Analyzing Locality of Mobile Messaging Traffic using the MATAdOR Framework

Quirin Scheitle, Matthias Wachs, Johannes Zirngibl, and Georg Carle

Technical University of Munich (TUM)
Department of Informatics
Chair for Networking Services and Architectures
Email: {scheitle, wachs, carle}@net.in.tum.de,
{zirngibl}@in.tum.de

Abstract. Mobile messaging services have gained a large share in global telecommunications. Unlike conventional services like phone calls, text messages or email, they do not feature a standardized environment enabling a federated and potentially local service architecture. We present an extensive and large-scale analysis of communication patterns for four popular mobile messaging services between 28 countries and analyze the locality of communication and the resulting impact on user privacy. We show that server architectures for mobile messaging services are highly centralized in single countries. This forces messages to drastically deviate from a direct communication path, enabling hosting and transfer countries to potentially intercept and censor traffic. To conduct this work, we developed a measurement framework to analyze traffic of such mobile messaging services. It allows to carry out automated experiments with mobile messaging applications, is transparent to those applications and does not require any modifications to the applications.

Keywords: Mobile Messaging·Security·WhatsApp·WeChat·Threema·TextSecure

1 Introduction

Mobile messaging services like WeChat or WhatsApp see a steady increase in both active users and messages sent, with a particular success in emerging markets like China, Brazil or Malaysia [18, 30]. Some researchers predict a shift in communication paradigms with mobile messaging services eradicating classical forms of electronic communication like email or text messages. As an example, the number of text messages sent in Germany shrunk by 62% from 2012 to 2014 [6], after it had been growing exponentially for over a decade.

Mobile messaging services and their design strongly differ from classic Internet communication services: established means of communication—like email, internet telephony or instant messaging—often rely on federated or decentralized architectures, with operators providing services to their customers and from within their domain.

Mobile messaging services tend to abandon established principles of openness and federation: messaging services are often realized in a closed, non-federated, cloud-centric environment built upon proprietary communication and security protocols neither standardized nor disclosed to the public.

This paradigm shift puts at risk the user's freedom and access to secure, confidential and privacy-preserving communication. With such services, the user—relating to her social network through such applications—strongly depends on the service provider to not modify or restrict the service. The user's privacy also depends on the legislation the operating company is subject to: governments are often interested in controlling Internet services [14, 31] and accessing messages [9] as well as metadata. The matters of security and privacy move along the same lines and generally involve a full trust into a closed system, a misleading assumption as we saw with WhatApp's announced *end-to-end-encryption*, which is supported on Android, but not Apple devices [1], without giving feedback on encryption status to the user. First attempts to analyze the security properties of mobile messaging services have for example been made by the EFF with its *Secure Messaging Scorecard* [4].

In this work, we analyze the implications of mobile messaging services on the users and their privacy. Similar to the discussion about a "nation-centric Internet" [32], we set out to understand the communication behavior and patterns of mobile messaging services by analyzing how *local* messaging traffic is from a geographic and legal point of view. We analyze how messaging traffic is routed through the Internet and which countries could therefore access this traffic. We compare this path with the direct communication path which could have been taken between communication partners to quantify the impact of mobile messaging services. For this analysis, we developed an analysis platform and testbed for mobile applications, called MATAdOR (Mobile Application Traffic Analysis plattfORm). We use MATAdOR to exchange messages between a large number of communication partners distributed over the world using different mobile messaging applications and automatically extract information about the network path the messages used.

Highlights of our results include: (a) Mobile messaging services largely distort traffic locality. (b) For Asian users (except Israel), Threema traffic is routed through the U.S. and hence 5 Eyes accessible. (c) Even South American internal traffic is routed through North America. (d) Europe-based users can reduce 5 Eyes access by routing messages through Threema's Switzerland servers. (e) Except WeChat, mobile messaging services showed globally uniform behavior.

2 Related Work

Several projects worked on analyzing the behavior and communication patterns of mobile messaging services and the challenges arising when conducting automated experiments with mobile devices and applications.

