Out in the Open: On the Implementation of Mobile App Filtering in India

Devashish Gosain[1,2], Kartikey Singh[3], Rishi Sharma[3], Jithin S[3], and Sambuddho Chakravaty[3]

[1] BITS Pilani Goa, India
[2] MPI-INF, Germany
[3] IIIT Delhi, India
devashishg@goa.bits-pilani.ac.in
{kartikey17242,rishi17260,jithins,sambuddho}@iiitd.ac.in

Abstract. In this paper, we present the first comprehensive study highlighting the evolving mobile app filtering within India. We study the recent mobile app blocking in India and describe in detail the mechanics involved. We analyzed 220 Chinese apps that were blocked due to official government orders. Our research reveals a novel "three-tiered" app filtering scheme, with each tier increasing the sophistication of filtering. After thoroughly analyzing the app blocking mechanisms, we present circumvention techniques to bypass the tiered app filtering. We were able to access all the blocked apps with the said techniques. We believe our analysis and findings from the case study of India will aid future research on mobile app filtering.

1 Introduction

There are numerous instances of wide-scale filtering in different parts of the globe [63]. While there exist a plethora of studies that systematically analyzed Internet filtering [61,55], a significant fraction of them focused on "Great Firewall of China" and reported its multi-faceted censorship [68,42,29,25,34,38,51,67]. In this paper, we take a different direction and focus on a less studied country India [41,72,60]. Previous censorship studies in India reported different categories of blocked websites [41], the techniques used [60], and a detailed analysis of mechanics employed by Indian ISPs to achieve *web* censorship [72]. However, for the first time, we study a new filtering ecosystem *i.e., mobile app blocking* within India.

Due to the rapid growth of mobile Internet applications, app blocking will become an essential component of any nationwide censorship system. Already, there is anecdotal evidence of banning mobile apps by different countries [27,3,56]. Thus, we conduct a comprehensive study to analyze the mechanics of this less-studied form of filtering. We consider India as a case study where the banning of mobile apps is a recent phenomenon, with 59 popular Chinese apps [15,13] being initially blocked on the orders of the Indian government in June

2020. Since then, India has continued its app-blocking spree, and presently 220 Chinese apps are blocked.

We begin by analyzing the blocking of a popular app *TikTok* (which has over 2.6 billion downloads worldwide [16]). We expected to see traditional filtering techniques being used by the Indian ISPs, *viz.* DNS filtering, TCP/IP blocking, and keyword filtering [58,67]. But, to our surprise, we observed no filtering from the Indian ISPs. Instead, a successful TLS connection was established between the app and the actual TikTok server (confirmed via TikTok's legitimate certificate). Moreover, we also noticed that the censorship notification banner was a part of the same TLS session, proving that the TikTok server sent it. *This confirmed that TikTok app blocking is not carried out by the Indian ISPs; rather, it was by the TikTok server itself.*

Our observations were similar for all other filtered apps as well *i.e.,* Indian ISPs were not at all involved in the filtering. Instead, the app servers were themselves selectively filtering the Indian users following the Indian government's blocking orders [5,6,7]. To identify the technique(s) for filtering, we analyzed traffic footprints (through `pcaps`) and even reverse-engineered a few apps. Our observations were surprising—some apps were even *probing the SIM card* for country information to restrict the user access (see §4 for details).

The app blocking kept evolving over the course of two years, with some apps changing their blocking mechanisms from just probing the SIM to a combination of inspecting the SIM and source IP addresses (*e.g.,* TikTok). We studied this evolving behavior, and based on our findings, we divided the app filtering mechanics into three tiers, with each tier adding a level of sophistication in blocking criteria.

1. *Tier three* apps are those that are unavailable in the Indian app stores. We found 160 such apps[4]. 136 out of these can be accessed directly after installation. We obtained the *apk* of these apps from third-party sources like `apkmirror.com`.
2. *Tier two* apps are those that, after installation, would still suffer blocking. These app publishers selectively filter Indian clients using geo-blocking [53]. 23 (out of the previous 160) apps fall under this category.
3. *Tier one* apps are those that employ the most sophisticated technique. As already explained with the example of TikTok, not only do these apps censor users by identifying locale information from the SIM cards installed, but they also use geo-blocking. In total, 7 (out of the previous 23) apps employ such filtering techniques. Note that *MICO Chat*, is the only exception that has exclusively adopted SIM-based blocking.

While analyzing the geoblocking of apps, we observed different geoblocking techniques employed by these apps. While most apps employed source IP blocking for geoblocking, there was one app of particular interest (ChessRush)

[4] Out of 220 blocked apps, 60 apps were defunct. We couldn't find them in official play stores of foreign countries as well.

that restricted the content on its CDN edge servers. We provide details of our investigation in uncovering such mechanisms in §4.3.

In general, our findings on mobile apps shed light on some grave concerns. Following this model, in the future, many app publishers may adopt similar censorship techniques if coerced by authoritarian regimes. Alarmingly, *all* apps could censor users, even without communicating with app servers as the logic to extract identifying users' locale (*e.g.,* from SIM cards) can be embedded within the apps' binary[5].

2 Background and Related Work

In this paper, we focus on India, which seems ambivalent about its censorship policies [41]. As already mentioned, two different forms of filtering mechanisms have evolved in the country—*viz.* website blocking and mobile app filtering.

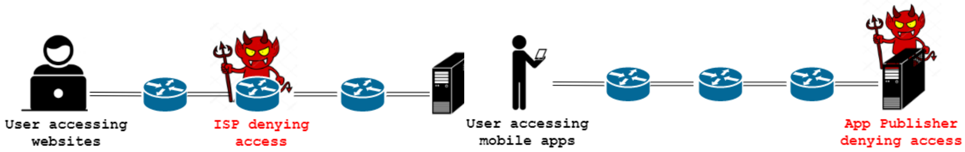
Web censorship: There exist a plethora of studies that reported Internet censorship across the globe [61,55]. Researchers have reported censorship in various countries like Syria [33], Iran [26], Greece [66], Italy [22], Pakistan [54,30], Saudi Arabia [23], Russia [58,71,11,18], Spain [64] *etc.* Notably, several studies particularly focus on analyzing the Great Firewall of China [70,52,39,67,46,44,48,25,38,34,8,45,24], owing to its technical sophistication.

