CloudAnalyzer: Uncovering the Cloud Usage of Mobile Apps

Martin Henze, Jan Pennekamp, David Hellmanns, Erik Mühmer,
Jan Henrik Ziegeldorf, Arthur Drichel, Klaus Wehrle
Communication and Distributed Systems, RWTH Aachen University, Germany
{henze,pennekamp,hellmanns,muehmer,ziegeldorf,drichel,wehrle}@comsys.rwth-aachen.de

ABSTRACT

Developers of smartphone apps increasingly rely on cloud services for ready-made functionalities, e.g., to track app usage, to store data, or to integrate social networks. At the same time, mobile apps have access to various private information, ranging from users' contact lists to their precise locations. As a result, app deployment models and data flows have become too complex and entangled for users to understand. We present CloudAnalyzer, a transparency technology that reveals the cloud usage of smartphone apps and hence provides users with the means to reclaim informational self-determination. We apply CloudAnalyzer to study the cloud exposure of 29 volunteers over the course of 19 days. In addition, we analyze the cloud usage of the 5000 most accessed mobile websites as well as 500 popular apps from five different countries. Our results reveal an excessive exposure to cloud services: 90 % of apps use cloud services and 36 % of apps used by volunteers solely communicate with cloud services. Given the information provided by CloudAnalyzer, users can critically review the cloud usage of their apps.

CCS CONCEPTS

• **Security and privacy → Human and societal aspects of security and privacy**; • **Networks →** *Network protocols*; • **Human-centered computing → Ubiquitous and mobile computing**;

KEYWORDS

Privacy, Smartphones, Cloud Computing, Traffic Analysis

ACM Reference Format:
Martin Henze, Jan Pennekamp, David Hellmanns, Erik Mühmer, Jan Henrik Ziegeldorf, Arthur Drichel, Klaus Wehrle. 2017. CloudAnalyzer: Uncovering the Cloud Usage of Mobile Apps. In *Proceedings of the 14th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (MobiQuitous 2017).* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3144457.3144471

1 INTRODUCTION

Smartphones have become important for storing and accessing personal information, ranging from contacts and calendar entries over pictures to work documents [26]. Additionally, smartphones *produce* data through their sensors which, e.g., enables localization or activity recognition [31]. With the right permissions, this abundance of sensitive data can be easily accessed by mobile applications (apps) through dedicated APIs [46]. Indeed, app developers increasingly rely on user data to improve the functionality of their apps or to increase revenue with targeted advertisement [40].

At the same time, major parts of apps' backend functionality, including tracking and advertising, are nowadays realized via cloud services. These services range from cloud infrastructure and content delivery networks (e.g., AWS and CloudFront) over reporting, analytics, and advertisement services (e.g., Crashlytics, Flurry, and AdMob) to consumer services (e.g., YouTube and Facebook). We discover that the most popular apps on Google Play utilize 4.3 cloud services on average, which highlights the prevalence of cloud usage.

In this situation, users have no knowledge about which cloud services are utilized by apps running on their smartphones. However, combining the sensitive data stored and sensed by smartphones with cloud computing—characterized by de facto monopolies, technical complexity, inherent non-transparency, and opaque legislation—raises severe privacy risks [35, 60]. Even worse, cloud services can be realized on top of each other, leading to indirect cloud exposure which is even harder for users to grasp. As an example, our work reveals that Unity (a popular game development platform) utilizes Amazon EC2 to (partly) deliver its services.

Any cloud service receiving sensitive information can use it for unintended purposes, e.g., personalized advertising [40] or forwarding to other entities [36]. Furthermore, users have no guarantee that their data is handled according to their legal requirements [36, 40]. Resulting from the de facto monopolized landscape of cloud services, data is further susceptible to breaches as evidenced by the compromise of 1 billion Yahoo accounts [29]. To put users back into control, it is important to raise their awareness of these risks [43] and provide them with means to protect their privacy.

Related work confirms the privacy risks of the access of apps to an abundance of private information. To assess and counter these risks, approaches aim at detecting privacy leakage by analyzing traffic [51, 57] or tracking apps' data flows [9, 26]. These related works provide information on *what* data is leaked. So far, a way for smartphone users to detect *where* (to which cloud services) their data is leaked, as a foundation to protect their privacy, is missing.

To bridge this gap between users' knowledge and information required to enforce their privacy, we present CloudAnalyzer, which provides users with detailed statistics of their personal cloud exposure caused by their smartphone apps. To achieve this goal, CloudAnalyzer *locally* monitors the network traffic produced by apps running on a user's device and compares observed communication patterns to 55 representative cloud services. Apart from revealing the hidden exposure to cloud services caused by smartphone apps,

MobiQuitous 2017, November 7–10, 2017, Melbourne, VIC, Australia

ACM ISBN 978-1-4503-5368-7/17/11…$15.00
https://doi.org/10.1145/3144457.3144471

CloudAnalyzer also detects the prevalent indirection in cloud usage where cloud services subcontract each other to realize their functionality. Based on CloudAnalyzer's observations, we support users in critically reviewing their exposure to cloud services and, as a result, change their app usage behavior or even decide to refrain from using certain apps. Likewise, CloudAnalyzer is a valuable tool for researchers to understand the characteristics of the usage of cloud services by smartphone apps and the relationships between cloud services. The following are our main contributions:

Representative Set of Cloud Services: We perform a thorough analysis of the landscape of cloud services commonly utilized by mobile apps today. Based on this, we derive the most influential services in each category of mobile cloud services. Our resulting set of 55 representative cloud services serves as a foundation for detecting cloud usage of mobile apps.

Cloud Usage Detection: We develop a methodology to identify cloud services based on patterns that can be directly obtained from passively observed network traffic. Our methodology identifies, besides the cloud service(s) an app is directly communicating with, the indirect use of cloud resources. We present CloudAnalyzer, an implementation of our methodology for unmodified Android devices that transparently monitors cloud usage of apps.

User Study and Measurements: We utilize CloudAnalyzer to study the cloud usage of 29 devices over the course of 19 days. Additionally, we study the cloud entanglement of the 5000 most popular mobile websites. Finally, we investigate the cloud usage of 500 popular apps in five countries. CloudAnalyzer reveals an alarming rate of 90 % of apps using cloud services and 36 % of apps used by volunteers communicating solely with cloud services.

2 MOBILE CLOUD SERVICES AND PRIVACY

Developers for mobile platforms increasingly rely on cloud services [65]. Their motivation ranges from reduced effort over cost reductions to the possibility to integrate third-party services, e.g., advertising networks. We first provide an overview of the landscape of mobile cloud services and derive a representative set of services. Based on this, we distill privacy risks in the face of potentially sensitive data collected by smartphones and discuss related work.

2.1 The Landscape of Mobile Cloud Services

To understand the extent of cloud exposure through mobile apps and the resulting privacy risks, we identify classes of mobile cloud services and their interweaving. As shown in Figure 1, a major portion of cloud usage originates from software development kits (SDKs) that app developers include to realize functionality ranging from interaction with social networks over crash reporting to targeted advertisement [13]. Depending on the individual SDK, different cloud services are utilized. In the following, we discuss the five different classes of mobile cloud services and their relationships. Furthermore, we compile a representative list of the most influential services for each class. We provide the full list of the 55 cloud services that we selected in Table 1.

Cloud Infrastructure (CI). Developers of mobile apps use cloud infrastructure (i.e., computing and storage resources) to operate the backend for their apps (instead of operating own servers). In our work, we consider the most important infrastructure providers as identified by Canalys' revenue analysis [16] and Skyhigh's study of application deployment [56]. The services covered by these studies account for a market share of 68.7 % respectively 85.7 %.

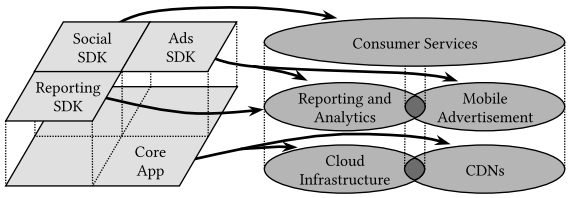

Figure 1: In the landscape of mobile cloud services, services on upper layers can, but not necessarily have to, rely on services on lower layers to provide their functionality.

