *aapt2* [13] to extract an app's permissions, version, and encrypted communication settings. In cases where *aapt* fails, we use *apktool* [20] to get decoded data from the apk. For app signature, we combine

Error Code	Error Message	Error Reason
Err1	Item not found	**Takedown**
Err2	Google Play purchases are not supported in your country. Unfortunately you will not be able to complete purchases	**Developer-blocking**
Err3	Your device is not compatible with this item	Device-targeting
Err4	This item is not available on your service provider	Not registered with the service provider
Err5	The Play Store application on your device is outdated and does not support this purchase	Missing payment and billing info
Err6	The item you were attempting to purchase could not be found	Possibly incorrect device location or billing setup error by developer
Err7	Caused by SSLError (bad handshake)	Possibly network interference

Table 2: **Download Errors.** This table shows the error messages observed from failed app download requests to Google Play and the error codes we assign. Error reasons are our findings that clarify why these errors occur, specifically whether it is a takedown or developer-blocking.

results from *keytool* [68] and *apksigner* [13]. For extracting third-party libraries, we use two tools used in prior research, `libradar` and `ExodusPrivacy` [4, 35, 55, 75]. `libradar` returns a mixed list of third-party libraries present in an app. Given there is no clear way to filter ad libraries from that list, we use `ExodusPrivacy` specifically to collect ad and analytics libraries. To extract policy text from a privacy policy web page, we use *ReadabiliPy* and *Beautiful Soup* [101, 102].

Ethics. All the data used for this study is collected using our own mobile devices with traffic proxied through non-residential VPN/Ses we purchased. For all purchases, we used our real identities and agreed to comply with the terms and conditions of the hosting provider. In our preliminary testing to validate vantage points, we developed and shared a mobile app (including its source code) with our collaborators (all security researchers) to fetch app metadata from Google Play. The app collected no personal or device data. Our collaborators gave us informed consent and the collected data was discarded after successful tests. We reported all our findings to Google.

4 Data Characterization & Exploration

Using our measurement testbed, we successfully downloaded 5,385 of our initial list of 5,684 apps from at least one country. In all, we collected 117,233 app instances and 112,607 privacy policy instances from 26 countries. These apps are from 22 different categories, with 5,667 free and 17 paid apps, 2,365 apps that support in-app purchases, and 1,120 apps that provide content rating descriptions. The majority of these apps (86%) were last updated in 2020, indicating that most apps are actively maintained. We leverage this large-scale and diverse app data to investigate geodifferences. To the best of our knowledge, ours is the most extensive multi-country app collection in the research community.

While we successfully downloaded a large portion of the app instances from our vantage points, surprisingly, we discovered many apps that failed to download, with 299 apps that failed to download from any country. The download error messages varied for these apps. Given that there is limited knowledge of why these errors occur, we perform in-depth exploratory research to characterize these failures.

4.1 Characterizing Download Errors

While downloading the apps, we observed various error messages that range from “Item not found” or “purchase not supported in your country” to “device not compatible”. Table 2 shows all the error messages and our assigned error codes. These error messages from the failed download requests to Google Play are opaque and do not clearly communicate why the download fails. As noted in subsection 2.1, we are specifically interested in discovering whether the errors result from a non-compliance takedown, a government-requested takedown, or developer-blocking. Therefore, we perform control experiments using apps that are publicly known to be unavailable for the above reasons.

We find that our control apps that are known to be taken down by Google fail with Err1. The error is the same for both government-requested takedowns (e.g., LinkedIn in Russia), and non-compliance takedowns (e.g., Luna VPN - `com.luna.vpn` [106])[1]. However, for government-requested takedowns, Google removes the app only from the country issuing the order. In contrast, Google’s non-compliance takedowns are applied to all 26 countries, confirming that such takedowns are global. Of course, if the developer decides to unpublish (remove) an app from everywhere, that also returns Err1 (global developer takedown), though we believe such occurrences are rare given our data consists of globally popular apps. We verify this by unpublishing our test app.

For the control apps that are known to be developer-blocked (e.g., SkypeLite in India [82]), we find Err2. We verify this by publishing our test app on Google Play and confirming Err2 when downloading from countries where we manually chose to block the app. We observe Err3 when some of our control apps fail to download because of developers device targeting their apps (e.g., Samsung’s Secure Folder). Our apps that are carrier targeted (e.g., Sprint Music) fail to download with Err4. All paid apps fail with Err5 because Google Play requires a user’s payment and billing information, which we did not set up. Finally, a few apps fail with Err6 in select

[1] Our characterization for takedowns rely on Google’s transparency report. Google uses “government requests” as an umbrella term for content removal requests from the following requesters: judicial, executive, suppression orders, consumer protection authority, information and communication authority, court order directed at Google, government officials, and others.

countries. Though Google's official documentation provides no insights, public reports (e.g., Google discussion groups) suggest that this error is due to incorrect billing set up by the developer or conflict in the device's country settings with a user's actual location [52, 107].

Error Characterization for 299 Unavailable Apps. Considering that we seed our initial list from app metadata in five countries with high FHIF scores, it was surprising that 299 apps (5.3%) failed to download from all of our 26 countries for the following reasons. Using the error characterization from our control experiment above, we find that 143 apps failed to download with Err1, suggesting either Google's non-compliance takedowns or global developer takedowns. 126 apps fail with Err3 because of incompatibility with our measurement device. All 17 paid apps fail to download with Err5, and four apps specific to a service provider fail with Err4. One app, `it.mirko.transcriber`, is the only app that is developer-blocked (Err2) and is flagged "early access" by the developer indicating its limited release [49]. The remaining eight apps fail due to takedowns in some countries and developer-blocking or "not found" in others. Subsequent download checks show that they were removed from Google Play, indicating that Google or the developer may have been in the process of removing these apps entirely. We exclude all 299 apps for the remainder of this study.

SSL Errors in Tunisia. Tunisia was unique among our countries in that a large number of apps (1,819), but not all, consistently failed to download with SSL bad handshake error (Err7), even on repeated attempts. We confirmed that these errors are not because of client and server misconfigurations through a series of tests, which we describe in Appendix B. We treat these apps that failed with SSL bad handshake error as unavailable in Tunisia for this study.

Characterizing Availability Inconsistency. From a user's perspective, an app is available if: (1) its Google Play homepage (or metadata) has a download link; and (2) the actual `apk` is downloadable. Our data shows that there are apps that have a download link yet are not downloadable. For instance, in Tunisia, 1,948 apps that have a download link in metadata could not be downloaded. The difference is less in the remaining countries, with between 101 apps in Iran and 154 apps in the US not downloaded, despite having a download link. We flag these apps as unavailable in our study. Conversely, we also observe 13 apps (e.g., Google Pay, YouTube) that have no download link in Iran, yet the `apk` is downloadable by going to its URL directly. We consider these apps as unavailable in Iran as a user cannot access them. These observations point to a bigger issue when reporting on app availability—we find that measurements that only use app metadata consistently overestimate availability. A previous study [119] on app censorship measured availability based on app metadata alone, which we show is insufficient. For our study, we consider both conditions—a user's access to an app's download link and the `apk`.

4.2 Characterizing Security and Privacy

Security Features. In this section, we outline the security choices made by app developers in three of Android's notable security features—permissions, settings for encrypted communication, and app signature. We extract permissions from our collected apps and classify them into one of the four Android protection levels based on Android 10's source code and developer documentation [8]. Collectively, our apps request 4,816 unique permissions, of which only 229 are core Android permissions. Of the 229 core permissions, 32 are *Dangerous*, 67 are *Normal*, 53 are *Signature*, and 77 are *SignatureOrSystem* permissions. The remaining 4,587 permissions are custom and vendor-specific permissions, which we exclude as there is no clear way to assign a protection level to them.

Regarding encrypted communication, 2,825 apps (52%) explicitly disable encrypted communication in at least one country through the app's network security configuration file. The remaining apps either explicitly enable encrypted communication or use the system's default settings, which may be set to unencrypted communication depending on the Android version. Regarding app signatures, only two apps are signed using multiple signature schemes, as recommended by Google. While more than 70% of apps use the V2 signature algorithm, about 12% (663) of apps use V1, which is known to be vulnerable [122, 125]. Surprisingly, two apps also use Android's Debug certificate, which is not accepted by most app stores [12]. Apps also use deprecated hashing methods and weak key lengths [86, 87, 122, 125]; e.g., 2,358 apps use SHA1 with RSA, and 1,735 use 1024 bit keys with RSA.