In this paper, we focus on India—a country with more than a billion Internet users [2]. In the year 2017, Gosain *et al.* [41] conducted the first (preliminary) study and reported that Indian ISPs follow a "federated model of censorship" *i.e.,* they have inconsistent censorship policies that result in huge differences in the censorship experienced by the netizens. Later, in 2018, Yadav *et al.* [72] conducted a detailed study of web censorship in India and reported its multi-faceted aspects. They confirmed the previous observations [41] and reported the presence of various censorship middleboxes positioned in India. They further demonstrated that using specially crafted web requests, similar to those reported in [43,49], it is possible to bypass such middleboxes. However, Yadav *et al.* did not report filtering of HTTPS websites. But recently, in 2020, Singh *et al.* [60] reported that one Indian ISP has started blocking HTTPS websites using TLS SNI extension.

Moreover, other groups like Citizen Lab [20] regularly assesses Internet filtering in different regions across the globe [10], [8], [21], study the deployed filtering and surveillance infrastructure [12], [17] and report privacy violations [36]. There exist other large-scale measurement projects like ICLab [55], OONI [40], and CensoredPlanet [61] that track and report censorship events across the globe, including India. These measurement projects are extensive in scale and report a breadth of important information about web censorship but do not study mobile app blocking.[6] Thus, in this paper, we devised our own approach and heuristics to analyze the prevailing mobile app filtering within India.

[5] With regular updates in the app, the publishers can easily introduce such changes.

[6] OONI, in addition to web filtering, also test the blocking of four instant messaging apps—WhatsApp, Facebook Messenger, Telegram, Signal [19].



(a) Web filtering

(b) App filtering

Fig. 1: Types of Internet filtering in India: (a) Web filtering is achieved by an ISP (b) App filtering is achieved by the app publisher themselves.

App filtering: In June 2020, for the *first* time, the Government of India officially banned 59 Chinese apps in response to growing tensions on the Indo-China border [50]. However, the official blocking order claimed that these apps pose a threat to the privacy and data security of Indian users [13]. By November 2020, the number of banned apps increased to 220 [1]. Banning these apps involved removing them from all the official app stores (*e.g.,* Google and Apple). Moreover, following the ban, even the pre-installed apps stopped working. As described ahead, several of these app publishers go to great lengths to identify Indian users so as to not serve content to them. This departs from the traditional model of censorship, where ISPs attempt to block content rather than the website maintainer itself. Further, we also confirm that the techniques used for censoring apps are quite different from those used in web censorship.

Previous studies also confirmed the unavailability of apps in the app stores in different countries. Ververis *et al.* [65] reported that many apps (*e.g.,* VPNs) are not available in the app stores of multiple countries (*e.g.,* China, Syria). However, the authors acknowledged that it is difficult to ascertain the precise cause of unavailability, *i.e.,* it is due to commercial reasons or because of the orders from the government. Similarly, Kumar *at al.* [47] conducted a measurement study to analyze the geo-differences in the mobile apps from 26 different countries. They report that 3,672 apps were geo-blocked in at least one of the said countries. Apps are unavailable in the app stores for various reasons, *e.g.,* takedown by the government, removal by Google due to noncompliance with its policy, and blocking by the developer due to commercial reasons.

In this work, we take a slightly different direction and focus solely on India. Our goal was to identify the *app filtering mechanics* for those apps that are known to be banned by the orders of the government. Our research reveals multiple app filtering techniques at play—almost all banned apps are not available in app stores, many apps are blocked based on the source IP addresses of the client, some apps are not available on select CDN edge servers, and some apps are fetching the locale of the client from the SIM card to restrict Indian users from accessing the content.

Our research shows that it is important to study app filtering and identify possible circumvention solutions. In the future, other censoring countries could adopt similar filtering techniques to censor mobile apps apart from traditional website filtering.

3 Ethical Considerations

Censorship measurement studies often require accessing blocked websites (and mobile apps) that are deemed objectionable by different governments. Thus, accessing the blocked apps may evoke suspicion of the authorities against the individuals involved. Thus, for this study, we carefully devised our experiments following the recommendations given by Belmont [28] and Menlo [37] reports. We applied for the university's IRB approval, and we obtained the same.

We obtained access to Indian ISPs by purchasing their SIM cards. We were extremely careful at this step; only the author(s) of this study (who are citizens of India) purchased the SIM cards. No third person (Indian or foreign) was involved in this. Later, all our mobile Internet connectivity experiments were performed using these SIM cards only. Moreover, throughout the study, we accessed mobile apps from our own infrastructure (mobile phones and servers).

In some experiments, we required sending DNS requests (containing filtered domains) to DNS resolvers. Thus, we required scanning various ISP prefixes for open DNS resolvers. Sending queries for filtered domains to non-ISP resolvers may further force such resolvers to communicate with the top-level DNS infrastructure, thereby putting them under the suspicion of the authorities. Thus, we only selected those that belong to ISPs' infrastructure and avoided non-ISP resolvers.

To do so, from the list of all available open DNS resolvers in the ISP under test, we first performed the reverse DNS PTR lookup for them. We selected those that likely belonged to ISPs' infrastructure. For instance in Airtel, we selected the ones that had a substring `airtelbroadband.in` in the reverse PTR. Similarly in ACT ISP, we selected those resolvers that had `broadband.actcorp.in` as a substring. Additionally, in both the ISPs, we also selected those that were likely authoritative nameservers for some domain. As suggested in [58], we selected only those resolvers whose PTR began with the regular expression "ns[0-9]+nameserver[0-9]".