Content Delivery Networks (CDN). To reliably, scalably, and timely deliver static content, Content Delivery Networks (CDNs) rely on globally distributed infrastructure. They can be realized on top of cloud infrastructure or built on dedicated infrastructure. We analyze all CDNs that have a market share $\geq 1\%$ in Datanyze's measurements of 1 M popular websites [19]. Together, the CDN services in our analysis have a market share of more than 90 %.

Reporting and Analytics (R&A). To support app developers with statistics on errors and app usage, reporting services track errors (e.g., crashes) of apps while analytics services gather statistics on the usage of apps (ranging from gathering user statistics to tracking user interaction). We cover all services behind reporting and analytics libraries with $\geq 1\%$ of installs according to AppBrain's measurements [5, 6]. Libraries that do not operate own cloud services are excluded from our analysis (e.g., ACRA).

Mobile Advertisement (MA). App developers often rely on mobile advertisement services to monetize their apps [55]. These services are usually realized on cloud infrastructure and/or CDNs. In our work, we include all services behind ad network libraries with $\geq 1\%$ of installs in AppBrain's statistics [4, 8]. In addition, we incorporate the advertisement companies with the highest traffic as identified by measurements of Pujol et al. [49].

Consumer Services (CS). Services directly addressing and interacting with consumers often rely on cloud infrastructure and CDNs, e.g., social networks, communication and video platforms, as well as file storage services. Such consumer services (e.g., Facebook and Twitter) can often be integrated into apps through an SDK. To capture this effect, we include the social network libraries with $\geq 1\%$ of installs according to AppBrain [7]. Furthermore, we cover the services with the highest amount of mobile traffic in North America according to Sandvine [53]. Additionally, we incorporate the 20 most prominent consumer services as identified by Skyhigh [56].

2.2 Privacy Risks of Mobile Cloud Services

When considering the landscape of mobile cloud services, it becomes evident that this deployment model poses serious privacy risks. Most notably, the challenge of protecting privacy is more complex and important on smartphones compared to traditional deployments. First, smartphones are equipped with a large number of sensors, facilitating detailed monitoring and tracking [31]. Second, users interact with their smartphones throughout the day, leading to a growing amount of sensitive information and meta data [31]. Thus, smartphones increasingly cover important aspects of

private life. When outsourcing potential sensitive data to (mobile) cloud services, these privacy risks further amplify—mainly due to the centrality, technical complexity, non-transparency, and opaque legislation of cloud computing [36], as we detail in the following.

Centrality. The cloud market is de facto centralized with a small number of services jointly dominating the market [56]. As a result, these services are a valuable target for attackers [36], exemplified by the attack on Yahoo in 2013, compromising 1 billion user accounts [29]. Users are very much aware of these imminent risks [40], and they significantly hinder cloud adoption [36, 61].

Technical Complexity and Non-Transparency. At the same time, the mobile cloud service landscape is *technically complex* and *non-transparent*: Mobile cloud services often subcontract other cloud services [36], e.g., to avoid operating own infrastructure, to increase scalability, or to strengthen resilience against attacks. This entanglement forces users to trust an unknown number of third-party cloud services with the sensitive data of their smartphones. Most notably, the situation nowadays is so complex that it is impossible for users to grasp. Hence, users are in need of support for taking an informed decision regarding their privacy.

Opaque Legislation. Given this technical complexity and non-transparency, the jurisdiction users' data falls under is often unclear, hence, offering users very limited legal protection [20]. Furthermore, legislation in many countries allows government agencies, e.g., law enforcement, to access and intercept data in the cloud [30, 36]. This threat became evident after the recent global surveillance disclosures [30]. Even app developers often fail to know where data (their app is responsible for) flows to [32].

As a result of these increased privacy risks, users perceive a loss of control over their data [36, 40, 61]. Hence, providing users with the means to counter this perceived loss of control when their sensitive data is sent to cloud services is an important challenge.

2.3 Related Work

As we discuss in the following, different lines of research provide valuable input for our goal to uncover cloud usage of apps.

Mobile Network Traffic. Xu et al. [65] study the usage behavior of apps in a cellular network. ProfileDroid [64] studies Android apps to understand their network behavior. AntMonitor [42] and Haystack [50] realize mobile measurement platforms that enable researchers to investigate the network usage of apps at large. With the goal to detect leaked private data, PrivacyGuard [57] and ReCon [51] intercept network traffic of apps. Ferreira et al. [28] study the network behavior of apps to differentiate between (in)secure connections and the location of communication endpoints.

These works focus on the patterns and content of apps' network communication (and partially on resulting privacy risks). They provide us with a solid foundation for our work, since they derive an understanding of the network-level behavior of mobile apps and offer mechanisms to detect leaked private data in traffic. In contrast to our work, these works neglect the added privacy risks of the complex and non-transparent interweaving of mobile apps with cloud services common today. As a first step in this direction, TRINICS [38] lays out the idea of comparing the cloud usage of different users to give them feedback on their privacy risks.

Service	Source(s)	CI	CDN	R&A	MA	CS	Additional Brand Names
AdColony	[4] [8]				●		
Adjust	[4] [5]			●	●		
Akamai	[19]		●				
Alibaba	[16]	●	○	●		○	Umeng
Amazon	[4] [16] [56]	●	●	○	●	●	A. Mobile Ads, A. S3, A. Web Services (AWS), Cloudfront, Twitch
Appboy	[7]			○		●	
Apple	[53]					●	iCloud, iTunes
AppLovin	[4]			○	●		
Appnext	[4]				●		
AppNexus	[49]				●		
AppsFlyer	[4] [5]			●	●		
Apteligent	[6]			●			Crittercism
Chartboost	[4] [8]				●		
Cloudflare	[19]		●				
comScore	[5]			●			ScorecardResearch
Criteo	[49]				●		
Dropbox	[56]					●	
Evernote	[56]					●	
Facebook	[7] [53] [56]				○	●	Atlas, Instagram, F. Messenger, WhatsApp
Fastly	[19]		●				
GitHub	[56]					●	
Google	[4] [6] [7] [16] [53] [56]	●	○	●	●	●	AdMob, Crashlytics, DoubleClick, Fabric, Gmail, G. Analytics, YouTube
imgur	[56]					●	
Incapsula	[19]		●				
InMobi	[4]				●		
KeyCDN	[19]		●				
Kochava	[4] [5]			●	●		
Leadbolt	[4]				●		
LinkedIn	[56]					●	
Localytics	[5]			●			
Microsoft	[6] [16] [56]	●	●	●	○	●	Bing, HockeyApp, Microsoft Azure, Office, OneDrive, Outlook, Skype
Mixpanel	[5]			●			
Netflix	[53]					●	
Oracle	[16]	●			○		
Pinterest	[56]					●	
Rackspace	[19] [56]	●	●				
RNTSMedia	[4] [7]				●	●	Fyber, HeyZap
Smaato	[4]				●		
Snap	[53]					●	SnapChat
SoftLayer	[16] [56]	●	○		○		
SoundCloud	[56]					●	
StackPath	[19]		●				Highwinds, MaxCDN
StartApp	[4] [7] [8]			○	●		
StumbleUpon	[56]					●	
Supersonic	[4]			○	●		IronSource, mobileCore, StreamRail
Tapjoy	[4]				●		
Tune	[4] [5]			●	●		MobileAppTracking
Twitter	[4] [7]				●	●	MoPub, Vine
Unity	[4] [8]			○	●		Applifier
Verizon	[4] [8] [19] [56]	○	●	●	●	●	AOL, EdgeCast, Flickr, Flurry, Millennial Media, Nexage, Tumblr, Yahoo
Vimeo	[56]					●	
VK	[7]					●	
Vungle	[4] [8]				●		
WeChat	[7]					●	
Yandex	[5] [6]			●		○	

Table 1: Our representative set of 55 services covers the different classes of mobile cloud services. We use ● to denote representative services for each class of mobile cloud services, while ○ denotes less prominent services for a class.

