Hope of Delivery: Extracting User Locations From Mobile Instant Messengers

Theodor Schnitzler*†, Katharina Kohls‡, Evangelos Bitsikas§¶, and Christina Pöpper¶
*Research Center Trustworthy Data Science and Security, TU Dortmund, Germany †Ruhr-Universität Bochum, Germany
‡Radboud University, Netherlands §Northeastern University, USA ¶New York University Abu Dhabi, UAE
theodor.schnitzler@tu-dortmund.de kkohls@cs.ru.nl bitsikas.e@northeastern.edu christina.poepper@nyu.edu

***Abstract*—Mobile instant messengers such as WhatsApp use delivery status notifications in order to inform users if a sent message has successfully reached its destination. This is useful and important information for the sender due to the often asynchronous use of the messenger service. However, as we demonstrate in this paper, this standard feature opens up a timing side channel with unexpected consequences for user location privacy. We investigate this threat conceptually and experimentally for three widely spread instant messengers. We validate that this information leak even exists in privacy-friendly messengers such as Signal and Threema.**

Our results show that, after a training phase, a messenger user can distinguish different locations of the message receiver. Our analyses involving multiple rounds of measurements and evaluations show that the timing side channel persists independent of distances between receiver locations – the attack works both for receivers in different countries as well as at small scale in one city. For instance, out of three locations within the same city, the sender can determine the correct one with more than $80\,\%$ accuracy. Thus, messenger users can secretly spy on each others' whereabouts when sending instant messages. As our countermeasure evaluation shows, messenger providers could effectively disable the timing side channel by randomly delaying delivery confirmations within the range of a few seconds. For users themselves, the threat is harder to prevent since there is no option to turn off delivery confirmations.

I. Introduction

In recent years, messaging applications (or messengers) have become the de-facto standard for mobile communication. They have transitioned into integral parts of daily lives, with the most prominent messenger, WhatsApp, connecting more than two billion monthly active users world-wide [47]. Messengers are used in a wide range of scenarios, from informal communication among working colleagues [29] and social engagement among elderly people [31], to parents coordinating school matters [44] and citizens organizing neighborhood watches [11]. In some cases, messengers are also used for official communication with government authorities [25], [38], thus composing large and heterogeneous sets of contacts in one application per user.

Whenever a user sends a message in a messenger, the client application displays the current status of the message – from being in transit, processed and forwarded by the messenger server, to delivered to the recipient, and (if enabled) read by the recipient [2], often indicated by small symbols such as checkmarks. This is helpful information for users to track if a message has successfully reached its destination.

However, as we will demonstrate in our paper, this feature can also serve as a side channel that allows to learn sensitive information about message recipients, such as revealing information about their current whereabouts, with undesired potential harm to location privacy.

In more details, we conduct a series of experiments in Signal [34], Threema [49], and WhatsApp [55] to evaluate and demonstrate to what extent we can classify different message receivers and their respective locations based on delivery notification timings of a set of subsequently sent messages. Deriving sensitive information about someone by sending them a few messages is problematic because it is simple, rather unsuspicious, and hard to mitigate. Users cannot effectively prevent receiving messages from people in their contact list, except for permanently blocking them and, therefore, stopping having mobile conversations with them at all.

Based on characteristics such as the location of a receiver, delivering a message and returning the respective confirmation takes a specific amount of time. Physical transmissions on the Internet are influenced by the travelled distance, they depend on the network topology, i. e., routing and the hops in-between, and processing by the messaging service. We show that *sending messages* using each of these three messengers *to receivers at different locations results in different and distinguishable delivery notification timing patterns*.

This issue is critical for multiple reasons: First, all three messengers we examine are generally considered secure as they use end-to-end encryption between clients. It is not intuitive for users that the mere usage of the messenger service may leak information about their whereabouts. Second, Signal and Threema are best known for their focus on privacy – Signal's protocol serves as the blueprint for provably secure key establishment between clients [9] and has been adapted by other applications such as WhatsApp. Leaking information of the user's location contradicts this notion of privacy. Third, a user cannot do much about someone in their contact list sending them instant messages. Other than read receipts that can be turned off by the receiver for privacy reasons, there is no such option for delivery notifications [54].

In order to experimentally validate this concept we need to take into account the server infrastructures of messengers.