4 App Censorship Investigation

For the first time, India officially banned 220 apps by the end of November 2020. Our objective was to study how exactly these apps were blocked in the country. More precisely, we attempted to answer the following questions.
(1) Are the apps available in app stores? If not, how can we obtain these apps?
(2) Are these apps accessible/usable once we install them from alternate sources?
(3) After installation, if the apps are not usable, what are the mechanisms used to censor them?

4.1 Experimental Setup

We installed apps on our own mobile phones (running the latest Android version 10 and the iPhone iOS version 14). Likewise, our overseas contact in Germany (a paper-author) used their mobile phones to install such apps and performed tests. For analyzing the app traffic, we required installing `mitmproxy` [9], which helps intercept, decrypt, and analyze TLS traffic. This required the installation of self-signed certificates within the mobile phones. However, the apps in Android versions newer than 7 cannot use certificates signed by untrusted issuers. This is called *certificate pinning* [59]. Thus, to bypass certificate pinning, only when analyzing app traffic, we used an older version of the Android (*i.e.,* 6). The `mitmproxy` was run on a Linux host running Ubuntu 20.04.1, equipped with quad-core x64 processor and 8 GB of RAM.

Accessing censored apps: The mobile apps were gradually banned in India in three stages, first in June, then in September and lastly in November 2020. We began our research by curating a list of the banned apps from the press releases of the Ministry of Electronics and IT, Government of India [5,6,7]. The banned apps belonged to different categories *viz.* gaming, social media, dating *etc.* (as reported by the app publishers). The gaming category alone constituted ≈ 22% of these apps (see Fig. 2).

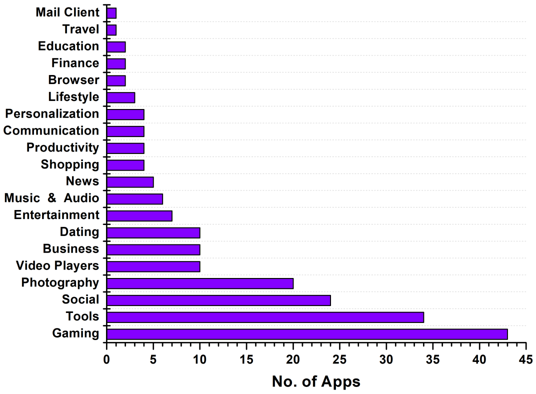


Fig. 2: Different categories of mobile apps filtered in India.

We began by searching for these apps on popular app stores (*e.g.,* Google and Apple). As expected, these apps were removed from them. Thus, to obtain the installation packages of the blocked apps (*e.g.,* `apk` files), we relied on our overseas contacts in *uncensored* countries where these apps were likely available. Interestingly, 60 among these 220 apps were unavailable even in their app markets, indicating that they were likely not operational. Additionally, we verified that the installation packages of these apps were unavailable even on third-party sources *e.g.,* `apkmirror.com` (ref. §A.3 for details). Finally, our overseas contacts downloaded the packages for the remaining 160 apps from their official stores and shared them with us. After installing these apps on our mobile phones in India, we manually tried accessing them. We found that 136 out of 160 apps (unavailable in the Indian app stores) could be directly accessed. After that, we investigated the censorship for the rest of the 24 apps.

Initial observations: We commenced our research by analyzing a few popular blocked apps like TikTok, PUBG and Vmate *etc.* Upon accessing TikTok, we observed a censorship notification issued in compliance with the government's orders and could not load the content. Next, when we attempted to open the PUBG game, it showed no explicit censorship message; instead, it simply reported a connection error and prevented us from launching the game. It was a "false notification" in the app; through `pcap` file, we confirmed that, indeed, there was a TCP connection followed by a successful TLS handshake between the app and the server.

In addition, several other apps like Vmate and UVideos neither showed any censorship notification nor any connection error. Instead, they showed no contents whatsoever. This indicated that different apps were likely being censored in different ways. These apps rely on standard web protocols (*i.e.,* HTTP and HTTPS). It is possible that Indian ISPs might be filtering the traffic of these apps, much like traditional web censorship [72,60].

4.2 Investigating ISP level filtering

Indian ISPs generally rely on keyword filtering (*e.g.,* DNS, HTTP(S) headers) [60,72] to filter traffic. Thus we began by inspecting the network traffic of the TikTok app via `pcap` files. However, we observed only the standard network protocol messages:

1. The domain `tiktok.com` resolved to public IP addresses belonging to Akamai's own AS. We resolved the same domain from other VPSs in uncensored countries and found that the IP addresses obtained also belonged to Akamai. Thus, DNS censorship was likely not being used as all IP addresses belong to the same hosting service.
2. With the said IPs, we were able to successfully complete the TCP handshake. This largely ruled out TCP/IP filtering.[7]
3. Following that, we were also able to successfully complete the TLS handshake with the same IPs. We observed a legitimate certificate signed by `DigiCert Inc.` bearing the common name as `*.tiktokv.com`. This confirmed that HTTPS censorship (using SNI extension of TLS ClientHello) was also not employed.
4. Eventually, we observed that encrypted data was exchanged between the app and the TikTok server.

We analyzed the network traffic for all other blocked apps, and our observations were consistent—*the responses received by the apps were from the app server itself, and not manipulated by the ISP* (as it usually happens with web censorship). Ironically, this indicated that the app publishers themselves, not the ISPs, restricted the Indian app users from accessing the app content following the government's blocking orders. We repeated the experiments in four popular ISPs

[7] It may happen that we complete the TCP handshake with some transparent proxy (in the ISP). In such cases, one may not detect TCP/IP filtering. But, this would also result in failure of the next steps. However, in practice, not even for a single app, we encountered this case.

that cumulatively capture more than 95% of the Indian clients [2]—Reliance Jio, Airtel, Vodafone-Idea, and ACT network, and observed the same behavior, *i.e.,* no network level interference by the ISPs.