Cloud Traffic. Bermudez et al. [12] identify DNS responses as viable input to identify cloud services. Subsequently, they detect network traffic flowing to Amazon Web Services [11]. Drago et al. [23] rely on DNS and TLS packets to study cloud storage systems based on passive network observations. To understand if and how web services are realized on top of cloud infrastructure, He et al. [34] perform DNS probing for popular web services. To understand the prevalence of cloud services in the email landscape, Henze et al. [39] analyze the cloud exposure of email users. These works perform large scale measurements to understand the anatomy of cloud services. Their methodology provides valuable input for our

approach of detecting cloud usage on smartphones. However, these approaches do not consider the privacy risks of smartphones communicating with cloud services, which is our main focus.

Mobile Advertising. Vallina-Rodriguez et al. [62] study mobile advertising based on traffic within the network of a mobile carrier. Focusing on advertisement *libraries* on Android, Book et al. [14] analyze the use of permissions for mobile advertising. From a different perspective, Chen et al. [18] investigate the privacy risks of mobile analytics services. Complementing these works, Vallina-Rodriguez et al. [63] study mobile advertising and tracking based on network traces of volunteers. Finally, Brookman et al. [15] measure the capability of advertisers to link users *across* different devices. These works highlight privacy risks of forwarding data to advertising services. However, mobile advertising is only one part of the mobile cloud landscape and, as we show, privacy risks further exacerbate when looking at the *complete* mobile cloud landscape.

Data Flow Tracking. Tracking the flow of data within smartphone apps allows detecting the leakage of sensitive data to third parties, even if apps try to obfuscate that they are sending out sensitive data. AndroidLeaks [31] and FlowDroid [9] are *static* flow tracking systems that are used ahead of time to detect potential leaks of sensitive information by covering all possible execution paths of an app. In contrast, *dynamic* flow tracking systems such as TaintDroid [26] and TaintART [59] track data flows during execution of an app to identify actual data leakage that occurs while executing an app. One challenge of data flow tracking is to identify whether an identified data flow is benign or constitutes a privacy risk. To this end, AppIntent [66] identifies data flows that have not been triggered by the user and marks those as critical.

Concluding, mobile operating systems today counter privacy risks by measures ranging from access control to sandboxing [10, 25]. These protect against malicious apps, but do not restrain privacy invasive apps exploiting *granted* permissions. Hence, users' privacy is insufficiently protected [58], especially since users remain oblivious of their exposure to a plethora of cloud services. Related work that addresses this challenge primarily focuses on detecting which private *content* is leaked from smartphones. In contrast, we study the privacy risks resulting from the *destination* of leaked content, especially with the advent of mobile cloud services.

3 DETECTING CLOUD USAGE OF APPS

Given the privacy risks when data is sent from smartphones to the cloud, users must be empowered to effectively assess these risks to make an informed decision about which apps to use or not. To this end, users need detailed information about the quality and extent of cloud exposure induced by apps. However, existing approaches today primarily focus on detecting the leakage of sensitive information, irrespective of where data is communicated to. Additionally, cloud exposure of users through their apps is highly individual, depending on the utilized apps and users' behavior when interacting with apps. Hence, users are in need of an *individual* assessment of the privacy risks resulting from cloud usage of their apps.

To achieve this goal, we present CloudAnalyzer that uncovers the cloud usage of smartphone apps by passively observing network traffic directly on users' devices. By doing so, we neatly complement existing work, especially on data flow tracking, since we enable

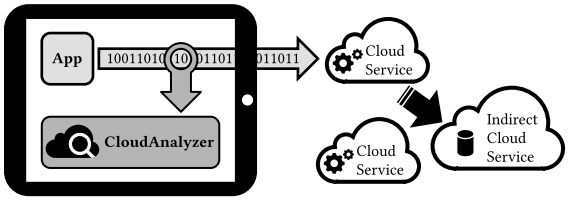

Figure 2: To uncover cloud usage, CloudAnalyzer analyzes network traffic created by apps directly on users' smartphones for communication with cloud services.

the attribution of privacy leaks to responsible cloud services. This attribution empowers users to adequately assess their individual privacy risks and take appropriate counter measures, e.g., uninstall a certain app or change their usage behavior. In the following, we first describe the overall architecture of CloudAnalyzer. We then present our methodology for dissecting network traffic to detect cloud usage and describe how we realize CloudAnalyzer for commodity off-the-shelf Android devices.

3.1 System Overview

CloudAnalyzer operates on network traffic of smartphone apps to detect contacted cloud services. We decided to realize all functionality for uncovering cloud usage solely within the control of the user, i.e., directly on her device. Since network traffic itself is extremely sensitive, processing it outside users' control would strongly contradict our goal of *improving* user privacy.

Our system for uncovering cloud usage of smartphone apps, CloudAnalyzer, operates as shown in Figure 2. Whenever an app uses one of the communication interfaces (cellular or wifi) to contact an Internet service, CloudAnalyzer *locally* obtains a copy of the communication transcript. Subsequently, CloudAnalyzer dissects the captured traffic to identify contacted cloud services. Based on this, CloudAnalyzer attributes the communication flow to one or multiple identified cloud services. CloudAnalyzer collects aggregated statistics on the amount of network packets and traffic that has been sent to and received from a specific cloud service, differentiating between direction of communication, encrypted and unencrypted communication, user-initiated and background traffic, as well as the used communication interface (cellular or wifi).

3.2 Dissecting Traffic to Detect Cloud Usage

At the core of CloudAnalyzer sits our methodology to detect cloud usage based on network traffic. We comprehensively analyzed the communication behavior of smartphone apps to derive different approaches for reliably identifying contacted cloud services.

IP Addresses. IP addresses are identifiers assigned to each networked computer [48] and hence also to each server that is used to realize cloud services. This can be used to identify the operator of the infrastructure a service is realized on (cloud infrastructure or CDN, cf. Section 2.1). To determine that a contacted server is operated by a cloud service, we rely on information from providers: Many cloud services (e.g., Amazon, Microsoft, Google, SoftLayer) make their IP addresses public, e.g., to enable customers to configure firewalls [38]. Often, such published information contains a (textual) description of the location of the data center, enabling us to also identify the corresponding jurisdiction. While IP addresses

often allow us to detect infrastructure services, we have to analyze application layer protocols to also detect services that fail to publish their IP addresses as well as services realized at higher layers.

DNS Responses. The domain name system (DNS) translates (human readable) domain names to IP addresses [44]. Whenever a smartphone app requests a resource from a specific domain name, the Android system transparently issues a DNS request to translate this domain name to an IP address. By observing subsequent DNS responses from a DNS name server, we derive the actual contacted service(s) [12, 23]. We mark all subsequent communication with this IP address as belonging to the identified cloud service. Using this approach, it is even possible to identify multiple services in the case of indirect cloud usage. Furthermore, some cloud services (e.g., Amazon) use domain names that contain information about the data center location, easing the detection of the applicable jurisdiction.

Server Name Indication. With the increasing use of encryption, server name indication (SNI) enables operators to still serve multiple domain names from one IP address. Support for SNI is available for the widely deployed transport layer security (TLS) protocol [24] and the evolving QUIC protocol [41]. Since clients send the SNI in plaintext, we can observe this information and utilize it, similar to DNS responses, to identify contacted cloud services.

TLS Certificates. When using TLS encrypted connections, servers have to authenticate themselves to clients using a TLS certificate [21]. This certificate typically identifies the institution operating a service. To establish trust into certificates, they have to be validated by a trusted certificate authority. Hence, the information in TLS certificates constitutes an especially reliable source for identifying cloud services. We use domain names and information about the organization holding a certificate to identify services.