Network and Distributed System Security (NDSS) Symposium 2023
27 February - 3 March 2023, San Diego, CA, USA
ISBN 1-891562-83-5
https://dx.doi.org/10.14722/ndss.2023.23188
www.ndss-symposium.org

This information is not publicly shared and it is a challenge in itself to reliably extract the relevant information such as the number and locations of messenger servers. To this end, we conduct experiments to collect and aggregate information about the geographical distribution of servers of popular instant messaging services and analyze if and how knowledge about the messaging server in use affects the outcome of the delivery timing evaluation. We note that the server infrastructure setup does not change frequently, so this step would not have to be redone for each user localization attempt. Beyond the proof-of-concept attack done in this work, knowledge about the messenger infrastructure may be useful for other purposes.

In summary, our paper makes the following contributions:

1) **Messenger Infrastructure Analysis.** We aggregate and provide an overview of the geographical distribution of servers of mobile messaging services from a series of experiments to discover and analyze their infrastructures.
2) **Empirical Messaging Experiments.** We conduct large-scale measurements collecting the transmission timings of message delivery notifications between devices in multiple locations in Europe and the Middle East.
3) **Attack and Countermeasure Evaluation.** We demonstrate to what extent we can distinguish different receivers and their respective locations from each other based on the measured delivery notification timings. We also show that this threat can be mitigated by randomly delaying delivery notifications in the range of a few seconds.

Experimental Overview: Figure 1 provides an overview of our experiments for each of the three parts, their results and connections with each other. We start with infrastructure discovery experiments that result in sets of server locations used to determine the infrastructure overhead in the messaging experiments. At the core of our study, we use sequences of message delivery notification timings to classify receiver locations at different granularity levels and measure the accuracy.

Disclosure Process: The timing side channel exploited in this paper may potentially affect the location privacy of millions of messenger users. Following the guidelines of responsible disclosure, we got in contact with the providers of the messenger apps (Signal, Threema, WhatsApp) and reported the vulnerability to them prior to the submission of this paper in May 2022. Whereas Signal and WhatsApp have not acknowledged the issue to date (October 2022), we have exchanged ideas for mitigating the problem with Threema and they are currently evaluating how specific countermeasures (cf. Section VI) would affect user experience.

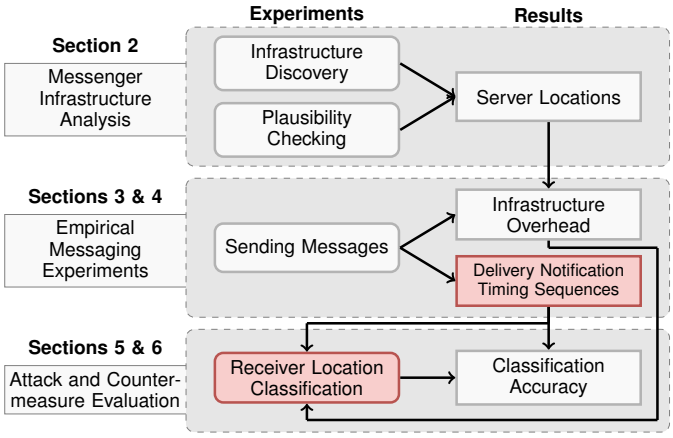


Fig. 1: Structural overview of the sequence of experiments (rounded nodes) and their outcomes (square nodes) in our paper and how the three main parts build upon each other.

II. Messenger Infrastructure Analysis

Our first goal is to obtain a comprehensive overview of the infrastructures of the messengers we use in our experiments, i. e., for Signal, Threema, and WhatsApp. For the delivery notification timing analysis, knowledge about the infrastructure is crucial to assess the different parts of the connection between sender and receiver, their distances, and timings.

A. Discovery and Aggregation

In order to gain first insights into the messenger infrastructures, we conduct a set of experiments to identify servers used by messaging services. In the first step, we set up two smartphones running client applications for all messengers under consideration and capture their network traffic when the applications are running. From the collected captures, we extract the IP addresses of the servers that the application on the smartphone connects to. Since we assume that messenger servers are geographically distributed, the resulting sets of IP addresses may only represent specific fractions of the messenger infrastructures, i. e., they comprise servers near to our own location.

To broaden the perspective derived from our local observations, we perform a two-step DNS analysis, as follows:

(1) For all IP addresses that appear in the communication using one of the messaging applications, we perform reverse DNS lookups to learn what (sub)domain names are used by the messenger operations.
(2) For each domain name in the set derived from reverse look-ups, we perform federated DNS resolving from multiple locations across all continents.

We continue to describe the exact procedures for each messenger individually.

1) Signal: For Signal, two specific IPv4 addresses are in use. Reverse DNS lookups point to the same domain name operated by Amazon Web Services (AWS), also when we perform these lookups from different geographical locations. When we resolve the resulting domain name, the same two IP addresses are returned, irrespective of the location. Even though the order of the two addresses varies, there is no indication that one address is preferred over the other at specific locations.