4.3 Investigating censorship mechanics used by app publishers

After ruling out the role of Indian ISPs in app censorship, our next concern was how app publishers were selectively filtering users from India and not elsewhere. We hypothesized that these blocked apps could be using a well-known technique, *IP geo-blocking* [53] for the same. IP geo-blocking involves web servers rejecting requests originating from specific regions, identified through their source IP address. This step often involves looking up geo-IP databases to map IP addresses to countries. However, an alternative could be to filter requests from specific regions. A large number of such app publishers might use CDNs, much like almost all popular web services [31]. In such a case, the app publisher could restrict content from being available to edge servers that serve requests from specific regions (*e.g.,* India). Thus, app publishers might geo-block clients broadly in two ways:[8]

1. *Geo-blocking based on source IP address:* The app publisher would restrict the users based on the IP address of the incoming requests.
2. *Geo-blocking by restricting content on CDN edge-servers:* The app publisher may have the control to restrict content on edge servers that serve requests from India.

For those apps that do not rely on CDNs, edge-server-based blocking could be directly ruled out. For others (that use CDNs), we would need to distinguish between the aforementioned two possibilities. Thus, we first identified whether the blocked apps were relying on CDNs or not.

Identifying the apps that use CDNs: Identifying the use of CDNs requires distinguishing between different types of CDNs. Broadly there are two types of CDNs *viz. DNS* based, and *anycast* CDNs [32]. In DNS-based CDNs [69] (*e.g.,* Akamai), DNS queries for web services are resolved often to the nearest edge-servers, generally identified from the clients' DNS resolvers' locations. However, in anycast-based CDNs [35] (*e.g.,* Cloudflare), edge-serves in different locations use the same IP address, which is announced through different BGP advertisements from different geographic locations. A client's web request is directed to the closest possible edge server based on the BGP policies of the client's ISP. We explain in Appendix A.1 how we identified the use of CDNs by these apps. Overall, we found that all the 24 app publishers hosted content on DNS-based CDNs. After confirming the role of CDNs with blocked apps, we revisit our problem of how app publishers censor Indian users and the role of CDNs (if any).

[8] There could be more ways to identify the client's locale *e.g.,* time-zone of the phone. We consider these possibilities where we ruled out the obvious next two.

Geo-blocking mechanisms employed: Our goal was to identify how app publishers are selectively filtering Indian users—on the basis of source IP or simply denying access to edge servers catering to Indian users. Accessing the apps via VPNs with end-points abroad could be an easy way to confirm if the apps are accessible or not. This may likely help circumvent IP geo-blocking. However, our aim was to first identify how exactly geo-blocking was implemented, but VPNs not only change the source IP addresses of apps' requests but also the edge servers to which they communicate (depending on the DNS resolvers the VPNs used). This makes it hard to discern how the requests are being filtered.

Thus, we devised heuristics involving changing a single factor at a time. Corresponding to these two factors, we examined four possible scenarios. For instance, one scenario is accessing apps by selecting foreign edge servers while still using an Indian IP address. Since the apps were using DNS-based CDNs, switching to DNS resolvers in uncensored countries could force the apps to communicate with foreign edge servers without changing the Indian source IP address. Alternately, another case would be to access apps with foreign source IP and connect to Indian edge servers. For this, our overseas contacts (whose phones bear foreign IPs) used Indian DNS resolvers to communicate with Indian edge servers. We now elucidate all four possible scenarios.

Sl No.	**App Name**	**App Type**	**Censorship Technique Used**		
			Client Source IP	**CDN Edge server**	**Client SIM Card**
1	PUBG	Gaming	✓	✗	✗
2	ShareIt	Tools	✓	✗	✗
3	Shein	Shopping	✓	✗	✗
4	Baidu	Tools	✓	✗	✗
5	Tantan	Social	✓	✗	✗
6	VooV	Productivity	✓	✗	✗
7	RomWe	Shopping	✓	✗	✗
8	Ludo	Gaming	✓	✗	✗
9	Rangers of Oblivion	Gaming	✓	✗	✗
10	Ali Suppliers	Business	✓	✗	✗
11	Baidu Express	Tools	✓	✗	✗
12	DingTalk	Productivity	✓	✗	✗
13	MangoTV	Video Players	✓	✗	✗
14	Heroes Evolved	Gaming	✓	✗	✗
15	Singol	Dating	✓	✗	✗
16	ChessRush	Gaming	✓	✓	✗
17	TikTok	Social	✗	✗	✓
18	Likee	Video Players	✗	✗	✓
19	Kwai	Social	✗	✗	✓
20	UC Browser	Browser	✗	✗	✓
21	FaceU	Photography	✗	✗	✓
22	Hago	Social	✗	✗	✓
23	V-Fly	Tools	✗	✗	✓
24	MICO Chat	Social	✗	✗	✓

Table 1: Filtering mechanisms employed by different blocked apps (before the permanent ban).

Case 1*: Indian Source IP and Indian edge-server:* The blocked apps were accessed directly from our Indian mobile phone, configured to use Indian resolvers. This is the typical situation where a regular user tries to use the app. Thus while 136 apps were trivially accessible after installation, the 24 apps we identified fell in this category and were inaccessible (ref. Table 1).

Case 2: *Indian Source IP and Foreign edge-server:* The apps were accessed from phones configured to use open resolver in non-censoring countries (Germany) that do not block apps based on government orders. This enabled the apps to connect to an edge server probably located in such countries. We verified this by inspecting the `traceroute` path from an Indian mobile phone to the resolved IPs of the edge servers. The last few IP addresses in the `traceroute` paths belonged to the same (uncensored) country as that of the DNS resolver.

Unfortunately, the 24 apps were still inaccessible, even when they communicated to foreign edge servers. While the DNS resolvers of the phones were changed, their IP addresses weren't. This indicated that the app publishers were filtering requests based on source IP through the edge servers, even when the latter were outside India.