In CloudAnalyzer, we use the above approaches to detect cloud exposure for traffic *flows* as follows. Whenever one of the above approaches detects a cloud service, we mark any future packets of the same traffic flow as being exposed to this cloud service as well. Strictly working on traffic flows prevents false classification that might result from more lenient approaches such as analysis of traffic patterns. Most notably, the combination of multiple of the above approaches also enables the detection of indirect cloud usage, i.e., one cloud service realized on top of another cloud service. In this case, we assign one traffic flow to more than one cloud service and use the most specific information available on these different cloud services, e.g., when assigning data center locations.

To apply the above approaches to detect *specific* cloud services, we need to create patterns for each cloud service. For example, we need to know which IP addresses a cloud service uses or how a cloud service's TLS certificate looks like. To this end, we researched these patterns for our 55 representative cloud services (cf. Section 2.1). Here, we relied on information provided by cloud services as well as other public information (e.g., filter lists for advertisement). Subsequently, we verified that our selection of cloud services and detection patterns is indeed representative by checking IP addresses, DNS and SNI domain names, as well as TLS certificates for a random subset of our measurements of the most used apps (cf. Section 4.3).

Our approach of creating patterns for representative cloud services might not necessarily detect all cloud services. However, given our goal to support users in empowering their privacy, we strive for correctness over completeness. Our rationale here is to keep users clear of incorrect information which might result from probabilistic approaches such as the topological analysis of autonomous systems [27] or IP geolocation databases [47]. Instead, the information provided by CloudAnalyzer constitutes a solid lower bound for the entanglement of cloud services. In return, we accept that we might be unable to detect cloud exposure to a few less important and seldom used cloud services. Additionally, cloud services might deliberately try to obfuscate their network communication. However, during our extensive tests of CloudAnalyzer, we observed only a single, negligible attempt to obfuscate a mobile advertising service.

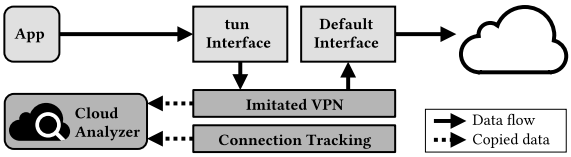

Figure 3: CloudAnalyzer accesses network packets by locally imitating a VPN using Android's `VPNService`.

3.3 Integrating CloudAnalyzer into Android

The core idea of CloudAnalyzer is to detect cloud entanglement in network traffic. Since network traffic is of sensitive nature, we have to realize CloudAnalyzer on users' devices. However, mobile operating systems such as Android lack an interface to access network traffic without system modification (i.e., rooting or custom firmwares). Since we mostly target not technically-minded users, we cannot dictate modifications to the operating systems in contrast to related work [26, 59]. Instead, we aim at a solution that enables users to uncover their cloud exposure simply by installing an app through well-established channels (e.g., Google Play).

To achieve this goal, we use an indirect path to access network traffic on *unmodified* Android devices: We realize an *imitated VPN* to gain access to the device's network traffic using the `VPNService` of the Android SDK [42, 50, 57] as shown in Figure 3 to create a `tun` interface that redirects all network traffic of the Android device into our imitated VPN. Our imitated VPN receives raw IP packets and performs two tasks: (i) it creates a copy of each received network packet which is then forwarded to CloudAnalyzer for further processing and (ii) it forwards the raw IP packets to their destination. The latter proves technically difficult, since Android prohibits the creation of raw sockets. Hence, our imitated VPN implements the essential parts of a Layer 3 and 4 network stack to forward data from the `tun` interface over a normal Java socket to an Internet host. This approach includes memorizing the state of all open connections to be able to retranslate payload *received* on a socket to corresponding IP packets to send them back to the application over the `tun` interface. Related work shows that this can be realized at modest throughput and energy costs [42, 50, 57].

Besides protecting privacy, capturing and analyzing network traffic directly on users' devices gives us an additional advantage: It allows to correlate network packets to the application they originate from. To this end, we track connections by extracting the user ID of the app that started a specific network flow from the kernel's `proc` directory. Subsequently, we translate this user ID to the package name of the app using Android's `PackageManager` API.

CloudAnalyzer's way of utilizing Android's `VPNService` prevents users from using an actual VPN connection. This limitation

can be circumvented by integrating CloudAnalyzer either into the VPN client or server. On a different perspective, CloudAnalyzer asks for permission to access sensitive network traffic and hence users need to trust CloudAnalyzer not to misuse this privilege. This requirement holds for all privacy enhancing technologies working on network traffic and we are convinced that increased privacy outweighs the required trust. Furthermore, unlike related work [50, 57], we do not require users to install a CA certificate to perform man-in-the-middle analyses. Hence, CloudAnalyzer intentionally remains oblivious of the *content* of encrypted sensitive communication.

In summary, by using Android's `VPNService` and keeping track of connections, we can observe network traffic on off-the-shelf Android devices (Version 4.4 and newer) without the need for system modifications. Furthermore and in contrast to in-network traffic monitoring, we are able to associate network packets to the app they originate from. Hence, we can use CloudAnalyzer to check the network traffic of an app for communication with cloud services.

4 REAL-WORLD CLOUD ENTANGLEMENT

We now set out to uncover the cloud usage of mobile apps using CloudAnalyzer. To this end, we first discuss our observations derived from running CloudAnalyzer on devices of volunteers. Subsequently, we report on additional measurements of popular mobile websites and the most used apps in multiple countries to highlight different aspects of cloud usage at larger scales.

4.1 Cloud Entanglement on User Devices

We begin our study by analyzing the cloud usage of actual users on their mobile devices. To this end, volunteers installed CloudAnalyzer on 29 devices and collected statistics on the cloud exposure caused by their apps over the course of 19 days.

Study Design. We advertised our study using mailing lists and personal contacts to attract volunteers, but did not offer monetary remuneration for participating in our study. People were already motivated to participate through the possibility to gain interesting insights into their exposure to cloud services. Study participants could at any time pause CloudAnalyzer's traffic analysis or examine their cloud usage through a GUI. As a result, volunteers could have changed their usage behavior based on the information provided by CloudAnalyzer. However, since our focus in this work lies on untangling the mobile cloud landscape, our experiments were not designed to capture these effects. Still, one volunteer contacted us to report on uninstalling an app based on the information provided by CloudAnalyzer, and we will further study such aspects in future work. We collected aggregated statistics on cloud usage detected by CloudAnalyzer as well as general statistics such as the amount of time CloudAnalyzer was running and the total amount of network traffic (serving as a baseline). For our analysis, we only consider data from days where CloudAnalyzer was running for at least 20 hours (to prevent partial measurements). In total, we were able to collect data for 347 days of mobile device usage covering 383 apps.

Privacy and Ethical Considerations. As the goal of CloudAnalyzer is to empower users to execute their right to privacy, we designed our study such that the risk of (inadvertently) harming the privacy of our volunteers is minimized. To this end, we followed the principles of privacy by design [17] and ethical research guidelines

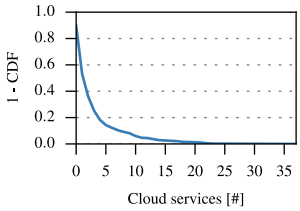

Figure 4: Cloud services accessed by user devices.

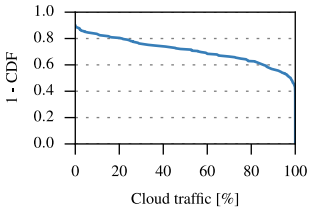

Figure 5: Cloud traffic produced by individual apps.

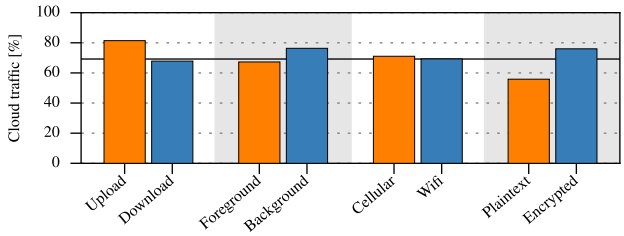

Figure 6: The fraction of cloud usage varies across the different dimensions of network traffic.