2) Threema: For Threema, we identify two similar IP addresses from the same IPv4/24 address range, for one of which the reverse DNS lookup points to a `threema.ch` domain name. Reverse lookup fails for the other address. We manually identify several more IP addresses whose domain names are resolved to `threema.ch`, resulting in an extended set of 12 IP addresses. However, it is unclear if all these IP

addresses are actually used for the messaging application of if they serve other purposes related to the same domain.

3) WhatsApp: Our reverse domain name resolving of server IP addresses reveals that WhatsApp establishes connections to servers in five different domain name ranges. Additionally, different servers within the same domain name range have been used. Irrespective of the location at which we perform the reverse DNS lookup for a particular IP address, it is resolved to the exact same domain name. Across the three messengers, we discover the largest number of different IP addresses when we explore the network traffic of WhatsApp.

The WhatsApp domain names within the same namespace only differ in 3-letter strings which appear to be IATA airport codes[1] near our experimental locations. Random checks of additional domain names with the identifier replaced with different ones (in other regions all over the world) reveal further IP addresses, strengthening our assumption.

Since all tested domain names resolve to similar IP addresses in five different IPv4/16 subnets, we conduct a full search of the respective address ranges. We record all domain names and their corresponding IPv4 addresses that contain a reference to WhatsApp (cf. Table I). We further extend the resulting set by manually spot-checking even more identifiers, which leads to a small number of additional servers. In total, our set of discovered WhatsApp servers comprises 410 server instances using 143 different location identifiers.

TABLE I: Namespace prefixes used by WhatsApp servers.

Namespace (Prefix)	No. of IPs	No. of Locations
fna-whatsapp	126	75
whatsapp-chatd-edge	94	73
whatsapp-chatd-msgr-edge	92	72
whatsapp-cdn	92	72
whatsapp-pp	6	4
Total Unique IPs/Locations	410	143

B. Location Analysis

In the next step, we map messenger servers to their individual geographical location and validate the mapping with the help of simple plausibility checks. We initially map each messenger server identified in Section II-A, i. e., their IP addresses, to a specific geographical location. We use different strategies depending on the information that we can obtain per messenger.

Little official information about messenger infrastructures is made public by their providers. In the set of messengers we explored, only Threema mentions that their servers are located in the Zurich area, Switzerland [50]. For Signal, no official information is available but several sources indicate that servers are hosted by AWS at the US east coast [4], [14], [45], [56] which is presumably located near Ashburn, VA.

The only information we find with relation to WhatsApp is a list of the locations of Facebook data centers on their website [15]. It is, however, unclear if these locations are also related to WhatsApp. We additionally take into account the presumable IATA location identifiers within the domain names associated with IP addresses used by WhatsApp. We perform look-ups for all 143 codes that appear in our data set and use the resulting city as baseline location for the server. In a few cases, identifiers could not be resolved – and we manually annotate them. For example, the codes *frx* and *frt* most likely belong to the area of Frankfurt, Germany (whose original IATA identifier is *fra*).

[1] https://www.iata.org/en/publications/directories/code-search/?airport.search=

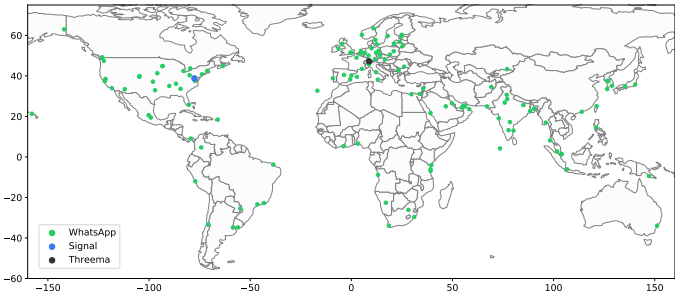


Fig. 2: Locations of Signal, Threema, and WhatsApp servers around the world (larger version in the Appendix).

We continue with a series of systematic Ping and Traceroute experiments from different geographical locations using a public API provided by CheckHost [7]. Over a period of four weeks we collect ping and routing information to all messenger servers. To confirm a location candidate as correct, we require that the shortest Ping time is received by the probe host that is closest to the location candidate and only accept minor deviations.