Case 3: *Foreign Source IP and Indian edge-server:* To use a foreign source IP, we set up a VPS in an uncensored country and ran our own `OpenVPN` service on it. We configured our Indian mobile phone (with blocked apps installed) to use the said `OpenVPN` service. This ensured that even if we accessed apps from the mobile phone (in India), the requests would bear a foreign source IP address. Both the `OpenVPN` service and the VPS host were configured to use an open DNS resolver in India. This forced the apps to connect to edge-servers that cater to Indian users. We found that 15 among the 24 apps mentioned above were accessible; their traffic bore foreign source IPs and was destined to Indian edge servers. Thus based on the hitherto 3 cases, we observed that:

1. All the 24 apps were censored when their requests bore Indian source IPs and connected to Indian edge servers.
2. They were censored even when connected with foreign edge servers while using Indian source IPs.
3. However, 15 of them *were accessible* when connected to Indian edge servers but used foreign source IPs. *This confirmed IP geo-blocking for these apps.*

Case 4: *Foreign Source IP and Foreign edge-server:* Finally, we accessed the apps through a VPN with endpoints in foreign countries. This resulted in the requests bearing foreign source IPs and terminating at foreign edge servers. We expected all the 24 censored apps to be accessible. To our surprise, we were able to access only 16 out of 24 apps! These 16 apps included the previously accessible 15 apps.

Interestingly, the additional app *Chess Rush,* was both IP geo-blocked as well as unavailable at the Indian edge server. It was only accessible when using both foreign source IP addresses and foreign edge servers. These 16 apps with their censorship mechanisms are listed in Table 1 (rows 1–16). To understand the censorship mechanics of the remaining 8 apps, we ran additional experiments.

Investigating the blocking of remaining eight apps: As previously mentioned, we were unable to access these apps using VPNs. Other than IP geo-blocking and CDN edge-server restrictions, likely there were additional location revealing parameters sent by the apps to their server, which might have led to censorship. In general, mobile phones (both Android and iOS) present multiple interfaces that reveal the location *e.g.,* GPS, and time-zone information. Before conducting our experiments we ensured that all such user-configurable interfaces were turned off from revealing the location, *e.g.,* we turned off the GPS, changed the time-zone of the phone to a foreign country *etc.* But, we were still unable to access these apps whether we used VPNs or not.

To investigate further, we once again selected TikTok for the detailed analysis (as it was one of the remaining 8 apps). We relayed its traffic via our MITM proxy [9], so as to see if location identifying parameters were being relayed via the requests (ref. §4.1). Interestingly, the TikTok app was sending the country code "IN" as a part of the query string in multiple HTTP requests to the app server, even when all configurable location revealing attributes were turned off (*e.g.,* GPS). Thus, we changed *some* of the parameters (*e.g.,* `op_region`) in the requests to a different country (*e.g.,* "US"), on the fly using the MITM proxy. But still, we suffered censorship.

As a last resort, we reverse-engineered the TikTok app to identify the potential censorship logic (if any), embedded within the app's code. We used the `jadx` decompiler [4] for the same. We obtained a partly decompiled code of the app. Careful inspection of TikTok's code, revealed the use of functions like `getSimCountryISO()`. This function is a part of the Android `TelephonyManager` API and is used to access SIM information. This revealed that TikTok might be fetching country-related information from the SIM card.

Thereafter, we confirmed that only when an Indian SIM is installed in the mobile phone, TikTok sends a `carrier_region=IN` parameter in HTTP requests; otherwise, this parameter was absent. Using the MITM proxy, we changed this parameter's value from "IN" to "US" on-the-fly, and finally, we were able to bypass the censorship. Suppressing the parameter also worked, hence an alternative is to simply remove the installed SIM card. It must be noted that there are other parameters like `op_region=IN` that are also sent in different HTTP requests. But only changing the `carrier_region` parameter resulted in circumvention.

Simply accessing the app without the SIM (through a WiFi network), was sufficient to bypass censorship for each of the remaining 8 apps. This confirmed that these apps were identifying requests from India by probing the installed SIM card. To further confirm our deductions, we ran some additional tests. (1) When our overseas contacts (in an uncensored country, Germany) accessed these eight apps with Indian SIMs on their phones, their requests were also censored. However, with foreign SIMs, they were able to freely access the apps. (2) Indian mobile phones installed with foreign SIM cards had no problems accessing these apps.

Additionally, in dual SIM phones, the apps inspect location information only from the primary SIM. Thus using an Indian SIM in the secondary slot, while leaving the primary slot empty (or installing with a non-Indian SIM) allows uncensored access.

To conclude, the apps transmit country information to the servers by probing the primary SIM. The app servers use this to identify and censor Indian users, irrespective of their actual geographic location. These 8 apps with their censorship mechanisms are enlisted in Table 1 (rows 17–24).

Permanent ban on the apps: At the end of January 2021, the Indian government imposed a permanent ban on the said 220 apps. Interestingly, soon after the permanent ban was enforced, we observed changes in the censorship mechanisms of *only* the previously mentioned eight apps. Out of the eight blocked apps (based on the SIM card's location), we were able to access only one app (MICO Chat) after removing the SIM. The remaining seven apps were inaccessible, even when the Indian SIM card was not present on the phone.

Sl No.	**App Name**	**App Type**	**Censorship Technique Used**		
			Client Source IP	**CDN Edge server**	**Client SIM Card**
1	TikTok	Social	✓	✗	✓
2	Likee	Video Players	✓	✗	✓
3	Kwai	Social	✓	✗	✓
4	UC Browser	Browser	✓	✗	✓
5	FaceU	Photography	✓	✗	✓
6	Hago	Social	✓	✗	✓
7	V-Fly	Tools	✓	✗	✓

Table 2: Censorship mechanisms employed by different blocked apps (after the permanent ban).

This change in censorship mechanism could be attributed to two possibilities: (1) The app servers (for these seven apps) have now adopted geo-blocking (using IP addresses or disabling the content on edge servers serving Indian users), or (2) instead of fetching the country information from the SIMs, the apps might be accessing locale information via other parameters.