Third-party Libs. We examine the third-party libraries in our apps, focusing on ad trackers. From the list of 400 trackers that `Exodus Privacy` looks for, we find a total of 274 unique trackers that are globally popular. We find 407 apps with no ad trackers (e.g., `org.torproject.android`). Consistent with existing reports [21, 33, 81], the top ad and analytics libraries in our apps are Google's—Firebase Analytics, AdMob, CrashLytics, Analytics—and Facebook's—Login, Share, Analytics, Ads, Places. Excluding ad trackers, the top five third-party libraries in our apps are `com/google/android/gms` (Google Mobile Services), `android/support/v4` (Android support), `com/google/gson` (JSON conversion), `okhttp3` (HTTP client), and `com/google/zxing` (Barcode processing).

Privacy Policies. We use a semi-automated approach to characterize the 112,608 privacy policies collected from 26 countries. A manual perusal of these policies shows that many downloaded pages are, in fact, error pages. To disambiguate an error (block) page from a valid policy page, we follow the approach taken by prior web censorship research [66, 80]. Using word count as a measure of page length, we study the distribution of word counts of all pages and find a threshold of 200 words reasonable to separate an error page from a policy web page. We further look for privacy-specific keywords in these pages, as we expect error pages to omit them. After excluding identical policies of an app, we apply the above heuristics on the remaining 17,025 policies and get 4,234 error pages. From the remaining non-error pages, we extract 8,737 English policy pages using Python's `langdetect`. We manually verify the disambiguated error and policy pages.

Besides finding error pages instead of privacy policies even where the app downloads, we observe other violations. For instance, 76 apps have no privacy policy link on their Google Play homepage, and another 56 apps have broken links or expired domains, all of which are explicit violations of Google's policy [45]. About half of these apps request *Dangerous* permissions, suggesting no disclosures on an app's access to sensitive information.

5 Results on Geodifferences

After characterizing and cleaning up our data, we investigate the prevalence of geodifferences in our 5,385 apps that are popular on Google Play. First, we discuss geoblocking, a type of geodifference in which an app is blocked to users from a particular country or region. Then, we describe another form of geodifference where an app's security and privacy offerings may vary based on the region.

5.1 Geoblocking

Even though Google Play is the most accessible app market, our data still suggests geoblocking in all 26 countries in our study. Of the 5,385 apps, 3,672 apps are geoblocked in at least one country. Iran (IR) and Tunisia (TN) have the highest blocking by a wide margin, with 2,256 and 2,681 apps geoblocked. Surprisingly, geoblocking varies in the other countries, with 300 apps blocked in the US to 800 apps blocked in Zimbabwe (ZW), indicating high variability based on the region. Compared to prior geoblocking investigation on the web [80], the blocking we observe is significantly high, even though our data consists of apps with millions of users globally. On the web, the maximum observed geoblocking was 71 domains from the Alexa top 10K sites in the most affected countries [80].

We next investigate the availability of each app across countries for all apps to detect potential blocking trends. We first compute the similarity between two countries C_i and C_j, as:

$$S(C_i,C_j) = \frac{|G_{C_i} \cap G_{C_j}| + |\sim G_{C_i} \cap \sim G_{C_j}|}{\bigcup_i \sim G_{C_i}} \qquad (1)$$

where, for country C_i, G_{C_i} is the set of apps that are geoblocked and $\sim G_{C_i}$ is the set of apps not geoblocked, and $\bigcup_i \sim G_{C_i}$ is the number of apps not geoblocked in at least one country (here, 5,385). We then use Ward's minimum variance method [67] to perform agglomerative clustering with the Hamming distance between countries computed as $S(C_i,C_j)$.

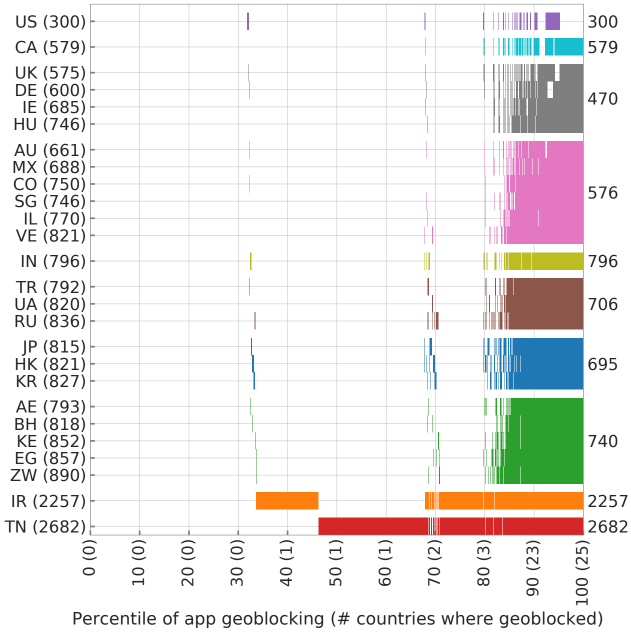


Figure 4: **Geoblocking per Cluster**: The countries are grouped by cluster, with geoblocking per country on the left axis and the intersection of geoblocked apps per cluster on the right axis. Each app is a data point on the x-axis (percentile when sorted by the number of geoblocked countries) shown in solid colors when geoblocked.

Figure 4 shows geoblocking in the 26 countries grouped by cluster, with the blocking per country on the left axis and the intersection of geoblocked apps within a cluster on the right axis. We find that the African countries experience the most geoblocking. Countries within the same region (e.g., the four EU countries) tend to have the same apps geoblocked, though minor variations exist. IR, TN, US, Canada (CA), and India (IN) are outliers and not clustered with any region.

We further investigate whether there exists a correlation between a country's app availability with its FHIF score and GDP using Spearman's rank correlation coefficient metric [41]. We find a moderate negative correlation (ρ=-0.58, p-value = 0.002) between a country's app availability and GDP. On the other hand, there is a moderate positive correlation (ρ= 0.64, p-value=0.001) between app availability and FHIF score. Despite the FHIF score's positive correlation with availability, we observe a few exceptions. For instance, CA, which has a higher FHIF score, has more geoblocking than the US and UK. Turkey (TR) and UAE (AE), which have very low FHIF scores, have higher availability than Japan (JP) and South Korea (KR). These exceptions could be because Freedom House, as far as we know, focuses only on government blocking of social media and communication apps [40].

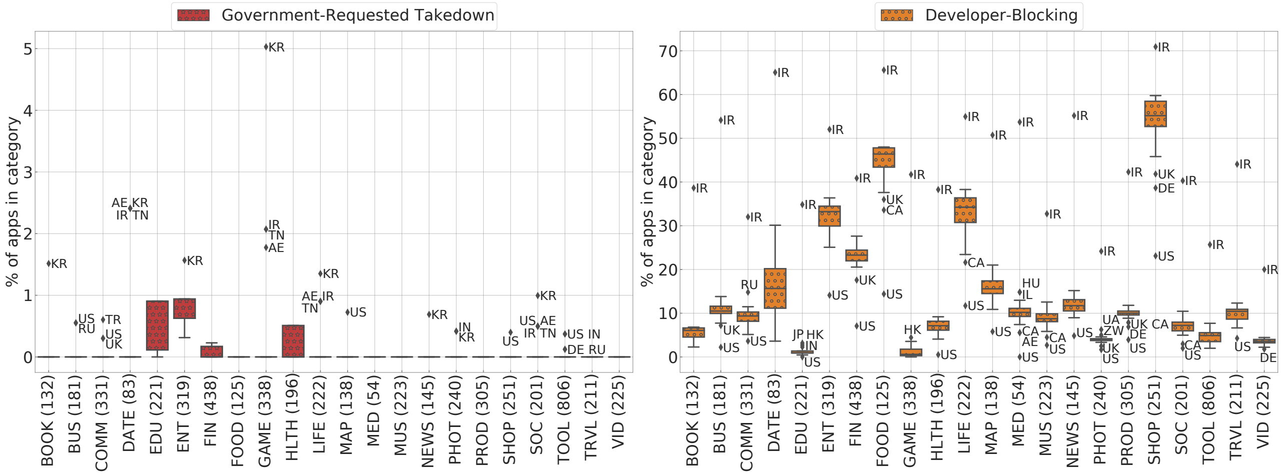


Figure 5: **Developer blocking vs. takedown category–wise (%).** Each figure shows the country distributions and outliers in each category (with their total apps) for takedowns on the left, and developer blocking on the right. KR is a frequent outlier in takedowns, while IR is the most developer-blocked in all categories. Categories like FOOD, SHOP, and LIFE tend to be naturally region-specific and thus blocked more.