Whereas for WhatsApp and Threema the results are consistent and confirm our initial assumptions about the baseline, the case is more difficult for Signal. Ping information is heavily inconsistent with results being within less than 10 ms from all different continents, which suggests that they are returned from different physical locations close to each of the probing hosts. While Traceroute information can only be partially retrieved for Signal, they include traces with hosts that are likely located in the US, which again strengthens the initial assumption of Signal servers to be US-based.

Figure 2 shows our extracted geographical overview of the server locations for the three messengers (a larger/more readable version can be found in Figure 14 in Appendix B).

III. Message Status Timing Side Channel

The main idea of the attack we present is the use of a timing side channel provided by message status information to derive characteristics of a target user's Internet connection. Whenever two users are in each other's contact list of a mobile messaging application, i. e., they have accepted to be in a conversation on that messenger, the application shows status information for exchanged messages.

Small icons (e. g., check marks) along with each message indicate whether a message has been sent to the messenger server, delivered to the receiver, or read by the receiver. The messages between users as well as the information about the message status are exchanged through TCP messages between the client application and the messenger server. We measure

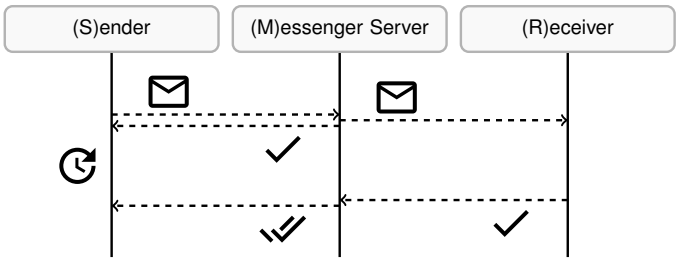


Fig. 3: Schematic overview of the message flow from the senders' perspective. The illustration is simplified since sender and receiver can be connected to different messenger servers.

the time between sending a message (i. e., the TCP packets containing the message leaving the sender's device) and the server and delivery confirmations (i. e., the TCP packets containing these confirmations) arriving at the sender's device. Observing the resulting timing difference allows us to reason about characteristics of the receiver, such as their location, or their network connection. A schematic overview of the information flow is depicted in Figure 3.

Factors such as the travelled distance between sender, server, and receiver, routing through the Internet between these entities, as well as processing at the messenger server and at hops in-between can affect the observed timings. Repeatedly sending messages to receivers under different conditions (e. g., location, network connection) and observing the timings between messages allows us to learn characteristics of the timings under these conditions in a controlled setup. For different receiver locations, the duration or the distribution of RTTs may be different, e. g., longer times likely represent longer distances between the communication partners.

Within our experiments, we demonstrate to what extent it is feasible to determine certain receiver characteristics upon observing delivery notification timings.

A. Threat Model

From a *technical* perspective, the adversary is required to operate a regular smartphone that is capable of running a messenger application. The adversary additionally needs to be able to access and analyze their own TCP traffic to extract timing information. This traffic can be captured either on a node in their local network, or directly running on the smartphone when running a packet capture app.

As an *operational* requirement for the attack, adversary and victim must be in each other's contact lists in the messenger. Thus, the threat is limited to parties who likely know each other, as the attack can only be conducted against users who have added the adversary to their contacts. However, the various contexts in which people have messenger conversations, be it in personal (extended family, acquaintances), professional (e. g., work collaborators) or other contexts (e. g., interaction with public institutions, clubs, authorities, within neighborhoods) in combination with low technical requirements still yield a considerable threat scope within social circles, e. g., for stalking.

In an initial training phase, the adversary sends messages to the victim and learns timing characteristics while knowing their whereabouts. Subsequently, the adversary can send new messages to the victim, and determine their location or network connection out of the set of previously seen plausible ones. Since the attack entails sending messages, the adversary's behavior might be observed by the victim and appear suspicious. Therefore, the attacker might leverage timings of messages they send anyway which would, however, narrow down the practical threat scope to people who regularly exchange larger numbers of messages.

B. Setup

We conduct measurements while sending messages between multiple smartphones in different geographical locations. Our setup comprises two types of devices:

(i) *Active* devices are used to send messages to other devices. Each active device is connected via USB to a computer scheduling the experiment and controlling the smartphone via Android Debug Bridge (ADB).
(ii) *Passive* devices are used to receive messages from active devices. The only requirement for a passive device is having an active Internet connection.

We conduct two rounds of measurements serving different purposes:

1): In the first round, we conduct long-distance measurements with devices distributed across different countries. During this round of measurements, each device is assigned a specific, permanent location. Out of three devices for active measurements, two are located in Germany (*DE-11* and *DE-12*) and one in Greece (*GR-11*). Our setup comprises three more passive devices, located in Germany (*DE-13*), the Netherlands (*NL-11*) and the Middle East (*AE-11*). This experiment is meant to demonstrate a proof of concept that the message-status timing side channel actually exists. For the sake of simplicity, all devices operated on a WiFi Internet connection for these measurements.