To check the possibility of geo-blocking, we removed the SIM card and repeated our previously mentioned four cases that involved changing the source IP address and the CDN edge servers. We confirmed that these 7 apps were now censored using IP geo-blocking.

Further, to check if the location information from the SIM card was still being used or not, we plugged the SIMs back into the phones before accessing the apps. Thereafter, we accessed the apps via VPNs, to ensure foreign source IPs and edge servers. To our surprise, the app servers were still filtering the apps' requests. Examination of the network traffic using MITM proxy revealed the presence of the earlier location parameter (`carrier_region=IN`). Like earlier, masking this parameter (either via the MITM proxy or by simply uninstalling the SIM) makes it difficult for the app servers to identify the country of origin.

We conclude that following the permanent ban, the app servers use both the source IP and the location revealing parameter (fetched by the app from the SIM card), to identify Indian users. We tabulate these 7 apps with their censorship techniques in Table 2.

Summary: *Tier 3* apps are not available in official app stores in India and can be accessed if their *apk* files can be downloaded from third-party sources like `apkmirror.com`. *Tier 2 apps*, even after installation, and restrict Indian users based on their source IP addresses; using VPNs can bypass the server-side filtering. Lastly, *tier 1 apps*, use both source IP and locale information extracted from SIM cards to block the Indian users. Thus, in addition to using VPNs, users need to remove their Indian SIM card and access the apps via WiFi to access the apps. Caution must be exercised by the users because VPNs can help bypass the filtering but may not safeguard them from surveillance as VPN providers have been required to collect and store user data in India [62].

5 Limitations and Future Work

In this work, we studied the blocking mechanisms of 220 officially banned apps in India. However, there can be more apps that are filtered in India but are not made public. In the future, we can use techniques such as those in [65] [47] to identify more apps that are otherwise not known to be filtered. Moreover, our research reveals that if users remove their Indian SIM cards from their phones and use VPNs over WiFI, they can bypass app filtering in India. However, we did not conduct any user studies confirming whether these techniques can be easily adopted by ordinary users without much technical knowledge. Thus, we keep it as our future work. Moreover, Indian users often rely on mobile Internet for Internet connectivity. Thus, removing SIM cards may not be the best solution for bypassing the filtering, and in the future, efforts can be made to change country parameters from `IN` to some other country within the mobile phone. This would enable users to access the blocked apps without removing the SIM card (see §4.3 for more details).

6 Conclusion

Internet filtering has been used by many nation-states in the past. In this paper, we focused on India (a country with more than 880 million Internet subscribers), to analyze the recent app blocking therein. Our research reveals a novel form of "three-tiered" mobile app filtering, with every tier increasing the censorship sophistication. Notably, India does not use the traditional model of censorship for blocking apps—*i.e.,* filtering performed by the ISPs. Rather, following the orders of the Indian government, app publishers are themselves filtering the Indian users on their servers. They achieve server-side censorship either by geo-blocking the clients based on source IP or by identifying Indian users using the country codes fetched from the SIM cards. This is a worrisome trend; other countries may coerce app publishers to adopt these techniques, and app filtering can be achieved simply by updating the app. This would not involve any upgradation in the censorship infrastructure or the involvement of the ISPs. This is particularly concerning for users who access the apps using mobile Internet, as removal of the SIM card would lead to loss of Internet connectivity.

References

1. Indian government bans 220 apps. https://bit.ly/31ECpeA
2. The Indian telecom services performance indicators report. https://www.trai.gov.in/sites/default/files/QPIR_21012021_0.pdf
3. Iran bans messaging apps. https://en.radiofarda.com/a/iran-lawmakers-aim-to-fully-ban-all-foreign-messaging-apps/30802448.html
4. Jadx decompiler. https://github.com/skylot/jadx
5. List of blocked apps. https://pib.gov.in/PressReleseDetailm.aspx?PRID=1635206
6. List of blocked apps. https://pib.gov.in/PressReleasePage.aspx?PRID=1650669
7. List of blocked apps. https://www.pib.gov.in/PressReleasePage.aspx?PRID=1675335
8. Missing links: A comparison of search censorship in china. https://citizenlab.ca/2023/04/a-comparison-of-search-censorship-in-china/
9. Mitm proxy. https://mitmproxy.org/
10. No access: Lgbtiq website censorship in six countries. https://citizenlab.ca/2021/08/no-access-lgbtiq-website-censorship-in-six-countries/
11. Not ok on vk an analysis of in-platform censorship on russia's vkontakte. https://citizenlab.ca/2023/07/an-analysis-of-in-platform-censorship-on-russias-vkontakte/
12. Planet blue coat: Mapping global censorship and surveillance tools. https://citizenlab.ca/2013/01/planet-blue-coat-mapping-global-censorship-and-surveillance-tools/
13. Press Information Bureau (Government of India) officially confirms the ban of 59 Chinese apps. https://pib.gov.in/PressReleseDetailm.aspx?PRID=1635206
14. Pubg to be relaunched in india. https://bit.ly/39viUcu
15. Tiktok blocked in India. https://www.nytimes.com/2020/06/29/world/asia/tik-tok-banned-india-china.html
16. Total downloads of Tiktok app. https://www.theverge.com/2020/4/29/21241788/tiktok-app-download-numbers-update-2-billion-users
17. Triple threat: Nso group's pegasus spyware returns in 2022 with a trio of ios 15 and ios 16 zero-click exploit chains. https://citizenlab.ca/2023/04/nso-groups-pegasus-spyware-returns-in-2022/
18. Website of canadian bobsledder blocked in russia due to collateral filtering. https://citizenlab.ca/2014/02/website-of-canadian-bobsledder-blocked-in-sochi/
19. what-do-ooni-probe-tests-do? https://ooni.org/support/faq/#what-do-ooni-probe-tests-do
20. What is citizen lab? https://citizenlab.ca/about/
21. You move, they follow: Uncovering iran's mobile legal intercept system. https://citizenlab.ca/2023/01/uncovering-irans-mobile-legal-intercept-system/
22. Aceto, G., et al.: Internet censorship in Italy: An analysis of 3G/4G networks. In: IEEE International Conference on Communications (ICC). pp. 1–6. IEEE (2017)
23. Alharbi, F., et al.: Opening Digital Borders Cautiously yet Decisively: Digital Filtering in Saudi Arabia. In: 10th USENIX Workshop on Free and Open Communications on the Internet (2020)