Fiadino et al. [7] investigated characteristics of WhatsApp communication based on a set of mobile network trace data from February 2014. In this set, they identified every DNS request to WhatsApp and resolved them in a distributed way through the RIPE Atlas service. They found the corresponding address to be exclusively located in the U.S. and focussed further on Quality of Experience analysis. Huang et al. [10] did similar work on WeChat, using network traces as well as controlled experiments. For the latter, they connected phones through WLAN, but relied on heavy manual work for message sending and traffic analysis. They do not mention a capability to proxy traffic

out through remote nodes. On the collected data, they heavily focus on dissecting the protocol and architecture. Mueller et al. [16] researched security for a wide set of mobile messaging services and found many weaknesses, e.g. on the authentication bootstrapping process. They used a testbed similar to MATAdOR, but had to explicitly configure the mobile device's proxy settings. Frosch et al. [8] provided a detailed protocol analysis for TextSecure based on its source code. The life cycle of network experiments, automated experimentation and testbed management is in the focus of several related projects. The OpenLab Project[1] focuses on improving network experimentation for future distributed and federated testbeds and to provide tools to researchers. Various tools for supporting testbed setup and experimentation exist [19], but many are outdated or unavailable. None of these tools support experimentation with mobile devices or geographic diversion of network traffic.

[33] provides an extensive list of commercial platforms aiming to integrate functional mobile application testing in the software development cycle. Many platforms support the use of real devices and some even provide testing over mobile carrier networks to ensure functionality. Many solutions are only provided as a paid service.

3 Analyzing Communication of Mobile Messaging Applications

In order to analyze the impact of mobile messaging services on traffic locality, our approach is to compare the *network path*, defined as direct network path between communication partners obtained with forward path measurements, and the *application path*, defined as the forward path measurements from both partners to the mobile messaging service's backend infrastructure.

We use the MATAdOR testbed to send a large number of messages using different mobile messaging services between communication partners distributed all over the globe. To do so, we use MATAdOR equipped with two mobile devices and the mobile messaging application under test. MATAdOR tunnels the application traffic to PlanetLab nodes as depicted in Figure 1. We intercept the applications' communication and extract the communication endpoints. Based on this information, we conduct forward path measurements to the mobile messaging service's backend servers to obtain the *application path* and between the nodes to obtain the *network path*.

We map the hops in both application path and network path to countries and analyze which jurisdictions and political frameworks the traffic traverses on its way between communication partners. As a result, we can give a qualified analysis how much the application path and the network path differ and if traffic is confined to a geographic region when both partners are located in this region.

3.1 Experimental Setup

The experimental setup of MATAdOR consists of a dedicated controller node, two WLAN routers, two Android mobile phones and the PlanetLab proxy nodes as depicted in Figure 1. The controller node orchestrates the overall experimentation process, configures the WLAN routers, configures the Android devices and instruments them to

[1] http://www.ict-openlab.eu

send messages. Device instrumentation is realized using the Android Debug Bridge to configure network connections, start applications, and issue input events to the devices to automate message sending. The routers spawn two wireless networks and establish tunnels to the respective PlanetLab nodes. The router's task is to route, intercept and modify traffic as well as to automatically parse network traces and start path measurements to targets. To leverage PlanetLab for this experiment, we use a tool currently under development at our chair. This tool is able to transparently proxy traffic over PlanetLab nodes. It is currently in beta status and pending public release.

Fig. 1. Overall experiment design.

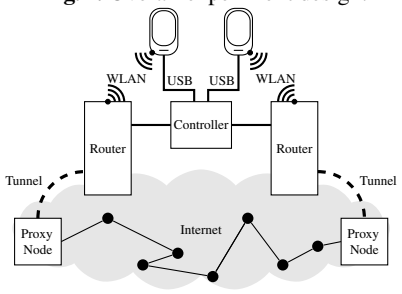
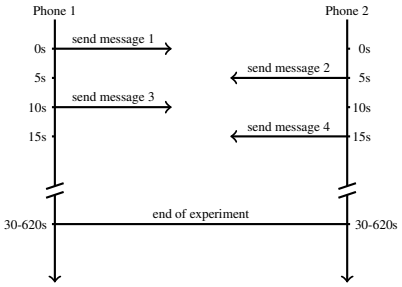