[22]. Notably, we are only interested in technical usage characteristics of cloud services, not in user behavior. Hence, we neither collected personally identifiable information nor other statistics on our volunteers. In fact, we do not even know who participated in our study (unless volunteers actively disclosed their participation). We strictly minimized the collection of data to the amount necessary and aggregated all statistics directly on the volunteers' devices at a granularity of one day (to minimize the risk of de-anonymizing users based on temporal information). Users were educated about the extent and purpose of data collection and had to explicitly agree to these conditions. We obliged ourselves to not share collected data with third parties. Furthermore, we gave users the possibility to exclude specific apps from the analysis. Finally, we offered them the option to disable automatic uploads of their statistics to manually review collected information before sending it to our server.

Overall Cloud Usage. In Figure 4 we show the complementary cumulative distribution function (1–CDF) for the *number of used cloud services per app* across all devices. On average, each app connects to 3.2 cloud services. Notably, 89.8 % of apps contact cloud services. Naturally, web browsers contact many cloud services (e.g., Chrome with 37 services), but also less obvious candidates, e.g., the fitness tracking apps `com.withings.wiscale2` (12) and `com.myfitnesspal.android` (11), contact many services.

When looking at the *fraction of cloud traffic per app* in Figure 5, we make an even stronger observation. While 89.8 % of apps produce cloud traffic, 53.8 % of apps send 95 % or more of their traffic to cloud services. Notably, 35.5 % of apps send *all* their traffic to cloud services. These numbers show that cloud entanglement is a real problem, concerning a majority of apps and often leading to complete exposure of apps' communication to cloud services.

Different Dimensions of Cloud Traffic. Cloud traffic can be generated in various ways, e.g., triggered by users or automatically by background processes, leading to different privacy risks. We study the *different dimensions of cloud traffic* in Figure 6, where we compare the fraction of traffic to and from cloud services along different dimensions of network traffic to the overall fraction of

Service	Traffic	Apps
Google	34.66 %	54.57 %
Facebook	9.71 %	24.80 %
Amazon	8.76 %	65.27 %
Akamai	5.92 %	27.94 %
Fastly	5.54 %	13.32 %
imgur	3.04 %	4.18 %
Cloudflare	1.27 %	12.27 %
Snap	1.07 %	1.04 %
Twitter	0.60 %	9.14 %
Verizon	0.60 %	16.45 %
StackPath	0.45 %	7.57 %
Microsoft	0.25 %	8.62 %
Chartboost	0.19 %	0.26 %
Dropbox	0.10 %	1.31 %
SoundCloud	0.07 %	2.87 %
GitHub	0.05 %	2.87 %
AppNexus	0.04 %	7.57 %
Criteo	0.04 %	5.74 %
Netflix	0.03 %	0.52 %
Tapjoy	0.03 %	0.26 %

(a) Top 20 cloud services for user devices

Service	Traffic	Sites
Akamai	13.74 %	43.24 %
Google	12.02 %	84.50 %
Amazon	10.33 %	76.82 %
Cloudflare	8.97 %	48.76 %
Fastly	2.91 %	41.08 %
Verizon	2.12 %	24.28 %
Facebook	1.56 %	47.86 %
StackPath	1.16 %	13.38 %
Microsoft	0.59 %	13.78 %
Twitter	0.53 %	10.46 %
Incapsula	0.47 %	3.26 %
Alibaba	0.46 %	3.58 %
Yandex	0.36 %	3.18 %
AppNexus	0.33 %	33.02 %
Vimeo	0.15 %	0.48 %
LinkedIn	0.10 %	2.28 %
Oracle	0.09 %	6.36 %
Criteo	0.09 %	9.34 %
GitHub	0.08 %	2.32 %
Rackspace	0.06 %	0.42 %

(b) Top 20 cloud services for mobile websites

Service	Traffic	Apps
Google	24.38 %	80.00 %
Amazon	20.90 %	80.27 %
Akamai	13.26 %	56.34 %
Facebook	5.84 %	50.98 %
Verizon	4.76 %	38.58 %
Unity	3.88 %	17.49 %
Chartboost	2.72 %	10.17 %
Fastly	2.12 %	17.69 %
StackPath	1.93 %	16.95 %
AppLovin	1.81 %	7.59 %
Cloudflare	1.38 %	18.58 %
Vungle	1.34 %	5.90 %
Microsoft	0.99 %	9.36 %
AppsFlyer	0.92 %	18.85 %
Yandex	0.71 %	3.86 %
Twitter	0.68 %	12.34 %
Criteo	0.48 %	13.69 %
Tapjoy	0.46 %	4.34 %
StartApp	0.46 %	3.25 %
Supersonic	0.42 %	4.68 %

(c) Top 20 cloud services for popular apps

Table 2: The cloud services with the highest traffic differ between user devices, mobile websites, and popular apps.

cloud usage (solid line). We observe a higher fraction of cloud usage in uploaded (81.4 %) than in downloaded traffic (67.9 %). These numbers indicate that a large fraction of data, potentially containing sensitive information, that leaves a smartphone is sent to cloud services. The higher cloud usage of 76.4 % for background (not directly triggered by users) compared to 67.3 % for foreground traffic (users interacting with the app) likely corresponds to synchronization tasks, e.g., updates of apps, typically happening in the background. We do not observe a large difference for the cloud usage of traffic sent over cellular compared to wifi networks. Furthermore, we observe that cloud usage is more prevalent for encrypted (76.0 %) than for plaintext traffic (55.8 %).

Most Prevalent Cloud Services. Given the overall high fraction of cloud traffic, we take a closer look at the *individual* cloud services that cause cloud traffic. In Table 2a, we list the 20 cloud services with the highest fraction of cloud traffic across all devices. We witness that several providers receive a large portion of traffic generated on mobile devices of our volunteers. Most notably, Google accounts for 34.7 % of traffic and is accessed from more than half of all apps. While Amazon accounts for significantly less traffic, Amazon is contacted by nearly two-thirds of all apps. These numbers highlight that few cloud services have a high market penetration, both in terms of traffic and numbers of apps. This distribution is especially problematic considering the imminent privacy risks resulting from a centralized cloud landscape (cf. Section 2.1).

Individual Perspective on Cloud Entanglement. To showcase that cloud entanglement has an individual nature, we evaluate how users' selection of and interaction with apps influences their cloud exposure. To this end, we study the *per-device cloud traffic for the 20 most installed apps* on our volunteers' devices in Figure 7. We exclude system apps, such as keyboards or contact synchronization. For each combination of device and app, we provide the fraction of cloud traffic ("–" denotes that an app did not produce *any* traffic on this device, likely because it was not installed). Comparing the apps of different devices, we notice that Devices 20, 23, and 25 use little to none of the 20 most popular apps. When looking at the apps used on these devices in more detail, we observe that these devices lack (the full stack of) Google apps, e.g., because of using custom ROMs. For these devices, we directly witness a lower fraction of cloud usage. However, Device 25 is a notable exception which seems to be running Amazon's version of Android, leading to a cloud usage comparable to those of devices with installed Google services. When looking at the cloud traffic for the same app *across* different devices, we observe two classes of apps: The first class contains a large number of apps where the fraction of cloud traffic is the same across all devices. Among the 20 most used apps,

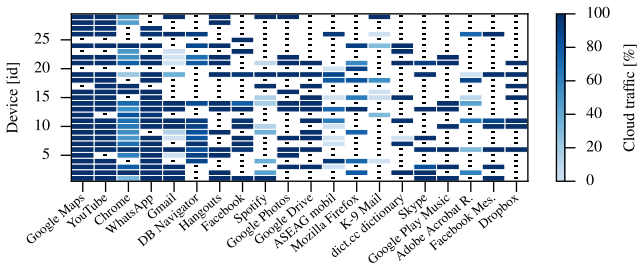

Figure 7: While most apps cause the same cloud entanglement across devices, certain apps' cloud traffic highly varies across different devices.

this class covers apps that nearly exclusively use cloud services. Nevertheless, we also found less common examples that produce no cloud traffic at all (e.g., the client for the self-hosted Nextcloud or banking apps). For the second class, we observe apps, e.g., web browsers and email clients, where the fraction of cloud traffic for the same app heavily deviates across devices. Hence, we discovered apps where cloud functionality is either built-in or not and others, where user behavior influences exposure to cloud services.