5.2 Who is Blocking?

While the statistics on geoblocking are valuable, we also want to know who is behind the blocking and the plausible reasons for such blocking. In this section, we are specifically interested in government-requested takedowns and developer-blocking of apps that are available in at least one country.

Government-requested takedowns. We observe that 61 unique apps are subject to government-requested takedowns (e.g., `com.truecaller` in TR, `mobi.jackd.android` in KR, and `com.mi.globalbrowser.mini` in the US). KR, specifically, has the most government-requested takedowns with 36 apps removed. The US has more than the median number of takedowns, while many countries with lower availability, particularly ZW, Bahrain (BH), Kenya (KE), and Egypt (EG), have the lowest of such takedowns.

We find a case of a government-requested takedown due to a violation of regional content regulation. For instance, in KR, 36 apps are taken down, of which 17 are gambling and game apps. This finding is consistent with reports of the Korean government's aggressive policy to block the distribution of apps with adult content, violence, or gambling [46]. Interestingly, KR is also an outlier in the categories DATE, ENT, LIFE, NEWS, and SOC, though to a lesser extent.

Certain app categories see more takedowns even when no government regulations appear to be violated. For instance, COMM apps encounter more takedowns in TR, and PHOT apps in IN. While countries with higher than the median geoblocking such as IR, RU, and TN are outliers in certain categories such as DATE, GAME, LIFE, surprisingly, countries such as the US, UK, and DE with lower geoblocking rates also are outliers in others such as BUS, TOOL, SHOP. Figure 5 (left) shows the distribution of government-requested takedowns by category and the outliers in each.

Developer-blocking. Compared to government-requested takedowns, we find that the proportion of apps that are developer-blocked is significantly higher in all countries and app categories, and has the most influence on geoblocking in the mobile domain. 2,419 (44.9%) unique apps are developer-blocked in at least one country (e.g., `free.vpn.unblock.proxy.turbovpn` in Hong Kong (HK), `com.google.android.apps.books` in Israel (IL), `com.twitter.android.lite` in the US). Consistent with prior web geoblocking study [80], IR is the most developer-blocked country with more than 50% blocking in eight categories and is the top outlier in every category.

While overall, countries within the same region (and hence, cluster) tend to have the same apps geoblocked, a closer look within a region shows different amounts of developer-blocking. For instance, the EU countries, UK, Germany (DE), Ireland (IE), and Hungary (HU) have 559, 580, 666, and 725 apps blocked, respectively. Examples of apps that are blocked differently in the EU countries are Paypal's `com.izettle.android`, which is blocked only in IE and HU, `com.lego.catalogue.global` that is blocked in the UK, IE, and DE, and `com.google.android.apps.walletnfcrel` and `com.yahoo.mobile.client.android.search`, both of which are blocked only in HU. Such region-specific differences in blocking may be a result of either local laws or consumer market segmentation for business.

We find 8 apps that are developer-blocked only in our four EU countries, possibly due to GDPR legislation. For instance, the news app `com.usatoday.android.news`, the largest gay social networking app, `com.blued.international`, and `com.ebates` are developer-blocked only in all four EU countries. While Facebook's Messenger for Kids is known to be geoblocked in our EU countries due to GDPR compliance

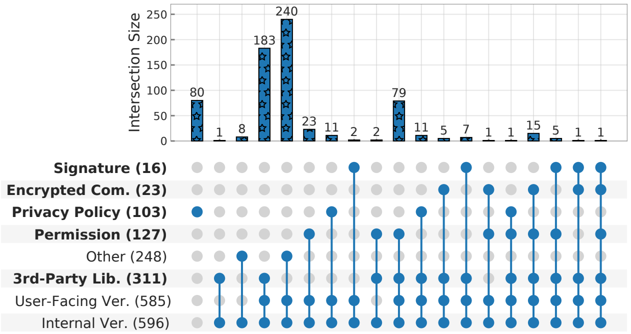


Figure 6: **Geodifferences in App Features.** The app feature and the number of apps with geodifferences in the specific feature is shown on the left. The number of apps with intersecting geodifferences (indicated by the dark circles) are annotated above. "Other" indicates features outside of the security and privacy related features in bold.

issues [115], we find that it is blocked in other countries like Ukraine (UA), Israel (IL), HK, RU, and KR as well.

Certain countries have more developer-blocking in certain categories. For instance, the African countries in our study—KE, ZW, TN, and EG— appear in seven out of the top 10 developer-blocked categories. JP has the most developer-blocking in VID and EDU, RU in COMM, and HU in NEWS apps. Consistent with prior research on the web [80], SHOP and FOOD apps are the most developer-blocked everywhere, possibly because these apps are often region-specific and involve financial transactions in local currency. However, contrary to that work, we find that EDU and MED are the least developer-blocked. Figure 5 (right) shows the distribution of developer-blocked categories and the outliers in each.

Countries also vary in the developer-blocking of the special-interest apps (security, privacy, ad blocker, crypto, and VPN) we study. The top three countries that have the most developer-blocking of these apps are IR, ZW, and HK with 298, 79, and 78 apps blocked, respectively. While prior research has noted high blocking of VPN apps in RU [119], we find HK and TR (25 and 17 apps) have higher blocking of VPN apps than RU (15 apps). The higher blocking in HK and RU of VPN apps is consistent with the recent upsurge of surveillance laws there [59, 117].

Although our data consists of free apps, some support in-app purchases that may cause US sanctions to play a role in geoblocking in embargoed countries like IR, ZW, and Venezuela (VE). Amongst the embargoed countries, IR has much higher blocking when compared to ZW and VE. While Google prevents in-app purchases in IR, they are supported in VE and ZW [48]. Despite this, we find that developers disable in-app purchases in all apps in VE and ZW.

5.3 Geodifferences in Security and Privacy

Until now, we studied the phenomena of geoblocking in the mobile app ecosystem. In this section, we go beyond and ask whether the apps available in more than one country have geodifferences that lead to differences in security and privacy offerings to users in a region. To the best of our knowledge, we are the first to conduct such a study.

Despite using a measurement design that ensures app updates to be rare, we observe geodifferences in 596 apps as seen by a binary diff of our `apks` across countries. To confirm that the observed number of apps with geodifferences is statistically higher than the expected number of app updates, we conduct a z-test by considering each app as a sample and the presence of geodifference in an app as a binary value. With this, we define the null hypothesis as the average percentage of apps with geodifferences (11.1%, here) to be less than or equal to the average percentage of apps updated on a given day (N(1.1, 0.05) from preliminary experiments in section 3). The resulting p-value of 0 is less than the significance level 0.05, which allows us to reject our null hypothesis and confirm a significantly large number of apps with geodifferences.

The presence of apps with geodifferences is consistent with Google's country targeting feature that allows developers to distribute different versions to different countries [43]. Since we expect apps with geodifferences to also have version differences, we study app versions and find that not all of these apps have differences in an app's user-facing version. While all 596 apps have differences in the internal versions, 11 apps have the same user-facing versions in all available countries[2].

Figure 6 shows the distribution of geodifferences in app features. We find apps with geodifferences in privacy policies (sometimes even when their `apks` have no geodifferences), permissions, signatures, settings for encrypted communication, third-party libraries, and in other app features such as assets and app components. Below, we focus on geodifferences in security and privacy related features in more detail. While overall, the geodifferences appear to be more prominent in certain countries, we did not observe regional differences related to localization.