Fig. 2. Overview of messaging timers



Mobile Phone Configuration To run the mobile messaging applications, we use two off-the-shelf, rooted Motorola Moto-E (2nd generation) smartphones running vanilla Android 5.0.2. For each device we created an individual Google Play account. To allow control through the Android Debug Bridge (ADB), devices are connected to the controller using USB. We use XPrivacy[2] to set the phone's location information according to the location of the specific PlanetLab node and `iptables` to restrict network communication to the specific mobile messaging application (based on its UID) under test. To prevent geolocation based on mobile network information, the phones were set to airplane mode with only WLAN enabled.

Router Configuration Two GNU/Linux PCs, configured to act as WLAN access points, provide two dedicated WPA2-protected wireless networks, one to each mobile phone. Through DHCP, they provide a RFC 1918 private address and the PlanetLab node's DNS server to the phones. The routers use `tcpdump` to intercept traffic and `scapy` to automatically process network traces.

Measurement Orchestration The measurements to conduct are defined as *experiments*. Within each experiment MATAdOR executes the respective set of commands. This involves setting up remote tunnels to two PlanetLab nodes, configuring the network settings on the routers according to the experiment, starting interception and manipulation software on the routers, configure the phone to use the wireless network, setting XPrivacy and firewall settings on the phone, capturing the phone's screen for later inspection, stepping through the experiment on the phones with ADB automation, parsing the network trace data automatically, and executing path measurements to all IP addresses found in the network trace.

[2] http://repo.xposed.info/module/biz.bokhorst.xprivacy

Experiment Parametrization To permit experimentation with different applications, all required experimental parameters are controlled through application-specific configuration files. This includes timers between the different steps of the experiment, blacklists of hosts not to include in path measurements (e.g. NTP or DNS servers), the text to send in the messages and how many messages to send with the application. Such messaging timers, depicted in Figure 2, are controlled through these configuration files.

Experiment Monitoring and Error Handling While running experiments, we have learned that using unaltered applications on physical devices in this complex setup is prone to errors. We therefore split the overall experiment into smaller junks to be able to reproduce missing or failing measurements. To be able to detect and analyze failures, the screen of the mobile devices is captured for each measurement.

Benefits Using the MATAdOR Testbed Our approach minimizes effort and cost using common available off-the-shelf hardware. Since MATAdOR does not rely on device or run time emulation, simulated network connections, adapted applications, or the devices being otherwise modified in an unusual way (e.g. setting an application or device proxy), the testbed environment is transparent to both the phone and apps and looks like a "normal" wireless network. All steps within the experiment life cycle have been automated. This provides the possibility to efficiently scale the number of applications and experiments. MATAdOR provides functionality to easily and automatically intercept all network traffic. It can also transparently redirect network traffic through hosts at remote locations, appearing to outsiders and the application itself as if the phone was located at that place. When proxying the phone's traffic through a remote location, the phone's location services are manipulated accordingly.

3.2 Methodology

The goal of our experiment is to collect information about the path that messages take on the Internet when two communication partners communicate with each other using a mobile messaging application. In addition, we want to learn about the regions and countries a message traverses on its way. To do so, we have to analyze the network path between both communication partners and the messaging service infrastructure.

In our experiment, we use a set of four carefully selected mobile messaging services and use their respective applications to exchange messages between the two mobile phones in our testbed. In a single measurement, we use one specific mobile messaging application, connect to the mobile messaging service on both phones and exchange messages between both devices. By doing so, we can extract the communication endpoints for the mobile messaging service from the network traffic. We can then perform path measurements to these communication endpoints from both mobile phones to obtain the network path to the service provider infrastructure. To get a global view on communication, we tunnel traffic through 28 PlanetLab nodes. This way, we can learn the path messages take for example for a WhatsApp user in Australia communicating with a user in North America. In addition, we conduct direct path measurements between both respective PlanetLab nodes to obtain the direct network path.

For the path measurements, we use the standard `traceroute` tool provided with GNU/Linux. From the network traces, we extract the protocol (i.e. TCP or UDP) and port number (e.g. 443) the mobile messaging service uses and apply these settings to

measure the network path to the mobile messaging service infrastructure. To obtain the path between nodes, we use `traceroute` with TCP and a random high port.