4.2 Cloud Entanglement of Mobile Websites

Web browsers are an important group of apps for which user behavior has a considerable influence on the level of cloud exposure. Hence, we analyze the cloud usage of the most popular websites to gain a deeper understanding of the cloud exposure they cause.

Measurement Setup. We mimic the mobile Chrome browser of a Google Nexus 5 smartphone and visit the mobile versions of the 5000 most popular websites (measured by Alexa [1]). We wait for each website to fully load and scroll to the bottom of the page to also trigger subsequent traffic resulting from embedded scripts.

Overall Cloud Usage. In Figure 8, we show the *number of cloud services per mobile websites*. We observe that 92.8 % of the popular mobile websites use cloud services and on average each of the websites exposes their visitors to 4.8 cloud services. In the extreme case, fetching the mobile version of `rollingstone.com` leads to connections with 16 different cloud services.

Additionally, we study the *resulting cloud traffic of mobile websites* in Figure 9. While 11.1 % of mobile websites are almost completely realized using cloud services (cloud traffic ≥ 99 %), we observe that the fraction of cloud traffic is nearly evenly distributed among websites. Hence, which websites a user frequently visits highly influence her *individual* exposure to cloud services.

Most Prevalent Cloud Services. We now identify the cloud services that are responsible for the most cloud usage. To this end,

we present the 20 *cloud services with the highest traffic from mobile websites* in Table 2b. In contrast to the most prevalent cloud services on mobile devices in general (cf. Section 4.1), we observe that Google has a significantly lower traffic share while CDNs play a more important role. Even though most cloud services do not account for large fractions of traffic generated by mobile websites, they are embedded in a large number of websites (e.g., AppNexus accounts for only 0.3 % of traffic but is embedded by 33.0 % of websites). Most notably, Google and Amazon are present on 84.5 % respectively 76.8 % of mobile websites. This most likely results from small scripts, e.g., for Google Analytics, that are embedded in a large number of mobile websites. As a result, these services have the potential to create detailed tracking profiles of users [52].

4.3 Cloud Entanglement of Popular Apps

So far, we have concentrated on studying cloud exposure of app usage. However, to thoroughly compare the cloud exposure caused by different apps and reveal the influence of differing locations on cloud usage, we now test apps under comparable conditions at *large scales*. We analyze the 500 most downloaded free apps in Google Play [33] for the five countries with the highest download numbers (Brazil, India, Mexico, Russia, and USA [3]).

Measurement Setup. We run our measurements on real hardware to create a realistic environment and prevent apps from changing their behavior due to detected virtualization [45]. To this end, we connect five Nexus 7 (Model 2013) devices running Android 6.0.1 each to a dedicated wireless router. Each router operates a VPN connection to a server in one of the five countries under study, similar to the setup proposed by MATAdOR [54]. However, we use commercial VPN endpoints instead of PlanetLab nodes. To account for the effect of different VPN speeds, we fix network bandwidth to 2 Mbit/s. We execute each app for 1 minute and provide random user input using Android's Application Exerciser Monkey [2] (communication with cloud services can be based on user input). We repeat our measurements in parallel for all five countries on 10 different days. In total, we study 1475 different apps (one app can be among the most popular apps in different countries).

Overall Cloud Usage. In Figure 10, we show the *number of utilized cloud services per app* for the different countries (across all 10 days). Notably, 90.0 % (India) to 94.8 % (USA) of the studied apps connect to at least one cloud service. On average, each app establishes a connection to 4.3 cloud services (3.8 in India to 4.9 in the USA). Each contacted cloud service constitutes a potential privacy risk (cf. Section 2.2). The app with the highest number of contacted services, `com.fingersoft.hillclimb`, a game with 7.9 million installs, uses 18 cloud services when launched in Russia.

Given these already high numbers, we now set out to quantify the *fraction of traffic flowing to cloud services*. For each of the five countries, Figure 11 contains the average fraction of cloud traffic for upload, download, and total traffic over all apps. The total fraction of cloud traffic ranges from 70.4 % in Russia to 80.3 % in the USA, which is in the order of those numbers we observed for foreground and cellular traffic on real devices in the wild (cf. Section 4.1). Notably, here we observe a higher fraction of cloud traffic for downloads compared to apps on real devices, likely because a large number of free apps download advertisements from cloud servers. These

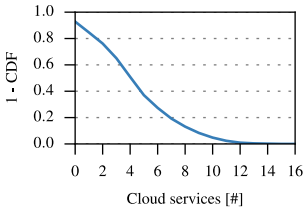

Figure 8: Number of cloud services used by websites.

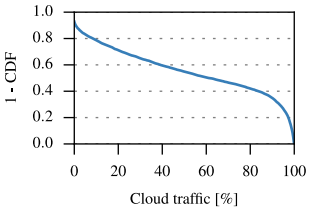

Figure 9: Cloud traffic produced by mobile websites.

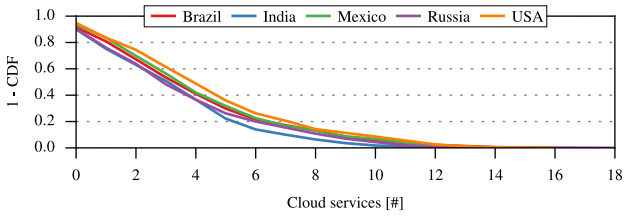

Figure 10: On average, each of the most popular apps uses 4.3 cloud services. Apps in the USA contact more cloud services, while apps in India and Russia use less cloud services.

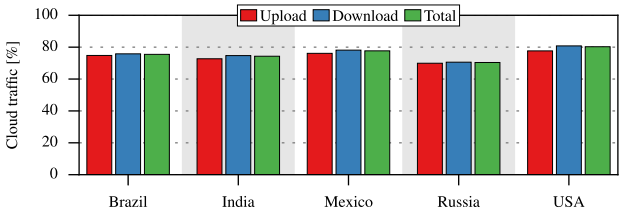

Figure 11: Traffic resulting from the most popular apps has a cloud exposure between 70 % and 80 %.

numbers highlight that our measurement setup is well suited to study the behavior of apps during their interactive usage.

Most Prevalent Cloud Services. Given the frequent usage of cloud services by the 500 most popular apps per country, we now identify the *most used cloud services* to understand which individual services are particularly responsible for this cloud exposure. To this end, Table 2c contains the 20 cloud services with the highest traffic across the 500 most popular apps in all five countries. Furthermore, we list for each cloud service the fraction of apps that established at least one connection to this service. We observe that the landscape of mobile cloud services is indeed highly centralized, with Google, Amazon, and Akamai each accounting for more than 10 % of an app's network traffic on average. Additionally, four cloud services (Google, Amazon, Akamai, and Facebook) are utilized by more than 50 % of the studied apps, significantly increasing the likelihood that users are exposed to these services. When studying these numbers, it is important to keep in mind that one network packet can belong to more than one cloud service when services are realized on top of each other. This situation occurs for, e.g., SoundCloud, which partly utilizes Amazon EC2 according to our findings.

Given the deviation in overall cloud usage between different countries identified in Figure 10, we now focus on what causes this effect by identifying the *most used cloud services in each country* in Figure 12. While overall we observe a similar trend across the five countries, notable differences exist: Verizon (2.7 % to 6.3 %) and

Unity (2.9 % to 4.8 %) are among the five most used services in only three of the countries. Furthermore, Facebook (4.0 %) is not among the five most used services in Russia. Finally, while Google accounts for the highest cloud usage in the other countries, Amazon (24.4 %) accounts for more traffic than Google (21.4 %) in the USA. Hence, the most popular apps in different countries lead to a different cloud exposure and thus different privacy risks.