Permissions Requested. We compare an app's permissions in all 26 countries and determine the number of additional requested permissions in each country. The number of extra permissions requested by an app a in a country is computed as $|P_{ac} - P_a|$, where P_{ac} is the set of permissions requested by the app in country c, and $P_a = \cap_{i \in Countries} P_{ai}$ is the intersection of requested permissions in countries where the app is available.

We found 127 apps that exhibit geodifferences in permissions requested. On average, the most frequently requested extra permissions are `READ_EXTERNAL_STORAGE` and `READ_PHONE_STATE`, both *Dangerous*, and `RECEIVE_BOOT_COMPLETED`, which is *Normal*. We find 49 apps that request *Dangerous* permissions only in certain countries. For instance, the app `com.fun.top.video` has two *Dangerous*

[2] There are two app versions in an app's manifest—a version that is displayed on the app's home page on Google Play, which we call the user-facing version, and an internal developer version.

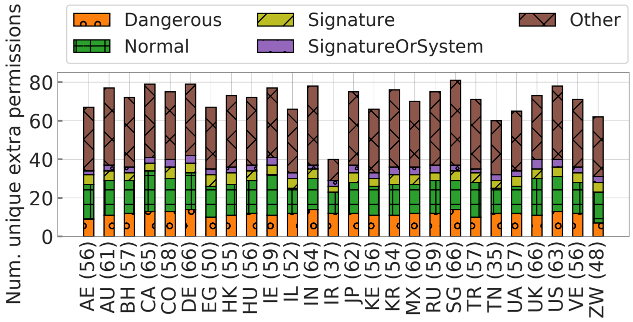


Figure 7: **Extra Permissions**– The Y-axis shows unique number of extra permissions requested of each type. X-axis paranthesis shows # apps requesting extra permissions in each country. Most apps only request a handful of extra permissions, the median being four. "Other" indicates custom or vendor specific permissions

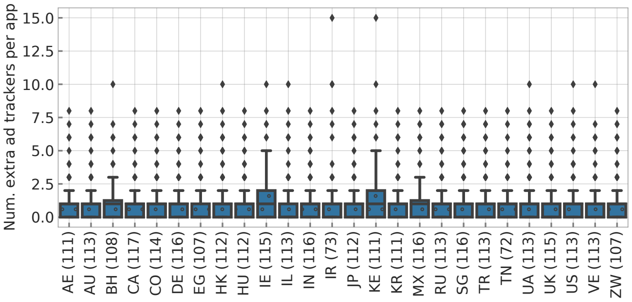


Figure 8: **Extra Ad Trackers.** The number of apps with geodifferences in ad trackers per country is shown in parenthesis on the X axis. While the majority of apps only include one extra ad tracker in certain countries, there are outliers present in all countries, with one app including 15 extra ad trackers in both IR and KE.

permissions (ACCESS_BACKGROUND_LOCATION and CAMERA) only in AE, BH, IR, and TN. com.honor.global has one extra permission ACCESS_FINE_LOCATION only in DE and UK. On average, we observe apps in BH, TN, CA, and DE to request the most extra *Dangerous* permissions (e.g., READ_EXTERNAL_STORAGE and READ_PHONE_STATE). Figure 7 shows the breakdown of extra permissions of each permission type by country.

We analyze app categories that request extra *Dangerous* permissions. We find that five categories request extra *Dangerous* permissions most often—LIFE, VID, DATE, ENT, and TOOL. While it makes sense for TOOL and VID apps to request *Dangerous* permissions, it is surprising that DATE and LIFE apps request extra *Dangerous* permissions, especially only in certain countries. For instance, com.datemyage and com.netatmo.camera, request READ_EXTERNAL_STORAGE and RECORD_AUDIO permissions in only 11 and 3 countries, respectively.

Third-party Libs. Similar to how we compute the number of extra permissions requested, we compute the number of extra third-party libraries in an app per country, focusing on ad trackers. We found 118 apps with additional ad trackers, with the top five most included ad trackers being Integral Ad Science, Moat, and Facebook's—Analytics, Places, and Share. On average, IR has the most extra ad trackers, followed by KE and UA. Figure 8 shows the geodifferences in extra ad trackers per country. There are outlier apps in every country, notably in IR and KE where the app com.outfit7.movingeye.swampattack includes 15 additional ad trackers. com.uc.vmate and com.mapfactor.navigator request 10 additional ad trackers each, the former in IE and UA, and the latter in BH, HK, IL, KE, US, and VE. The median number of extra ad tracker per country is one (e.g., com.shareitagain.bigemoji includes AdColony only in DE and IN). The top categories with the most extra ad trackers are GAME (e.g., com.outfit7.movingeye.swampattack), ENT (e.g., com.graymatrix.did), and SOC (e.g., messenger.chat.social.messenger.lite). Overall, on average, TN, AE, and UA have the most extra third-party libraries.

Encrypted Communication. We find 23 apps that selectively use unencrypted communication settings for some countries. For instance, com.honor.global (which has two additional *Dangerous* permissions only in UK and DE) uses encrypted communication only in DE and UK. Three of the apps that have geodifferences in this communication setting are VPN apps. For instance, com.vpnproxy.connect uses unencrypted communication only in UK, TN, and VE.

Signature Algorithms. We find that 16 apps have geodifferences in the signature algorithms used, suggesting that some apps in certain geolocations are at a higher security risk from using weak signatures. For instance, com.thomsonreuters.reuters, which is available everywhere except TN, is signed with just the signature scheme V1 in 16 countries, including the US, CA, and DE, and uses the V2 scheme in other countries. ca.bell.selfserve.mybellmobile uses V1, V2, and V3 signature schemes everywhere, except in JP and MX, where the app does not use the V3 signature scheme. This is against Google's recommendation of signing the apps with all three signature schemes for maximum security [12].

Privacy Policy. We find 103 apps with geodifferences in privacy policies. Our data shows that regional legislation such as the GDPR in the EU and the CCPA in the US [63, 108] actually have a positive influence on app developers. However, we also find that countries not covered by CCPA or GDPR have a higher privacy risk. For example, 71 apps from Google have additional clauses to comply with GDPR only in our EU countries and for the CCPA only in the US. While the US policy of gbis.gbandroid was updated recently in 2020 to comply with CCPA, the privacy policies in countries with older legislation (e.g., AU - Privacy Act 1988) were last updated in 2017 [31]. Norwegian company Opera has six apps, including `com.opera.browser`, with policies that use two different data protection standards: European in the European Economic Area and Singaporean in other countries.

We find instances of privacy policies that fail to download even when the apps themselves are available. Privacy policy downloads of 37 apps returned 403 Forbidden errors (at least once in all countries), and for another 20 apps, returned explicit server-side blocking error pages (at least once in 21 countries). Notably, 12 policies hosted on Google's App Engine (GAE) did not download in IR because GAE is blocked due to embargo rules there [77]. We could not download the privacy policy for `org.openobservatory.ooniprobe` (hosted on ooni.torproject.org) in TR, where Tor is known to be blocked [91] and `org.telegram.messenger` in IR, where Telegram is blocked [38]. Surprisingly, we received the IR government blockpage for policy URLs of four apps (e.g., `com.geekslab.applockpro`).

While we expect privacy policies to differ across regions when apps request different *Dangerous* permissions, we found 28 apps with identical privacy policies despite having geodifferences in requested *Dangerous* permissions. Additionally, not all privacy policy URLs point to a valid policy page as required by Google. For example, the policies of the apps `com.jagbani` and `net.bitburst.pollpay` link to a company website and to a deactivated page, respectively.

6 Limitations & Future Work

Our work is the first large-scale data-driven study into the prevalence of geodifferences in the mobile app ecosystem. We chose to look at the popular apps where we expected developers to put effort to maintain the apps. Any geodifferences observed in these apps greatly impact users because of their popularity (and we found many). There may be region-specific trends that can be discovered by further studying the long tail of less popular apps on Google Play.

While we contribute significantly to characterizing geodifferences and who is responsible for them in the mobile app ecosystem, we are limited in our study on why these geodifferences exist. Prior work on web geoblocking summarizes the motivations for geoblocking to be one or more factors such as data protection law, economic sanctions, revenue, national security, political censorship, or unintentional [118]. They emphasize that distinguishing between these motives is hard given how they are interdependent on each other. Thus, future work could include focused investigations to distill the reasoning i.e., from financial to data protection or others, and their impacts on users, or study popular regional apps itself to understand that. Currently, we only examine an app's static features, and future studies could use results from runtime behavior analysis of apps within a region.