Selection of Applications For this work, we carefully selected four different mobile messaging services based on different characteristics depicted in Table 1.

Based on their popularity, we picked WhatsApp and WeChat as the two mobile messaging services built for mobile chat. Due to its high rank in the EFF Scorecard with respect to security and privacy and being free software with its source code open to the public, we picked TextSecure as a third application for this experiment. We chose Threema for its promise of servers based in Switzerland and claim of strong privacy for the users. In addition, Threema is one of the few Europe-based providers. Since all of the previous solutions rely on a centralized client/server architecture, we select Bleep as a fifth candidate due to its decentralized peer-to-peer architecture. However, we could not enforce peer-to-peer behavior in our testbed and observed minute-long delays between messages. We concluded that peer-to-peer architectures require closer investigation including the use of NAT traversal techniques in our framework. For this reason, we excluded Bleep from the set of applications. We did not further pursue Firechat as it advertises peer-to-peer behavior only for local mesh networks.

Table 1. Properties of mobile messaging services and applications.

Application (Version)	Monthly active users[1] [22]	EFF Scorecard[2] Points [4]	Architecture	Server Distribution	Mobile First
WhatsApp (2.12.176)	800-900mn [12, 23] [27, p.23]	2	client-server	n/a	✓
WeChat (6.2.4)	400-600mn [27, p.22] [26, p.4]	n/a	client-server	n/a	✓
Facebook[3]	350-600mn [5], [27, p.22]	2	client-server	n/a	✗
Skype	300mn [15]	1	client-server	n/a	✗
QQ International	843mn [26, p.4]	2	client-server	n/a	✗
Viber	249mn [21]	1	client-server	n/a	✓
LINE	211mn [13]	n/a	client-server	n/a	✓
Kik	200mn[4] [25]	1	client-server	n/a	✓
Tango	70mn [24]	n/a	client-server	n/a	✓
KakaoTalk	48mn [2]	n/a	client-server	n/a	✓
Yahoo Messenger	n/a	1	client-server	n/a	✗
TextSecure (2.24.1)	>10mn[4] [17]	7	client-server	global	✓
Silent Text	n/a	7	client-server	n/a	✓
Telegram	30-50mn [27, p.22] [28]	4[5]	client-server	global	✓
Wickr	4mn[6] [20]	5	client-server	global	✓
Bleep (1.0.616)	n/a	n/a	P2P	n/a	✓
FireChat	n/a	n/a	mesh P2P	n/a	✓
Threema (2.41)	3mn[4] [29]	5	client-server	Switzerland	✓
SIMSme	1 mn[6]	n/a	client-server	Germany	✓

1: Around July 30, 2015, for exact date see app-specific source 2: EFF Secure Messaging Scorecard [4]
3: Stand-alone Facebook Messenger 4: Registered users 5: Score of 7 in secure chats 6: App Store Downloads

Node Selection To achieve a global view on messaging communication, we compiled a list of PlanetLab nodes providing a wide geographical distribution. The objective for this list was to cover as many regions and countries as possible. However, PlanetLab does not provide equal coverage in all regions and availability of nodes strongly differs across regions. When we conducted our experiment, PlanetLab featured nodes

in 49 countries, but we only found 28 countries with at least one stable and responsive node, providing good coverage for North America, Europe, Asia and Oceania. For South America only a single node in Argentina and Brazil was provided, for Africa no nodes could be accessed at all.

For our experiment, we therefore used 4 nodes in the Americas (North America: 2, South America: 2), 7 nodes in Asia (Eastern Asia: 4, South-Eastern Asia: 2, Western Asia: 1), 16 nodes in Europe (Eastern Europe: 3, Northern Europe: 5, Southern Europe: 4, Western Europe: 4) and 2 nodes in Oceania.

Limitations It is important to note that our path measurements only record a country as being part of a path if a hop from that country replies to path measurements. This can be biased by (a) nodes not answering those requests and (b) countries being passively traversed. Especially the latter is relevant, as intelligence services are known to also wiretap passively. For example, some measurements from Switzerland offer direct paths to Hong Kong or the U.S., but obviously more countries in between would have passive access to the cables in-between.