Influence of Location. To answer the question whether differences in cloud usage result from different apps used in the five countries or if cloud usage indeed differs based on users' location, we study the influence of location on cloud usage by testing *identical* apps for the five countries. We tested the 73 apps that are among the 500 most popular apps in *all* of our five countries and synchronized measurements across countries to rule out dependencies on time factors. Again, we ran the experiment on 10 different days.

We first study the cloud usage of the 73 apps by comparing the resulting fraction of cloud traffic for the five *cloud services with the highest traffic in each country* in Figure 13. While we observe an overall similar pattern of utilizing cloud services across all countries, we still derive differences between the individual countries: First, India (16.5 %) and Russia (15.3 %) show more traffic for Akamai than the other countries (10.0 % to 12.2 %). Second, Microsoft is among the five cloud services with the highest amount of traffic in India, compared to Verizon for the other countries. Hence, the exposure of users to different cloud services does not only depend on the used apps, but also on the (network) location where apps are used.

To further study the influence of location, we rely on information on the location of data centers for some, especially larger cloud services (cf. Section 3.2). We use this information to investigate whether the (network) location of a mobile device has an influence on the *geographically distribution of contacted cloud services*. More specifically, we show the fraction of traffic that we were able to assign to a geographic location (aggregated based on continents) in Figure 14. While the majority of traffic for which we could derive a location flows to North America (8.4 % to 9.7 % of overall traffic), we can observe that apps tend to connect to geographically near cloud data centers. This observation is illustrated by an increased fraction of cloud traffic to South America for Brazil, to Asia for India, and to Europe for Russia. Such information on the location of data centers used by an app allows users to execute their right to privacy, e.g., when deciding between two apps with similar functionality.

5 CONCLUSION

Apps on smartphones have access to a growing amount of sensitive information. As apps nowadays increasingly realize their functionality through cloud services, they potentially expose users' private information to a variety of third parties. Even worse, users are often unaware of this erosion of their privacy. Starting from these observations, we provide a detailed analysis of the mobile cloud landscape which reveals and concretizes significant privacy risks.

Our problem analysis makes evident that users need to regain control over their privacy. As a first step towards this goal, we have to raise their awareness about their individual exposure to cloud services and the implied privacy risks. Hence, we present CloudAnalyzer which provides users with detailed statistics of their *individual* cloud exposure caused by their smartphone apps.

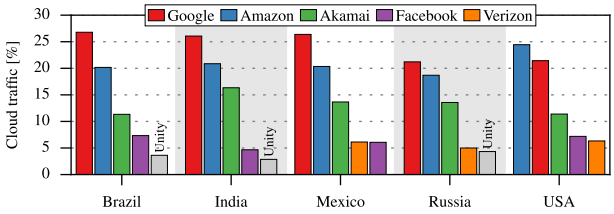

Figure 12: Despite a similar trend, we observe notable differences in cloud traffic of popular apps of different countries.

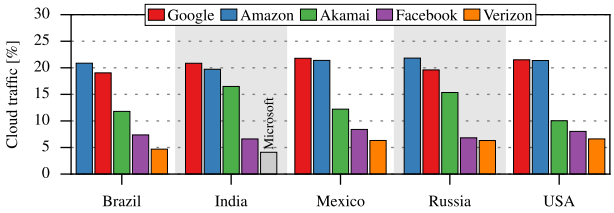

Figure 13: Identical apps utilize cloud services differently when operated in different countries.

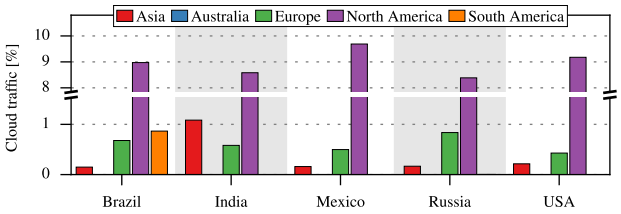

Figure 14: Identical apps partly use data centers on different continents when operated in different countries.

CloudAnalyzer *locally* monitors network traffic of apps and detects communication with 55 cloud services that represent the mobile cloud computing landscape. As a consequence, we not only reveal the hidden exposure to cloud services caused by smartphone apps, but also untangle complex and non-transparent data flows caused by indirection and subcontracting between cloud providers.

To show the applicability of CloudAnalyzer, we deploy CloudAnalyzer to 29 devices to reveal the cloud exposure of actual users over the course of 19 days. Additionally, we analyze the cloud entanglement caused by the 5000 most used mobile websites as well as the 500 most popular apps in five different countries. Our results confirm that smartphone users are indeed exposed to cloud services: About 90 % of all studied apps contact at least one cloud service and 36 % of apps used by volunteers exclusively communicate with cloud services. One volunteer even reported on uninstalling an app due to excessive cloud usage uncovered by CloudAnalyzer.

To conclude, CloudAnalyzer empowers users to critically review their individual exposure to cloud services. With a clear view of their exposure and risk, users are encouraged to adapt their app usage behavior or to take more informed decisions when choosing between apps with similar functionality. In a second step, CloudAnalyzer can be used as a foundation to enable users to compare their personal app-induced cloud exposure to that of their peers to discover potential privacy risks of deviating from normal usage behavior [37, 67]. Notably, CloudAnalyzer also constitutes a valuable tool for researchers interested in understanding the characteristics

of users' exposure to cloud services. Similarly, CloudAnalyzer is beneficial for app developers to ensure compliance with data protection regulations. Based on the information provided by CloudAnalyzer, developers can ensure that their app (and included third-party libraries) does not inadvertently contact cloud services, especially if they are located in countries with weaker data protection regulations [20]. CloudAnalyzer provides developers with the necessary means to monitor their apps for (unintended) cloud usage.

Acknowledgments. The authors would like to thank Torsten Zimmermann for providing the measurements of mobile websites and all participants of the user study. This work was funded by the German Federal Ministry of Education and Research (BMBF) under project funding reference no. 16KIS0351 (TRINICS). The responsibility for the content of this publication lies with the authors.