Though our study is focused on Google Play, future work could do a broader study that includes other app markets. Our study is limited to 26 countries and 5,684 popular apps. While we consider our choice of countries reasonable for a first geographic study, additional countries could provide more insights. We use the FHIF score as a metric for country selection following prior work on the web but acknowledge a potential bias in the metric [109]. Hence, we do not draw any large conclusions based on the metric.

7 Discussion

We focus on Google Play because it is the largest and the most accessible app market. Our findings suggest differences in app equity as a result of geoblocking and geodifferences in apps. We find certain countries experiencing more blocking than others, alarming cases of the same apps asking for different permissions in different countries, and overt violations of privacy disclosures in some countries, to name a few. While Google has taken some steps towards a transparent ecosystem [9], the geodifferences that we observe are not generally known due to lack of research in this space. Our work highlights the shortcomings in Google's auditing of the app ecosystem.

There are a few steps that Google and other global app market proprietors (e.g., Amazon Appstore) could take to address some of the issues we find. For instance, Google could consider pushing for transparency from developers to specify regional differences in app features, including permissions, ad trackers, and privacy policies. Google could also do better testing for an app's compliance with its existing guidelines (e.g., on privacy policy and signatures) across countries since our study discovers both non-compliance and geodifferences for the same app. Considering that auditing external links are hard, Google could host an app's privacy policy themselves to track policy changes. Google could provide an app's release history (as provided by some app markets like APKMirror and Apple's AppStore), which could help audit developer behavior in the ecosystem.

Developers on Google Play have fine-grained access to country and device targeting features. Prior web geoblocking studies have shown how an unrestricted country targeting feature provided by Cloudflare, a web CDN, led to significant blocking, isolating certain countries more than the others [80]. In 2018, Cloudflare reverted to its older business model that limited country targeting only to their enterprise customers. In the mobile app ecosystem too, we note certain countries (e.g., Iran) that are more isolated than the others, even though the apps we study are all free apps and popular apps. We believe that some of the geoblocking is partly because developers have unrestricted access to country targeting features on Google Play. Global centralized app market owners like Google can prevent exacerbating this divide.

Recently there have been several instances of governments relying on Google Play to ban content. In June 2020, the

Indian government banned 59 Chinese apps [114]. Per our longitudinal metadata, Google took down the apps in India on July 2, 2020, less than a week after the government issued a public notice. A similar pattern emerged in the US when the government threatened to ban Chinese apps like Tiktok [112]. These were high-profile takedown threats. But, our work shows that takedowns can also happen without much public knowledge or debate. As a platform owner, Google could provide better insights on why they removed each app in its transparency reports [44] and clearly disclose the reason an app is not available in a country on the app's homepage. Detailed transparency reports coupled with informative error messages (as in the web ecosystem) will enable researchers and third parties to audit the app market for app equity and for citizenry to be better informed on reasons for unavailability.

Potentially as a result of Google's policies in some countries, regional app markets are becoming increasingly popular. For instance, Google's approach towards Iranian apps and developers [36, 116] has likely contributed to the popularity of local app markets like Cafe Bazaar in Iran. We find that some apps that are developer-blocked on Google Play are available on Cafe Bazaar, potentially driving users to the local app market. What is concerning here is that these local app markets may not necessarily have adequate app vetting policies. Moreover, app market owners may provide altered versions of apps to users; a user has no straightforward way to distinguish an altered app from a legitimate one.

8 Related Work

Geoblocking has garnered much attention from regulating organizations in recent years. In a bid to curtail discriminatory practices by service providers through geoblocking, the EU commission introduced regulations in 2017 to prevent unjustified geoblocking, and foster an unfragmented digital market [34]. A study by the Australian parliament in 2013 found exorbitant prices in Australian markets due to geoblocking and concluded that geoblocking should be regulated [26].

Prior research has studied geoblocking in the web ecosystem. Tschantz et al. conducted a preliminary study on the motivations for server-side geoblocking and showed that root-causing the reasons for blocking is non-trivial [118]. McDonald et al. performed a wide-scale measurement study on geoblocking by customers of web CDNs by using Cloudflare's CDN as a case study [80]. Afroz et al. used a combination of automated page loads, manual checking, and traceroutes to confirm geoblocking of developing countries [1].

Geoblocking by mobile app markets has been studied at limited scale. Ververis et. al [119] looked at censorship of 11 censorship-circumvention apps in Russia and China by querying public search engines of app markets (Google, Apple, Tencent). In a recent study on digital filtering in Saudi Arabia, authors noted an increasing availability of 18 mobile apps from none in 2017 to 93% in 2019 [3]. Some studies on Apple's and Google's app markets examine app admissions and removals to understand their reasons for such decisions [22, 58, 70, 76, 121], but do not discuss geoblocking or geodifferences in app features.

Prior research has looked at finding privacy policy violations by analyzing policy texts and apps [6, 7, 126], characterizing and querying privacy policies [56], and studying differences in the policies of free and paid apps [55]. Sun et al. showed that policies generated by automated privacy policy generators (APPG) are often incomplete [110]. Longitudinal studies of web privacy policies show that longer policies are slow to comply with recent legislation such as the GDPR [5, 30, 71]. Shen et al. conducted a study on the types of sensitive information that leak from mobile apps [105].

Research on app markets such Google Play and Tencent has studied their app auditing processes, transparency efforts, impact of app releases, download distribution of apps, app monetization schemes, and behavior of app developers [65, 78, 79, 93, 120, 123, 127]. Lim et al. [69] conducted a large-scale survey on mobile app user behavior that affects app market downloads. Other large scale studies examined update behavior of apps on Google Play [113], update timing delay [88], update frequency [95] and target fragmentation as a result of developers targeting older Android versions [84].

9 Conclusion

We performed the first large-scale study of geodifferences in the mobile app ecosystem as seen by users in 26 countries. We designed and implemented a parallel, semi-automatic measurement testbed using which we collected 5,684 popular mobile apps from Google Play in our countries using direct measurement vantage points. Our data showed high amounts of geoblocking in all 26 countries. While we corroborated anecdotal instances of takedowns due to government requests, we found that blocking by developers had the most influence on the geoblocking. We also found instances of developers that release different app versions to different countries, with some apps having weaker security settings and privacy disclosures. Based on our findings, we provided recommendations for app market proprietors to address the issues we found.

10 Acknowledgement

We thank our anonymous reviewers for their valuable feedback on our work. We also thank our collaborators Sreeram Rajendran, Radhesh Krishnan K., Sreesh Kishore, and Naihao Deng, for their help during the initial testing phase of our tool and their insightful discussions. Shoutout to Aryana Ensafi Halderman, who accompanied us throughout this work, slogging through endless brainstorms and crazy timelines.

References

[1] S. Afroz, M. C. Tschantz, S. Sajid, S. A. Qazi, M. Javed, and V. Paxson. Exploring server-side blocking of regions. https://arxiv.org/pdf/1805.11606.pdf, 2018.

[2] A. Akhavan Niaki, S. Cho, Z. Weinberg, N. P. Hoang, A. Razaghpanah, N. Christin, and P. Gill. ICLab: A Global, Longitudinal Internet Censorship Measurement Platform. In *IEEE S&P*, 2020.

[3] F. Alharbi, M. Faloutsos, and N. Abu-Ghazaleh. Opening digital borders cautiously yet decisively: Digital filtering in Saudi Arabia. In *USENIX FOCI*, 2020.

[4] S. Ali, M. Elgharabawy, Q. Duchaussoy, M. Mannan, and A. Youssef. Betrayed by the guardian: Security and privacy risks of parental control solutions. In *ACSAC*, 2020.

[5] R. Amos, G. Acar, E. Lucherini, M. Kshirsagar, A. Narayanan, and J. Mayer. Privacy Policies over Time: Curation and Analysis of a Million-Document Dataset. https://arxiv.org/pdf/2008.09159.pdf, 2020.