4 Postprocessing Experiment Results

Despite limiting application communication, the resulting network traces included some irrelevant flows. For this experiment, we solely want to evaluate traffic between the mobile messaging application and the mobile messaging service's backend. Therefore we had to classify network flows and assemble a black- and whitelist of network flows for exclusion or inclusion. Here, we went through several steps:

First, we limited background traffic by firewalling communication to only allow the specific mobile messaging application under test to access the network. Second, we conducted six measurements from America, Europe and Asia without the mobile messaging application running. This resulted in network traces containing "background noise" we could exclude after manual validation. Third, we manually inspected several dozens of traces per mobile messaging application to determine additional background traffic. The sources for this traffic were manually added to the filtering blacklist. Fourth, we separated authentication and other background traffic for every application from messaging traffic through temporal correlation with message timers.

For Threema, TextSecure and WhatsApp, we found all messaging servers to be resolved through DNS and to resolve uniformly across the globe, confirming the results of [7] for WhatsApp. We found WeChat to use both DNS requests and a custom-built DNS-over-HTTP protocol for name resolution, providing different name resolution when queried from within or from outside China. This DNS-over-HTTP uses a 30-minute timeout and therefore "contaminates" our name resolution cache, which we flush after every experiment, typically lasting five to ten minutes. We therefore built the whitelist for WeChat analysis through manual analysis. The resulting detailed DNS table can be found online[3].

In a last step we automatically processed all traces and classified all addresses into this black- or whitelist. We manually classified all remaining addresses.

[3] http://www.net.in.tum.de/pub/mobmes/dnstable.pdf

4.1 Mapping Path Measurements to Countries and Regions

To obtain the countries the traffic traverses, both the application path and the network path were processed to provide a geolocation of the IP addresses. With some manual corrections, we found the ip2location[4] country database to provide the most accurate results. To not overly rely on that database, we manually validated the mappings in at least one trace per target subnet and source country. With respect to known inaccuracies of both reverse DNS labels and geolocation databases, as described in [11, 34], we paid special attention to round-trip times found in forward path measurements.

To analyze locality with respect to a specific geographic region, we used the United Nations geoscheme[5] to assign countries to regions and subregions. This scheme relies on 5 regions (Africa, Americas, Asia, Europe, Oceania) which are further divided into geographic subregions (e.g. for the Americas: Latin America and the Caribbean, Central America, South America, and Northern America).

4.2 Mapping Countries to Interest Groups

In addition to geographic locality, we analyzed the possibility of several jurisdictions and similar entities to access the network traffic. In this analysis, we defined several *interest groups* and checked for the different mobile messaging services if these interest groups can access the traffic. For this analysis we defined the following interest groups:

- *5 Eyes* consisting of: Great Britain, United States, New Zealand, Canada
- *European Union* consisting of: Austria, Belgium, Bulgaria, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, Netherlands, Poland, Portugal, Romania, Slovakia, Slovenia, Spain, Sweden, United Kingdom
- *Arab League* consisting of: Algeria, Bahrain, Comoros, Djibouti, Egypt, Iraq, Jordan, Kuwait, Lebanon, Libya, Mauritania, Morocco, Oman, Palestine, Qatar, Saudi Arabia, Somalia, Sudan
- *Russia* with the only member Russia
- *China* with the only member China

5 Results

With our experiments, running from Sep 30 2015 to Oct 12 2015, we conducted 406 measurements between the 28 PlanetLab nodes using the 4 selected mobile messaging services, resulting in 1624 measurements in total.

Table 2 shows the path comparisons between application path and network path. The first columns evaluate the direct measurements between nodes and show how many % of measurements failed to stay within the region. We found that all traffic from Israel to other Asian countries is being routed through Europe and the U.S. As we use seven nodes in Asia, six measurements from Israel fail to remain within region. Also, with

[4] http://www.ip2location.com

[5] http://millenniumindicators.un.org/unsd/methods/m49/m49regin.htm

Table 2. Mobile messaging services in almost all cases direct traffic out of region.