REFERENCES

[1] Alexa. 2016. Actionable Analytics for the Web. http://www.alexa.com/. (2016).
[2] Android. 2017. UI/Application Exerciser Monkey – Android Studio. https://developer.android.com/studio/test/monkey.html. (2017).
[3] App Annie. 2015. App Annie Index^{TM}: Market Q2 2015. (2015).
[4] AppBrain. 2017. Ad networks – Android library statistics. https://www.appbrain.com/stats/libraries/ad, February 15, 2017. (2017).
[5] AppBrain. 2017. Android analytics libraries. https://www.appbrain.com/stats/libraries/tag/analytics/android-analytics-libraries, February 15, 2017. (2017).
[6] AppBrain. 2017. Android crash reporting libraries. https://www.appbrain.com/stats/libraries/tag/crash-reporting/android-crash-reporting-libraries, February 15, 2017. (2017).
[7] AppBrain. 2017. Social SDKs – Android library statistics. https://www.appbrain.com/stats/libraries/social, February 15, 2017. (2017).
[8] AppBrain. 2017. Video ads. https://www.appbrain.com/stats/libraries/tag/video-ads/video-ads, February 15, 2017. (2017).
[9] S. Arzt et al. 2014. FlowDroid: Precise Context, Flow, Field, Object-sensitive and Lifecycle-aware Taint Analysis for Android Apps. In *ACM PLDI 2014*.
[10] N. Asokan et al. 2014. Mobile Platform Security. *Synthesis Lectures on Information Security, Privacy, & Trust* 4, 3 (2014).
[11] I. Bermudez et al. 2013. Exploring the Cloud from Passive Measurements: the Amazon AWS Case. In *IEEE INFOCOM 2013*.
[12] I. N. Bermudez et al. 2012. DNS to the Rescue: Discerning Content and Services in a Tangled Web. In *ACM IMC 2012*.
[13] R. Bhoraskar et al. 2014. Brahmastra: Driving Apps to Test the Security of Third-Party Components. In *USENIX Security 2014*.
[14] T. Book et al. 2013. Longitudinal Analysis of Android Ad Library Permissions. In *MoST 2013*.
[15] J. Brookman et al. 2017. Cross-Device Tracking: Measurement and Disclosures. In *PETS 2017*.
[16] Canasys. 2017. Cloud infrastructure market up 49%, intensifying global data center competition. Press release 2017/1630. (2017).
[17] A. Cavoukian. 2011. Privacy by Design – The 7 Foundational Principles. Information and Privacy Commissioner of Ontario. (2011).
[18] T. Chen et al. 2014. Information leakage through mobile analytics services. In *ACM HotMobile 2014*.
[19] Datanyze. 2017. CDN market share in the Alexa top 1M. https://www.datanyze.com/market-share/cdn/Alexa%20top%201M, February 17, 2017. (2017).
[20] P. De Filippi and S. McCarthy. 2012. Cloud Computing: Centralization and Data Sovereignty. *Europ. J. Law Technol.* 3, 2 (2012).
[21] T. Dierks and E. Rescorla. 2008. The Transport Layer Security (TLS) Protocol Version 1.2. IETF RFC 5246. (2008).
[22] D. Dittrich and E. Kenneally. 2012. *The Menlo Report: Ethical Principles Guiding Information and Communication Technology Research.* Technical Report. U.S. Department of Homeland Security.
[23] I. Drago et al. 2012. Inside Dropbox: Understanding Personal Cloud Storage Services. In *ACM IMC 2012*.
[24] D. Eastlake 3rd. 2011. Transport Layer Security (TLS) Extensions: Extension Definitions. IETF RFC 6066. (2011).
[25] N. Elenkov. 2014. *Android Security Internals: An In-depth Guide to Android's Security Architecture* (1st ed.). No Starch Press.
[26] W. Enck et al. 2010. TaintDroid: An Information-Flow Tracking System for Realtime Privacy Monitoring on Smartphones. In *USENIX OSDI 2010*.
[27] B. Fabian et al. 2015. Topological analysis of cloud service connectivity. *Computers & Industrial Engineering* 88 (2015).
[28] D. Ferreira et al. 2015. Securacy: An Empirical Investigation of Android Applications' Network Usage, Privacy and Security. In *ACM WiSec 2015*.
[29] J. Finkle and A. G. Tharakan. 2016. Yahoo says one billion accounts exposed in newly discovered security breach. Reuters, http://www.reuters.com/article/us-yahoo-cyber-idUSKBN1432WZ, December 15, 2016. (2016).
[30] B. Gellman. 2013. Edward Snowden, after months of NSA revelations, says his mission's accomplished. *The Washington Post* (2013). Dec. 24, 2013.
[31] C. Gibler et al. 2012. AndroidLeaks: Automatically Detecting Potential Privacy Leaks in Android Applications on a Large Scale. In *TRUST 2012*.
[32] P. Gilbert et al. 2011. Vision: Automated Security Validation of Mobile Apps at App Markets. In *ACM MCS 2011*.
[33] Google. 2016. Top Free in Android Apps – Android Apps on Google Play. https://play.google.com/store/apps/collection/topselling_free. (2016).
[34] K. He et al. 2013. Next Stop, the Cloud: Understanding Modern Web Service Deployment in EC2 and Azure. In *ACM IMC 2013*.
[35] M. Henze et al. 2016. A Comprehensive Approach to Privacy in the Cloud-based Internet of Things. *Future Generation Computer Systems* 56 (2016).
[36] M. Henze et al. 2016. Moving Privacy-Sensitive Services from Public Clouds to Decentralized Private Clouds. In *IEEE IC2E Workshop CLaw 2016*.
[37] M. Henze et al. 2017. Privacy-preserving Comparison of Cloud Exposure Induced by Mobile Apps. In *MobiQuitous 2017*.
[38] M. Henze et al. 2016. Towards Transparent Information on Individual Cloud Service Usage. In *IEEE CloudCom 2016*.
[39] M. Henze et al. 2017. Veiled in Clouds? Assessing the Prevalence of Cloud Computing in the Email Landscape. In *IEEE/IFIP TMA 2017*.
[40] I. Ion et al. 2011. Home is Safer Than the Cloud!: Privacy Concerns for Consumer Cloud Storage. In *SOUPS 2011*.
[41] A. Langley and W.-T. Chang. 2016. QUIC Crypto. Google, Revision 20161206. (2016).
[42] A. Le et al. 2015. AntMonitor: A System for Monitoring from Mobile Devices. In *ACM SIGCOMM Workshop C2B(1)D 2015*.
[43] D. Malandrino et al. 2013. Privacy Awareness about Information Leakage: Who knows what about me?. In *ACM WPES 2013*.
[44] P. Mockapetris. 1987. Domain names – concepts and facilities. IETF RFC 1034. (1987).
[45] S. Mutti et al. 2015. BareDroid: Large-Scale Analysis of Android Apps on Real Devices. In *ACSAC 2015*.
[46] J. Pennekamp et al. 2017. A Survey on the Evolution of Privacy Enforcement on Smartphones and the Road Ahead. *Pervasive and Mobile Computing* (2017).
[47] I. Poese et al. 2011. IP Geolocation Databases: Unreliable? *SIGCOMM Comput. Commun. Rev.* 41, 2 (2011).
[48] J. Postel. 1981. Internet Protocol. IETF RFC 791. (1981).
[49] E. Pujol et al. 2015. Annoyed Users: Ads and Ad-Block Usage in the Wild. In *ACM IMC 2015*.
[50] A. Razaghpanah et al. 2015. Haystack: In Situ Mobile Traffic Analysis in User Space. *arXiv preprint arXiv:1510.01419* (2015).
[51] J. Ren et al. 2016. ReCon: Revealing and Controlling PII Leaks in Mobile Network Traffic. In *ACM MobiSys 2016*.
[52] F. Roesner et al. 2012. Detecting and Defending Against Third-Party Tracking on the Web. In *USENIX NSDI 2012*.
[53] Sandvine. 2016. 2016 Global Internet Phenomena – Latin America & North America. (2016).
[54] Q. Scheitle et al. 2016. Analyzing Locality of Mobile Messaging Traffic using the MATAdOR Framework. In *PAM 2016*.
[55] S. Shekhar et al. 2012. AdSplit: Separating Smartphone Advertising from Applications. In *USENIX Security 2012*.
[56] Skyhigh. 2016. Cloud Adoption & Risk Report Q4 2016. (2016).
[57] Y. Song and U. Hengartner. 2015. PrivacyGuard: A VPN-based Platform to Detect Information Leakage on Android Devices. In *ACM CCS Workshop SPSM 2015*.
[58] C. Spensky et al. 2016. SoK: Privacy on Mobile Devices – It's Complicated. In *PETS 2016*.
[59] M. Sun et al. 2016. TaintART: A Practical Multi-level Information-Flow Tracking System for Android RunTime. In *ACM CCS 2016*.
[60] H. Takabi et al. 2010. Security and Privacy Challenges in Cloud Computing Environments. *IEEE Security & Privacy* 8, 6 (2010).
[61] M. Theoharidou et al. 2013. Privacy Risk, Security, Accountability in the Cloud. In *IEEE CloudCom 2013*.
[62] N. Vallina-Rodriguez et al. 2012. Breaking for Commercials: Characterizing Mobile Advertising. In *ACM IMC 2012*.
[63] N. Vallina-Rodriguez et al. 2016. Tracking the Trackers: Towards Understanding the Mobile Advertising and Tracking Ecosystem. In *DAT Workshop 2016*.
[64] X. Wei et al. 2012. ProfileDroid: Multi-layer Profiling of Android Applications. In *ACM Mobicom 2012*.
[65] Q. Xu et al. 2011. Identifying Diverse Usage Behaviors of Smartphone Apps. In *ACM IMC 2011*.
[66] Z. Yang et al. 2013. AppIntent: Analyzing Sensitive Data Transmission in Android for Privacy Leakage Detection. In *ACM CCS 2013*.
[67] J. H. Ziegeldorf et al. 2015. Comparison-based Privacy: Nudging Privacy in Social Media (Position Paper). In *DPM 2015*.