[6] B. Andow, S. Y. Mahmud, W. Wang, J. Whitaker, W. Enck, B. Reaves, K. Singh, and T. Xie. PolicyLint: Investigating internal privacy policy contradictions on Google Play. In *USENIX Security*, 2019.

[7] B. Andow, S. Y. Mahmud, J. Whitaker, W. Enck, B. Reaves, K. Singh, and S. Egelman. Actions speak louder than words: Entity-sensitive privacy policy and data flow analysis with PoliCheck. In *USENIX Security*, 2020.

[8] Android. Manifest.permission. https://developer.android.com/reference/android/Manifest.permission, 2021.

[9] New safety section in Google Play will give transparency into how apps use data. https://android-developers.googleblog.com/2021/05/new-safety-section-in-google-play-will.html, 2021.

[10] Android Open Source Project. Filters on Google Play. https://developer.android.com/google/play/filters, 2019.

[11] Android Open Source Project. App Manifest Overview. https://developer.android.com/guide/topics/manifest/manifest-intro, 2020.

[12] Android Open Source Project. Application signing. https://source.android.com/security/apksigning, 2020.

[13] Android Open Source Project. Command line tools. https://developer.android.com/studio/command-line, 2020.

[14] Android Open Source Project. Google Play. https://developer.android.com/distribute, 2020.

[15] Android Open Source Project. Network security configuration. https://developer.android.com/training/articles/security-config, 2020.

[16] Android Open Source Project. Sell digital purchases with Play In-app Billing. https://developer.android.com/distribute/best-practices/earn/in-app-purchases, 2020.

[17] Android Open Source Project. Version your app. https://developer.android.com/studio/publish/versioning, 2020.

[18] Android Open Source Project. android:usescleartexttraffic. https://developer.android.com/guide/topics/manifest/application-element, 2021.

[19] Google removes Fortnite from the Play Store for violating in-app payment policy. https://9to5google.com/2020/08/13/google-play-removes-fortnite/, 2020.

[20] Apktool. https://ibotpeaches.github.io/Apktool/.

[21] Android ad network statistics and market share. https://www.appbrain.com/stats/libraries/ad-networks.

[22] Apple Censorship. Apple Censorship. https://applecensorship.com/?l=en, 2020.

[23] ARTICLE 19. The Global Expression Report 2019 /2020. https://www.article19.org/wp-content/uploads/2020/10/GxR2019-20report.pdf.

[24] S. Aryan, H. Aryan, and J. A. Halderman. Internet censorship in Iran: A first look. In *USENIX FOCI*, 2013.

[25] E. Athanasopoulos, S. Ioannidis, and A. Sfakianakis. CensMon: A web censorship monitor. In *USENIX FOCI*, 2011.

[26] Austrlian House Standing Committee on Infrastructure and Communications. At what cost? IT pricing and the Australia tax, July 2013. https://www.aph.gov.au/Parliamentary_Business/Committees/House_of_Representatives_Committees?url=ic/itpricing/report.htm.

[27] M. Backes, S. Bugiel, and E. Derr. Reliable third-party library detection in Android and its security applications. In *ACM CCS*, 2016.

[28] A. D. Blog. Turning it up to 11: Android 11 for developers. https://android-developers.googleblog.com/2020/09/android11-final-release.html, 2020.

[29] App stores list. https://www.businessofapps.com/guide/app-stores-list/#1, 2021.

[30] M. Degeling, C. Utz, C. Lentzsch, H. Hosseini, F. Schaub, , and T. Holz. We Value Your Privacy ... Now Take Some Cookies: Measuring the GDPR's Impact on Web Privacy. In *NDSS*, 2019.

[31] DLA Piper. Data Protection Laws of the World. https://www.dlapiperdataprotection.com/system/

modules/za.co.heliosdesign.dla.lotw.data_protection/functions/handbook.pdf?country=all, 2020.

[32] Best alternative Android markets: review by country. https://thinkmobiles.com/blog/alternative-app-stores-for-android/, 2021.

[33] What are the biggest tracker networks and what can I do about them? https://spreadprivacy.com/biggest-tracker-networks/, 2021.

[34] European Commission. A Digital Single Market for the benefit of all Europeans. https://ec.europa.eu/digital-single-market/en/policies/shaping-digital-single-market, 2019.

[35] Exodus Privacy. https://exodus-privacy.eu.org, 2021.

[36] Financial Tribune. Sanctions make Iran developers witty. https://financialtribune.com/articles/sci-tech/72057/sanctions-make-iran-developers-witty, 2017.

[37] Fortune. Russia Demands LinkedIn App Takedown, Apple and Google Comply. https://fortune.com/2017/01/08/russia-linkedin-google-apple/, 2018.

[38] Freedom House. Freedom on the Net 2019: Iran. https://freedomhouse.org/country/iran/freedom-net/2019, 2019.

[39] Freedom House. Freedom on the Net Report 2019. https://freedomhouse.org/report/freedom-net/2019/crisis-social-media, 2019.

[40] Freedom on the Net Research Methodology. https://freedomhouse.org/reports/freedom-net/freedom-net-research-methodology, 2020.

[41] A. Garcia Asuero, A. Sayago, and G. González. The correlation coefficient: An overview. *Critical Reviews in Analytical Chemistry*, 2006.

[42] J. Geddes, M. Schuchard, and N. Hopper. Cover Your ACKs: Pitfalls of Covert Channel Censorship Circumvention. In *ACM CCS*, 2013.

[43] Google. Distribute app releases to specific countries. https://support.google.com/googleplay/android-developer/answer/7550024?hl=en, 2020.

[44] Google. Google Transparency Report. https://transparencyreport.google.com/government-removals/overview?hl=en, 2020.

[45] Google. Prepare your app for review. https://support.google.com/googleplay/android-developer/answer/9815348, 2020.

[46] Google. Requirements for distributing apps in specific countries/regions. https://support.google.com/googleplay/android-developer/answer/6223646?hl=en, 2020.

[47] Google. Select a category and tags for your app or game. https://support.google.com/googleplay/android-developer/answer/113475, 2020.

[48] Google. Supported locations for distribution to Google Play users. https://support.google.com/googleplay/android-developer/table/3541286, 2020.

[49] Google. Try new Android apps before they're officially released. https://support.google.com/googleplay/answer/7003180?hl=en, 2020.

[50] Google. User data. https://support.google.com/googleplay/android-developer/answer/9888076, 2020.

[51] Google. View & restrict your app's compatible devices. https://support.google.com/googleplay/android-developer/answer/7353455?hl=en, 2021.

[52] Google Play Help. https://support.google.com/googleplay/thread/22123351?hl=en, 2019.

[53] Google Transparency Report. https://transparencyreport.google.com, 2021.

[54] Google-Play-Scraper. https://github.com/JoMingyu/google-play-scraper.

[55] C. Han, I. Reyes, A. Feal, J. Reardon, P. Wijesekera, N. Vallina-Rodriguez, A. Elazari, K. A. Bamberger, and S. Egelman. The Price is (Not) Right: Comparing Privacy in Free and Paid Apps. In *PETs*, 2020.

[56] H. Harkous, K. Fawaz, R. Lebret, F. Schaub, K. G. Shin, and K. Aberer. Polisis: Automated Analysis and Presentation of Privacy Policies Using Deep Learning. In *USENIX Security*, 2018.

[57] Y. He, X. Yang, B. Hu, and W. Wang. Dynamic privacy leakage analysis of Android third-party libraries. *Journal of Information Security and Applications*, 2019.

[58] L. Hestres. App neutrality: Apple's app store and freedom of expression online. *Journal of Communication*, 2013.

[59] HideMyAss! HideMyAss! is pulling out of Russia. https://blog.hidemyass.com/en/hidemyass-is-pulling-out-of-russia, 2019.

[60] M. Hilbert. The bad news is that the digital access divide is here to stay: Domestically installed bandwidths among 172 countries for 1986–2014. *Telecommunications Policy*, 2016.

[61] Hulu. Why can't I use Hulu internationally? https://help.hulu.com/s/article/cant-use-internationally, 2020.

[62] Internet Fragmentation. https://icannwiki.org/Internet_Fragmentation, 2016.

[63] Information Commissioner's Office. Guide to the GDPR. https://ico.org.uk/for-organisations/guide-to-data-protection/guide-to-the-general-data-protection-regulation-gdpr/, 2020.