		Traffic leaving region					
		Network Path		Application Path			
Region	# Measurements	#	%	TextSecure	Threema	WeChat	WhatsApp
Europe	120	0	0%	100%	0%	100%	100%
Oceania	3	0	0%	100%	100%	100%	100%
Asia	28	6	21%	100%	100%	50%	100%
Americas	10	0	0%	0%	100%	100%	0%
South America	3	1	33%	100%	100%	100%	100%
Northern America	3	0	0%	0%	100%	100%	0%

Legend: > Network Path

two nodes in South America, the measurement between those two nodes leaves South America for routing through North America. As the two in-country measurements stay in the region, the 33% understate the effect, caused by the low number of nodes. As a result we highlight that only Europe and North America feature at least one messenger that keeps traffic local. Asia traffic for WeChat does not remain local because of Israel's aforementioned routing and also because of traffic from Singapore and Thailand being routed to the Chinese WeChat servers through U.S. IXPs.

Table 3 shows how measurements from a specific region were subject to various interest groups, both for the network path and for the specific application path:

Table 3. Mobile messaging services in most cases increase traffic accessibility for interest groups.

			Accessible for Interest Group									
	Interest		Network Path		TextSecure		Threema		WeChat		WhatsApp	
Region	Group	#Total	#	%	#	%	#	%	#	%	#	%
Europe	5 Eyes	120	86	72%	120	100%	68	57%	119	99%	120	100%
Europe	EU	120	118	98%	119	99%	119	99%	120	100%	120	100%
Europe	China	120	0	0%	0	0%	0	0%	120	100%	0	0%
Oceania	5 Eyes	3	3	100%	3	100%	3	100%	3	100%	3	100%
Oceania	EU	3	0	0%	0	0%	3	100%	0	0%	0	0%
Oceania	China	3	0	0%	0	0%	0	0%	3	100%	0	0%
Asia	5 Eyes	28	6	21%	28	100%	21	75%	14	50%	28	100%
Asia	EU	28	6	21%	7	25%	18	64%	7	25%	7	25%
Asia	China	28	10	36%	7	25%	7	25%	28	100%	7	25%
South America	5 Eyes	3	1	33%	3	100%	3	100%	3	100%	3	100%
South America	EU	3	0	0%	0	0%	2	67%	0	0%	0	0%
South America	China	3	0	0%	0	0%	0	0%	3	100%	0	0%
North America	5 Eyes	3	3	100%	3	100%	3	100%	3	100%	3	100%
North America	EU	3	0	0%	0	0%	2	67%	0	0%	0	0%
North America	China	3	0	0%	0	0%	0	0%	3	100%	0	0%

Legend: < Network Path > Network Path

Europe to Europe: 72% of network path measurements within Europe were accessible to 5 Eyes (by routing through UK). 98% of measurements were accessible to the European Union, with only measurements internal to Switzerland and Norway not being accessible. For application paths, Threema reduces the 5 Eyes access by 16% as it effectively proxies traffic through Switzerland, which enforces continental routing for some routes (e.g. Poland - Switzerland - Spain as compared to Poland - UK - Spain). 99% of WeChat measurements within Europe were accessible to 5 Eyes because of routing through the U.S. Only the Switzerland internal measurement offered a direct path to Hong Kong. As Switzerland has a direct path to the U.S. as well, this also explains the one case where EU can not access TextSecure messages. When using Threema within Switzerland, the application path remains in Switzerland as well, hence the EU cannot access those measurements.

Oceania to Oceania: As Australia and New Zealand are both members of 5 Eyes, obviously all measurements are accessible to the latter. It is remarkable that all WeChat traffic, e.g. generated by exile Chinese, is routed through China.

Asia to Asia: At a network level, both 5 Eyes, China and the European Union can access about 20% to 40% of traces sent within Asia. This is largely caused by the before mentioned Israel routing. 75% of Threema traffic is 5 Eyes accessible by routing to Switzerland through the U.S. Also, a large portion of WeChat traffic (46%) is accessible to 5 Eyes, both by Israel routing through the U.S. and by Singapore routing to WeChat's Chinese backend through an U.S. IXP.