24. Anonymous: Censored Commemoration Chinese Live Streaming Platform YY Focuses Censorship on June 4 Memorials and Activism in Hong Kong. https://citizenlab.ca/2019/06/censored-commemoration-chinese-live-streaming-platform-yy-focuses-censorship-june-4-memorials-activism-hong-kong/
25. Anonymous, et al.: Triplet censors: Demystifying great firewall's DNS censorship behavior. In: USENIX workshop on Free and Open Communications on the Internet (FOCI). USENIX Association (Aug 2020)
26. Aryan, S., et al.: Internet censorship in Iran: A first look. In: USENIX workshop on Free and Open Communications on the Internet (FOCI) (2013)
27. BBC: China bans mobile apps. https://www.bbc.com/news/technology-55230654
28. Beauchamp, T.L.: The belmont report. The Oxford textbook of clinical research ethics pp. 149–155 (2008)
29. Beznazwy, J., Houmansadr, A.: How China detects and blocks shadowsocks. In: Proceedings of the Internet Measurement Conference. pp. 111–124 (2020)
30. Bock, K., et al.: Detecting and evading {Censorship-in-Depth}: A case study of {Iran's} protocol whitelister. In: 10th USENIX Workshop on Free and Open Communications on the Internet (2020)
31. Calder, M., et al.: Mapping the expansion of Google's serving infrastructure. In: Proceedings Internet measurement conference. pp. 313–326. ACM (2013)
32. Calder, M., et al.: Analyzing the performance of an anycast CDN. In: Proceedings of Internet Measurement Conference. pp. 531–537. ACM (2015)
33. Chaabane, A., et al.: Censorship in the wild: Analyzing internet filtering in Syria. In: Proceedings of Internet Measurement Conference. pp. 285–298. ACM (2014)
34. Chai, Z., et al.: On the importance of encrypted-sni (ESNI) to censorship circumvention. In: USENIX workshop on Free and Open Communications on the Internet (FOCI) (2019)
35. Cicalese, D., et al.: A first look at anycast CDN traffic. arXiv preprint arXiv:1505.00946 (2015)
36. Deibert, R.: Citizel lab's list of publication. https://citizenlab.ca/publications//
37. Dittrich, D., et al.: The menlo report: Ethical principles guiding information and communication technology research. Tech. rep., US Department of Homeland Security (2012)
38. Dunna, A., et al.: Analyzing China's blocking of unpublished tor bridges. In: USENIX workshop on Free and Open Communications on the Internet (FOCI) (2018)
39. Ensafi, R., et al.: Examining how the great firewall discovers hidden circumvention servers. In: Proceedings of Internet Measurement Conference. pp. 445–458 (2015)
40. Filasto, A., Appelbaum, J.: OONI: Open observatory of network interference. In: USENIX workshop on Free and Open Communications on the Internet (FOCI) (2012)
41. Gosain, D., et al.: Mending wall: On the implementation of censorship in India. In: International Conference on Security and Privacy in Communication Systems. pp. 418–437. Springer (2017)
42. Griffiths, J.: The Great Firewall of China: How to build and control an alternative version of the internet. Zed Books Ltd. (2019)
43. Jermyn, J., Weaver, N.: Autosonda: Discovering rules and triggers of censorship devices. In: USENIX workshop on Free and Open Communications on the Internet (FOCI) (2017)

44. Khattak, S., et al.: Towards illuminating a censorship monitor's model to facilitate evasion. In: USENIX workshop on Free and Open Communications on the Internet (FOCI) (2013)
45. Knockel, J., Ruan, L.: Bada Bing, Bada Boom: Microsoft Bing's Chinese Political Censorship of Autosuggestions in North America. https://citizenlab.ca/2022/05/bada-bing-bada-boom-microsoft-bings-chinese-political-censorship-autosuggestions-north-america/
46. Knockel, J., et al.: Measuring decentralization of Chinese keyword censorship via mobile games. In: USENIX workshop on Free and Open Communications on the Internet (FOCI) (2017)
47. Kumar, R., et al.: A large-scale investigation into geodifferences in mobile apps. In: 31st USENIX Security Symposium (USENIX Security 22). pp. 1203–1220 (2022)
48. Levis, P.: The collateral damage of internet censorship by DNS injection. ACM SIGCOMM CCR **42**(3) (2012)
49. Li, F., et al.: lib● erate,(n) a library for exposing (traffic-classification) rules and avoiding them efficiently. In: Proceedings of Internet Measurement Conference. pp. 128–141 (2017)
50. MacRae, P.: India-china border tension: app developers, tech sector win with chinese apps banned. https://www.scmp.com/week-asia/economics/article/3170368/india-china-border-tension-app-developers-tech-sector-win
51. Marczak, B., et al.: An analysis of China's great cannon. In: USENIX workshop on Free and Open Communications on the Internet (FOCI) (2015)
52. Marczak, B., et al.: China's great cannon. Citizen Lab **10** (2015)
53. McDonald, A., et al.: 403 forbidden: A global view of CDN geoblocking. In: Proceedings of the Internet Measurement Conference 2018. pp. 218–230 (2018)
54. Nabi, Z.: The anatomy of web censorship in pakistan. In: USENIX workshop on Free and Open Communications on the Internet (FOCI) (2013)
55. Niaki, A.A., et al.: ICLab: a global, longitudinal internet censorship measurement platform. In: IEEE Symposium on Security and Privacy (SP). pp. 135–151 (2020)
56. O'DRISCOLL, A.: Apps banned in Russia. https://www.comparitech.com/blog/vpn-privacy/websites-blocked-russia/
57. Punj, V.: App ban in India could be temporary. https://www.businesstoday.in/current/economy-politics/tiktok-denies-plans-for-legal-recourse-against-ban/story/408741.html
58. Ramesh, R., et al.: Decentralized control: A case study of russia. In: Network and Distributed Systems Security (NDSS) Symposium (2020)
59. Rowley, J.: Certificate pinning. https://www.digicert.com/dc/blog/certificate-pinning-what-is-certificate-pinning/
60. Singh, K., et al.: How India censors the web. In: 12th ACM Conference on Web Science. pp. 21–28 (2020)
61. Sundara Raman, R., et al.: Censored planet: An internet-wide, longitudinal censorship observatory. In: Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security. pp. 49–66 (2020)
62. Varsha Bansal: VPN Providers Flee India as a New Data Law Takes Hold. https://www.wired.co.uk/article/vpn-firms-flee-india-data-collection-law
63. Verkamp, J.P., Gupta, M.: Inferring mechanics of web censorship around the world. In: USENIX workshop on Free and Open Communications on the Internet (FOCI) (2012)
64. Ververis, V., et al.: Understanding internet censorship in europe: The case of spain. In: 13th ACM Web Science Conference 2021. pp. 319–328 (2021)