[64] Internet Society. Brief History of Internet. https://www.internetsociety.org/wp-content/uploads/2017/09/ISOC-History-of-the-Internet_1997.pdf, 2017.

[65] S. Jansen and E. Bloemendal. Defining app stores: The role of curated marketplaces in software ecosystems. In *Software Business*. Springer, 2013.

[66] B. Jones, T.-W. Lee, N. Feamster, and P. Gill. Automated Detection and Fingerprinting of Censorship Block Pages. In *IMC*, 2014.

[67] J. H. W. Jr. Hierarchical Grouping to Optimize an Objective Function. *Journal of the American Statistical Association*, 1963.

[68] keytool. https://docs.oracle.com/javase/8/docs/technotes/tools/unix/keytool.html.

[69] S. L. Lim, P. J. Bentley, N. Kanakam, F. Ishikawa, and S. Honiden. Investigating country differences in mobile app user behavior and challenges for software engineering. *IEEE Trans. on Software Engineering*, 2014.

[70] F. Lin, H. Wang, L. Wang, and X. Liu. A Longitudinal Study of Removed Apps in IOS App Store. In *WWW*, 2021.

[71] T. Linden, R. Khandelwal, H. Harkous, and K. Fawaz. The Privacy Policy Landscape After the GDPR. In *PETs*, 2020.

[72] B. Lindsay and M. T. Poindexter. The Internet: Creating Equity through Continuous Education or Perpetuating a Digital Divide? *Journal of Comparative Education Review*, 2003.

[73] X. Liu, J. Liu, S. Zhu, W. Wang, and X. Zhang. Privacy Risk Analysis and Mitigation of Analytics Libraries in the Android Ecosystem. *IEEE Trans. on Mobile Computing*, 2020.

[74] Luminati. https://luminati.io.

[75] Z. Ma, H. Wang, Y. Guo, and X. Chen. LibRadar: Fast and Accurate Detection of Third-Party Libraries in Android Apps. In *IEEE/ACM Software Engineering Companion*, 2016.

[76] D. Mac Sıthigh. App law within: rights and regulation in the smartphone age. *Journal of Law and Information Technology*, 2013.

[77] A. J. Martin. Google faces calls to lift anti-censorship blocks in Iran. https://news.sky.com/story/google-faces-calls-to-lift-anti-censorship-blocks-in-iran-11192888, 2018.

[78] W. Martin, F. Sarro, and M. Harman. Causal Impact Analysis for App Releases in Google Play. In *ACM SIGSOFT*, 2016.

[79] W. Martin, F. Sarro, Y. Jia, Y. Zhang, and M. Harman. A Survey of App Store Analysis for Software Engineering. *IEEE Trans. on Software Engineering*, 2017.

[80] A. McDonald, M. Bernhard, L. Valenta, B. VanderSloot, W. Scott, N. Sullivan, J. A. Halderman, and R. Ensafi. 403 Forbidden: A Global View of CDN Geoblocking. In *IMC*, 2018.

[81] Top 10 Android App Analytics Platforms. https://medium.com/android-news/top-10-android-app-analytics-platforms-b3496c55b6f1, 2017.

[82] Microsoft. Skype Lite. https://www.skype.com/en/skype-lite/, 2020.

[83] multipass. https://github.com/canonical/multipass.

[84] P. Mutchler, Y. Safaei, A. Doupé, and J. Mitchell. Target Fragmentation in Android Apps. In *IEEE S&P*, 2016.

[85] W. Navidi. *Statistics for Engineers and Scientists*, chapter 4. McGraw-Hill Education, 2014.

[86] NIST Policy on Hash Functions. https://csrc.nist.gov/Projects/Hash-Functions/NIST-Policy-on-Hash-Functions, 2015.

[87] NIST Special Publication. Transitioning the Use of Cryptographic Algorithms and Key Lengths. https://csrc.nist.gov/CSRC/media/Publications/sp/800-131a/rev-2/draft/documents/sp800-131Ar2-draft.pdf, 2019.

[88] E. Novak and C. Marchini. Android App Update Timing: A Measurement Study. In *IEEE Mobile Data Management*, 2019.

[89] 10 Mobile Usage Statistics Every Marketer Should Know in 2020. https://www.oberlo.com/blog/mobile-usage-statistics, 2020.

[90] Open Observatory of Network Interference. https://ooni.org.

[91] Open Observatory of Network Interference. OONI API. https://api.ooni.io.

[92] P. Pearce, B. Jones, F. Li, R. Ensafi, N. Feamster, N. Weaver, and V. Paxson. Global measurement of DNS censorship. In *USENIX Security*, 2017.

[93] E. Peltonen, E. Lagerspetz, J. Hamberg, A. Mehrotra, M. Musolesi, P. Nurmi, and S. Tarkoma. The hidden image of mobile apps: Geographic, demographic, and cultural factors in mobile usage. In *Human-Computer Interaction with Mobile Devices and Services*, 2018.

[94] PlaystoreDownloader. https://github.com/ClaudiuGeorgiu/PlaystoreDownloader.

[95] R. Potharaju, M. Rahman, and B. Carbunar. A Longitudinal Study of Google Play. *IEEE Trans. on Computational Social Systems*, 2017.

[96] Proxychains. https://github.com/rofl0r/proxychains-ng.

[97] Qualys. SSL Server Test. https://ssllabs.com/ssltest, 2020.

[98] M. Ragnedda, M. L. Ruiu, and F. Addeo. Measuring Digital Capital: An empirical investigation. *New Media & Society*, 2020.

[99] R. Raman, A. Stoll, J. Dalek, R. Ramesh, W. Scott, and R. Ensafi. Measuring the Deployment of Network Censorship Filters at Global Scale. In *NDSS*, 2020.

[100] R. Ramesh, R. S. Raman, M. Bernhard, V. Ongkowijaya, L. Evdokimov, A. Edmundson, S. Sprecher, M. Ikram, and R. Ensafi. Decentralized Control: A Case Study of Russia. In *NDSS*, 2020.

[101] Readabilipy. https://github.com/alan-turing-institute/ReadabiliPy.

[102] L. Richardson. Beautiful Soup. https://www.crummy.com/software/BeautifulSoup/, 2020.

[103] RISKIQ. 2020 Mobile App Threat Landscape Report. https://www.riskiq.com/wp-content/uploads/2021/01/RiskIQ-2020-Mobile-App-Threat-Landscape-Report.pdf, 2020.

[104] Selenium. https://github.com/SeleniumHQ/selenium/.

[105] Y. Shen, P. Vervier, and G. Stringhini. Understanding Worldwide Private Information Collection on Android. In *NDSS*, 2021.

[106] Sophos Naked Security. Analytics firm's VPN and ad-blocking apps are secretly grabbing user data. https://nakedsecurity.sophos.com/2020/03/12/analytics-firms-vpn-and-ad-blocking-apps-are-secretly-grabbing-user-data/, 2020.

[107] "The item you were attempting to purchase could not be found" Android in-app billing. https://stackoverflow.com/questions/23918190/the-item-you-were-attempting-to-purchase-could-not-be-found-android-in-app-bil.

[108] State of California Department of Justice. California Consumer Privacy Act (CCPA). https://www.oag.ca.gov/privacy/ccpa, 2020.

[109] N. D. Steiner. Comparing Freedom House Democracy Scores to Alternative Indices and Testing for Political Bias: Are US Allies Rated as More Democratic by Freedom House? *Journal of Comparative Policy Analysis: Research and Practice*, 2016.

[110] R. Sun and M. Xue. Quality Assessment of Online Automated Privacy Policy Generators: An Empirical Study. In *Evaluation and Assessment in Software Engineering*, 2020.

[111] R. Sundara Raman, P. Shenoy, K. Kohls, and R. Ensafi. Censored Planet: An Internet-wide, Longitudinal Censorship Observatory. In *ACM CCS*, 2020.

[112] A. Swanson, D. McCabe, and J. Nicas. Trump Administration to Ban TikTok and WeChat From U.S. App Stores. https://www.nytimes.com/2020/09/18/business/trump-tik-tok-wechat-ban.html, 2020.