North America to North America: As expected, 100% of traffic is 5 Eyes accessible. For Threema, traffic from Canada to Switzerland was again routed through a direct hop from Miami to Zurich, resulting in two measurements seeming inaccessible to EU.

South America to South America: Measurements from Argentina were routed through a direct tunnel from Miami to Zurich and hence were not accessible for the EU in our metric. Hence only 2 out of 3 Threema measurements from South America are accessible for the EU. However, South America's communication is, independently of the mobile messaging service being used, always susceptible to 5 Eyes.

Russia and Arab League: None of the measurements did traverse Russia or the Arab League. We hence excluded those from the table.

6 Summary and Conclusion

We conducted traffic locality measurements between 28 countries for four mobile messaging services. We found those apps to heavily distort locality of traffic and hence drastically widen the set of actors able to access it. With a few notable exceptions, e.g. when using Threema in Switzerland, this has large negative impact on the users' privacy. This could be alleviated by decentralizing the mobile messaging services' backend infrastructures or even the services themselves, using P2P techniques. With this being the first study on this particular topic, we hope to raise user and operator awareness for this problem. To conduct our measurements, we introduced the MATAdOR framework to analyze messaging traffic characteristics on mobile phones. A detailed overview over the MATAdOR framework can be found in [35]. We fully release both the MATAdOR

framework and the dataset produced in our measurements through our website[6]. This enables future work to easily validate our results or do further analysis, such as deeper protocol analysis on the apps. Future work might also include analysis of WeChat's regional optimization within China, focus on peer-to-peer services like Bleep, or further dissect protocols of mobile messaging services. To improve quality of path measurement results, future work could use additional techniques such as fiber maps [3].

Acknowledgments: We thank Andreas Loibl for early access to his Measurement Proxy software.

References

1. R. Brandom. WhatsApp rolls out end-to-end encryption using TextSecure code. https://www.theverge.com/2014/11/18/7239221/whatsapp-rolls-out-end-to-end-encryption-with-textsecure, 2014. Accessed Sep 14, 2015.
2. Daum Kakao. 2Q15 earnings report. http://www.kakaocorp.com/upload_resources/ir/siljeok/siljeok_20150813080737.pdf, August 2015. Accessed Sep 23, 2015.
3. R. Durairajan, P. Barford, J. Sommers, and W. Willinger. InterTubes: A Study of the US Long-haul Fiber-optic Infrastructure. In *SIGCOMM'15*.
4. Electronic Frontier Foundation. Secure Messaging Scorecard. https://www.eff.org/secure-messaging-scorecard, 2014. Accessed Sep 14, 2015.
5. Facebook. Messenger at f8. http://newsroom.fb.com/news/2015/03/messenger-at-f8/, March 2015. Accessed Sep 17, 2015.
6. Federal Network Agency for Electricity, Gas, Telecommunications Post and Railway. Annual report 2014. *Annual Report 2014*, page 81, 2014.
7. P. Fiadino, M. Schiavone, and P. Casas. Vivisecting WhatsApp in Cellular Networks: Servers, Flows, and Quality of Experience. In *Traffic Monitoring and Analysis*. 2015.
8. T. Frosch, C. Mainka, et al. How Secure is TextSecure? Technical report, 2014.
9. J. Golson. Apple fighting the US government over encrypted iMessages. http://www.techrepublic.com/article/apple-fighting-the-us-government-on-turning-over-encrypted-imessages/, 2015. Accessed Sep 14, 2015.
10. Q. Huang, P. P. Lee, et al. Fine-Grained Dissection of WeChat in Cellular Networks.
11. B. Huffaker, M. Fomenkov, and k. Claffy. DRoP: DNS-based router positioning. *ACM SIGCOMM CCR*, 44(3), 2014.
12. J. Koum. Whatsapp - now serving 900,000,000 monthly active users. https://www.facebook.com/jan.koum/posts/10153580960970011, September 2015, Accessed Sep 23, 2015.
13. Line Corporation. LINE Corporation Announces 2015Q2 Earnings. http://linecorp.com/en/pr/news/en/2015/1043, Juli 2015. Accessed Sep 17, 2015.
14. M. Marlinspike. A Saudi Arabia telecom's surveillance pitch. http://www.thoughtcrime.org/blog/saudi-surveillance/, 2013. Accessed Sep 14, 2015.
15. L. McMurchy. Skype connection hub ads provide increased scale for marketers. http://advertising.microsoft.com/en/blog/29331/skype-connection-hub-ads-provide-increased-scale-for-marketers, December 2014. Accessed Sep 17, 2015.