65. Ververis, V., et al.: Shedding light on mobile app store censorship. In: Adjunct publication of the 27th conference on user modeling, adaptation and personalization. pp. 193–198 (2019)
66. Ververis, V., et al.: Understanding internet censorship policy: The case of Greece. In: USENIX workshop on Free and Open Communications on the Internet (2015)
67. Wang, Z., et al.: Your state is not mine: A closer look at evading stateful internet censorship. In: Proceedings of the Internet Measurement Conference. pp. 114–127 (2017)
68. Winter, P., Lindskog, S.: How the Great Firewall of China is blocking Tor. In: USENIX workshop on Free and Open Communications on the Internet (FOCI) (2012)
69. Wohlfart, F., et al.: Leveraging interconnections for performance: The serving infrastructure of a large CDN. In: Proceedings of the SIGCOMM '18. pp. 206–220
70. Xu, X., et al.: Internet censorship in China: Where does the filtering occur? In: Proceedings of Passive and Active Network Measurement. pp. 133–142. Springer (2011)
71. Xue, D., et al.: Tspu: Russia's decentralized censorship system. In: Proceedings of the 22nd ACM Internet Measurement Conference. pp. 179–194 (2022)
72. Yadav, T.K., et al.: Where the light gets in: Analyzing web censorship mechanisms in India. In: Proceedings of the Internet Measurement Conference 2018 (2018)

A Appendix

A.1 Identifying type of CDNs

In §4.3, we investigated censorship mechanics used by app publishers. This required us to identify whether app publishers were using CDNs or not. Our approach to identify the use of CDNs and their types relies on how DNS and anycast CDNs work. We now describe the same. For each of the 24 apps:

1. We recorded the app's network traffic as a `pcap` file.
2. Using the `pcap` file we identified the unique domains, to which apps communicate.
3. We resolved the same set of domains from 5 different uncensored countries[9], and recorded the IP addresses obtained from each location.
4. Across each location, for each of the domains, we checked if the resolved IP addresses were the same or different.
   (a) If from each of the 5 locations we observed different IP addresses corresponding to the same domain, we classified the domain to be using DNS based CDN.
   (b) Else, if a domain is resolved to the same IP address at every location, then two possibilities exist:
      i. The domain is unicasted (*i.e.,* it is hosted on a non-CDN infrastructure): To confirm the same, we ran `traceroute` from the five geographically diverse VPSs to the same IP addresses. If all

[9] We used VPSes in these countries for the same.

the traceroute paths end in the same country, we classified it as unicasted. [We used the Maxmind geolocation database to map the IP addresses of the routers (in the traceroute path) to their respective country. We observed that at least the last two IP addresses in all the traceroutes (to unicasted domains) belong to the same country.]

ii. The domain is anycasted: If the traceroute paths end in separate countries (likely the ones where they originate), we classified the domain as anycasted.

We observed that, on average, each of the apps was communicating with 8 unique domains, and the majority of these (> 6) were using DNS-based CDNs. A few of them were anycasted (or unicasted).

A.2 Why app publishers are filtering Indian users

Overall, it is natural to ask why app publishers are filtering Indian users when many of them do not operate from India anymore. There was anecdotal evidence that they were hoping that the ban would be temporary [57], and thus they obliged with it. Some companies were in communication with the Indian government and were awaiting their approval for relaunching the apps [14]. However, in January 2021, the government imposed a permanent ban on all the 220 apps (ref. §4.3). Even then, the app publishers not only continued the filtering but also imposed stricter censorship for Indian users, while the ISPs continued to have no role in this. Before the ban, 8 apps were using only SIM-based censorship, but after the ban, 7 of them adopted IP geo-blocking as well. The precise reason for app publishers (and not the ISPs) filtering Indian users remains unclear.

A.3 The apps unavailable in official play stores in non-censoring countries

In §4, we mentioned that our overseas contacts (residing in uncensored countries) were also unable to find 60 apps that were blocked in India. Thus, we assumed that they were likely defunct, or else they would be available at least in uncensored countries.

However, it could be argued that some of these apps might be functional yet unavailable in uncensored countries. This is because some of these apps may have been launched specifically for Asia (or India). Thus, to confirm that this was not the case, and 60 apps were likely defunct, we searched them on third-party sources *e.g.,* apkmirror.com. Barring a few, the installation packages of most of these apps were unavailable on such sites as well. For the few that we found, they were last updated around four to five years ago. This indicated that they were no longer operational. Our overseas contacts installed them and confirmed that they were unable to access them. Thus, we ignored these 60 apps and focused on the remaining 160 apps.