[113] V. F. Taylor and I. Martinovic. To Update or Not to Update: Insights From a Two-Year Study of Android App Evolution. In *Asia CCS*, 2017.

[114] The Hindu. Government bans 59 apps including China-based TikTok, WeChat. https://www.thehindu.com/news/national/govt-bans-59-apps-including-tiktok-wechat/article31947445.ece, 2020.

[115] The Telegraph. Facebook faces legal hurdle to launch Messenger Kids app in UK. https://www.telegraph.co.uk/technology/2020/05/02/facebook-faces-legal-hurdle-launch-messenger-kids-app-uk/, 2020.

[116] Think Progress. Iranians outraged over removal of apps from Google Play store. https://archive.thinkprogress.org/google-play-iranian-apps-sanctions-2e7ba4b1649d/, 2017.

[117] TorGuard. Has Hong Kong started blocking VPN providers? https://torguard.net/blog/has-hong-kong-started-blocking-vpn-providers/, 2020.

[118] M. C. Tschantz, S. Afroz, S. Sajid, S. A. Qazi, M. Javed, and V. Paxson. A Bestiary of Blocking: The Motivations and Modes behind Website Unavailability. In *USENIX FOCI*, 2018.

[119] V. Ververis, M. Isaakidis, V. Weber, and B. Fabian. Shedding Light on Mobile App Store Censorship. In *User Modeling, Adaptation and Personalization*, 2019.

[120] H. Wang, H. Li, and Y. Guo. Understanding the evolution of mobile app ecosystems: A longitudinal measurement study of Google Play. In *WWW*, 2019.

[121] H. Wang, H. Li, L. Li, Y. Guo, and G. Xu. Why are Android apps removed from Google Play? A large-scale empirical study. In *MSR*, 2018.

[122] H. Wang, H. Liu, X. Xiao, G. Meng, and Y. Guo. Characterizing Android App Signing Issues. In *IEEE/ACM Automated Software Engineering*, 2019.

[123] H. Wang, Z. Liu, J. Liang, N. Vallina-Rodriguez, Y. Guo, L. Li, J. Tapiador, J. Cao, and G. Xu. Beyond Google Play: A large-scale comparative study of Chinese Android app markets. In *IMC*, 2018.

[124] B. Warf. Geographies of global Internet censorship. *GeoJournal*, 2011.

[125] K. Yoshida, H. Imai, N. Serizawa, T. Mori, and A. Kanaoka. Understanding the origins of weak cryptographic algorithms used for signing Android apps. In *IEEE Computer Software and Applications*, 2018.

[126] L. Yu, X. Luo, J. Chen, H. Zhou, T. Zhang, H. Chang, and H. Leung. PPChecker: Towards Accessing the Trustworthiness of Android Apps' Privacy Policies. *IEEE Trans. on Software Engineering*, 2018.

[127] N. Zhong and F. Michahelles. Google Play is not a long tail market: an empirical analysis of app adoption on the Google Play app market. In *ACM Applied Computing*, 2013.

[128] J. Zittrain and B. Edelman. Internet filtering in China. *IEEE Internet Computing*, 2003.

A Regional similarity

A world map of countries and their clusters based on app availability is shown in Figure 9. We found a high similarity in available apps between countries in the same region.

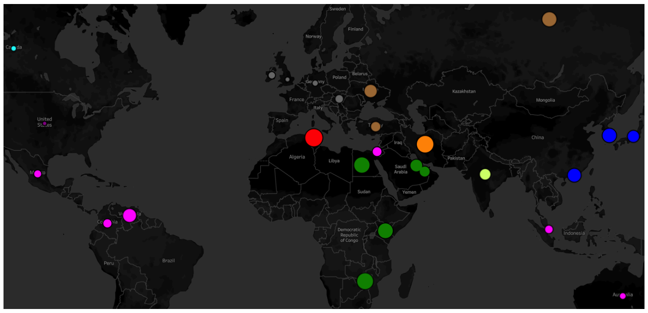


Figure 9: **World map of clusters**: Countries in the same geographic region have comparable app availability.

B SSL Error in Tunisia

In Tunisia, 1,819 apps consistently failed to download with SSL bad handshake error (Err7), even on repeated attempts. We confirmed that these errors are not because of client and server misconfigurations. Our packet capture of the client's communication with Google servers in Tunisia and other countries show that the SSL/TLS handshake completes successfully, even when the error from Tunisia suggests otherwise. We also ran the Qualys SSL Server Test [97] on Tunisia's endpoint and found no anomalies, suggesting that the SSL errors occur during the data transfer associated with the `apk` download. We found that the client (Linux host) received duplicate or replayed ACK packets during an app download, causing the failure.

To further confirm a download error, we also manually attempted to install several apps via a VPN in Tunisia on our phone. For apps that failed, the install remained in a wait state indefinitely. This may suggest a network interference practice to prevent the apps from being installed. Prior research has shown that censors may do deep packet inspection on SSL traffic and interfere with packets containing ACKs, including replaying them, to slow down or halt the data transfer [42].

C GDP vs. FHIF

In our preliminary study, we examine whether there exists a correlation between a country's GDP and its FHIF score using Spearman's rank correlation coefficient metric. We find that there is a low, positive correlation (0.42) between Gross Domestic Product (GDP) and Freedom House's Internet Freedom (FHIF) Score, as shown in Figure 10. However, we observe countries like IN, which has relatively high GDP and lower FHIF score, and HU, which has lower GDP and a free FHIF score. We also observe countries with comparable GDP and different FHIF classifications (e.g. free CA and not free RU).

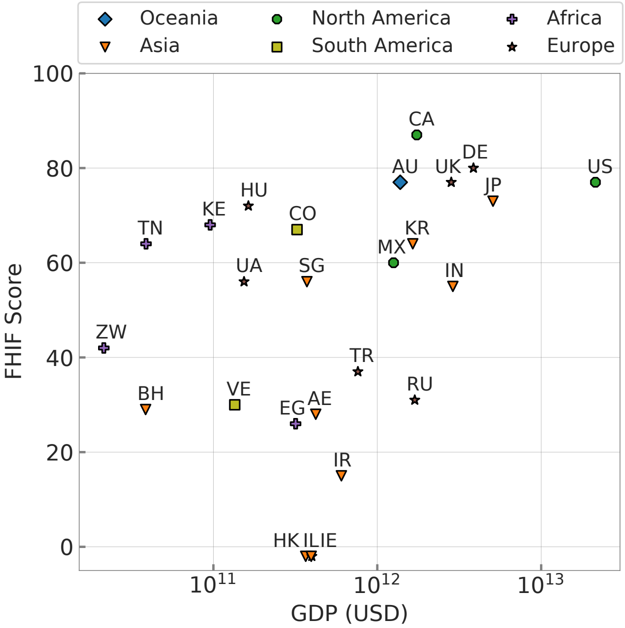
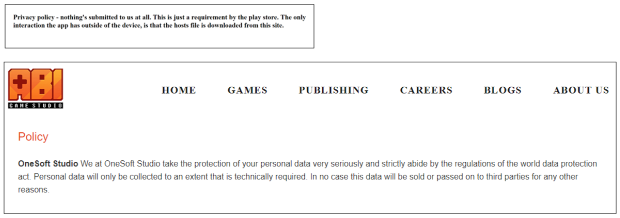


Figure 10: **GDP vs. FHIF Score**: The x axis is log-scaled. While there is a low, positive correlation between GDP and FHIF score, there are countries with comparable GDP with varying FHIF classifications (e.g. CA and RU). Note that HK, IL, and IE do not have a FHIF score.

D Privacy Violations

Outside of geographic differences, we also observe policies that disregard user privacy or make false claims (Figure 11).



Privacy policy - nothing's submitted to us at all. This is just a requirement by the play store. The only interaction the app has outside of the device, is that the hosts file is downloaded from this site.

Policy

OneSoft Studio We at OneSoft Studio take the protection of your personal data very seriously and strictly abide by the regulations of the world data protection act. Personal data will only be collected to an extent that is technically required. In no case this data will be sold or passed on to third parties for any other reasons.

Figure 11: **Privacy Violations**: The policy of `com.segfaultstudios.suckmyads` (top) disregards user privacy, while the policy of `com.alien.shooter.galaxy.attack` (bottom) follows the "world data protection act".