[6] http://net.in.tum.de/pub/mobmes/

16. R. Mueller, Schrittwieser, et al. What's new with WhatsApp & Co.? Revisiting the security of smartphone messaging applications. In *iiWAS*, 2014.
17. Open Whisper Systems. Textsecure, now with 10 million more users. https://whispersystems.org/blog/cyanogen-integration/, Dec 2013, Accessed Sep. 23, 2015.
18. Pew Research Center. Mobile messaging and social media 2015. http://www.pewinternet.org/files/2015/08/Social-Media-Update-2015-FINAL2.pdf, 2015. Accessed Sep 14, 2015.
19. PlanetLab Central. User tools. https://www.planet-lab.org/tools. Accessed Sep 17, 2015.
20. R. Reader. Wickr CEO Nico Sell: behind the glasses. http://venturebeat.com/2015/01/13/wickr-ceo-nico-sell-behind-the-glasses/, January 2015. Accessed Sep 23, 2015.
21. Statista. Number of monthly active viber users. http://www.statista.com/statistics/316423/, April 2015, Accessed Sep 23, 2015.
22. Statista. We are social. (n.d.). most popular global mobile messenger apps as of August 2015. http://www.statista.com/statistics/258749/, Accessed Sep 23, 2015.
23. Statista. Number of monthly active WhatsApp users worldwide. http://www.statista.com/statistics/260819/number-of-monthly-active-whatsapp-users/, Accessed September 23, 2015.
24. Tango. 200 million members! http://www.tango.me/blog/200-million-members, March 2014. Accessed Sep 17, 2015.
25. TechCrunch. Chat app kik hits 200m registered users. http://techcrunch.com/2015/01/28/dont-expect-kik-maus/, Jan 2015, Accessed Sep 23, 2015.
26. Tencent. 2015Q2 results. http://www.tencent.com/en-us/content/ir/news/2015/attachments/20150812.pdf, August 2015. Accessed Sep 23, 2015.
27. The European Commission. Case No COMP/M.7217 - FACEBOOK/ WHATSAPP. http://ec.europa.eu/competition/mergers/cases/decisions/m7217_20141003_20310_3962132_EN.pdf, 2014.
28. The Telegram Team. Telegram reaches 1 billion daily messages. https://telegram.org/blog/billion, December 2014. Accessed Sep 17, 2015.
29. Threema. If you value security and privacy. https://threema.ch/press-files/1_press_info/Press-Info_Threema_EN.pdf, September 2014. Accessed Sep 17, 2015.
30. TNS Global. The new social frontier: Instant messaging usage jumps 12%. http://www.tnsglobal.com/press-release/new-social-frontier-instant-messaging-usage-jumps, 2015. Accessed Oct 7, 2015.
31. Vodafone. Law enforcement disclosure report 2015. https://www.vodafone.com/content/index/about/sustainability/law_enforcement.html, 2015. Accessed Sep 14, 2015.
32. M. Wählisch, T. Schmidt, et al. Exposing a nation-centric view on the German internet–a change in perspective on AS-Level. In *Passive and Active Measurement*, 2012.
33. Wikipedia. Mobile application testing. https://en.wikipedia.org/wiki/Mobile_application_testing#Some_Mobile_Application_Testing_Tools, 2015. Accessed Sep 17, 2015.
34. M. Zhang, Y. Ruan, V. S. Pai, and J. Rexford. How DNS Misnaming Distorts Internet Topology Mapping. In *USENIX*, 2006.
35. J. Zirngibl. Security Analysis of Mobile Messaging Traffic with an Automated Test Framework. Bachelor's thesis, Technische Universität München, 2015.