2): In a second round of measurements, we send messages from a single active device to passive devices at locations closer to each other, i. e., within the same city, and also rotate passive devices through these locations. Furthermore, passive devices switch between WiFi and cellular Internet connections. We replicate this type of setup in Germany (*DE-2X*) and the Middle East (*AE-2X*). This round of measurements is meant to demonstrate a more practical and realistic attack scenario, imitating a natural everyday behavior of a target messenger client, e. g., being at their home and work location (WiFi) and moving in between and around (cellular). Furthermore, this second round also shows to what extent the attack works at a smaller scale, which is less obvious than comparing timings at country level.

In Table II, we provide an overview of the devices and their locations involved in the two rounds of our experiments. For each location, we also indicate whether we use WiFi (W), or cellular (4G/4G+/5G) connections, or both for measurements at the respective location. Additionally, Table III lists distances between locations for all three setups.

C. Measurement Procedure

We measure the time it takes for a message from a sender device to be delivered to the messenger server and to the

TABLE II: Devices and locations in our measurements.

ID	Model (Year)	Type	Locations
Round 1			
AE-11	Huawei P40 (2020)	P	AE-A (W)
DE-11	Xiaomi Mi A3 (2019)	A,P	DE-A (W)
DE-12	Huawei P8 Lite (2017)	A,P	DE-B (W)
DE-13	OnePlus 7 Pro (2019)	P	DE-B (W)
GR-11	Samsung Note 10+ (2019)	A,P	GR-A (W)
NL-11	Samsung S6 (2015)	P	NL-A (W)
Round 2 (United Arab Emirates)			
AE-21	Huawei P40 (2020)	A	AE-B (W)
AE-22	Samsung Note 10 (2019)	P	AE-A, AE-D (W, 4G+)
AE-23	Samsung S22 (2022)	P	AE-B (W, 5G)
AE-24	Nokia X10 (2021)	P	AE-C (W, 4G+)
Round 2 (Germany)			
DE-21	Huawei P8 Lite (2017)	A	DE-A (W)
DE-22	Huawei P8 Lite (2017)	P	DE-A (W), DE-B (W, 4G), DE-C (W)
DE-23	Google Pixel 3a (2019)	P	DE-A (W, 4G), DE-B (W), DE-D (W)
DE-24	Samsung S6 (2015)	P	DE-A (W), DE-B (W, 4G), DE-E (W)

Locations: *AE-A,B,C,D:* Abu Dhabi, UAE; *DE-A,B,D,E:* Bochum, Germany; *DE-C:* Essen, Germany; *NL-A:* Nijmegen, Netherlands; *GR-A:* Athens, Greece

TABLE III: Distances [km] between device locations.

Round 1					*Round 2 (UAE)*				*Round 2 (Germany)*				
	DE-B	*NL-A*	*GR-A*	*AE-A*		*AE-B*	*AE-C*	*AE-D*		*DE-B*	*DE-C*	*DE-D*	*DE-E*
DE-A	1.5	98.7	1972.9	4981.0	AE-A	7.8	0.4	19.3	DE-A	1.5	14.4	3.4	5.4
DE-B		97.5	1974.4	4982.2	AE-B		8.1	24.9	DE-B		13.5	2.3	4.0
NL-A			2065.8	5079.5	AE-C			18.9	DE-C			11.2	10.3
GR-A				3263.3					DE-D				2.3

recipient. To this end, we capture an active smartphone's network traffic directly on the device using the *tPacketCapture* app. The phone is connected to a computer via USB and a Python script controlling the phone via *Android Debug Bridge (ADB)* automatically schedules the processes of sending messages and capturing network traffic. The script uses system commands to open and close the packet capture and messaging apps, and interacts with the UI to navigate within the apps, i. e., simulates human touch input to select contacts or type messages.

In a single experiment iteration, the phone subsequently sends a series of five messages to all receivers, with each messenger that is running on the sender and on the receiver device. The texts of the messages remain the same throughout the whole experiments. The first four messages are short and only consist of a single wordeach, while the last message is a whole text paragraph. We send the first four messages at an interval of 10 seconds to allow for the confirmations to arrive before sending the next message, while we increase the waiting time before the last messages to 20 seconds in order to accommodate the longer time it takes to type the long text, thus facilitating the analysis of the packet captures. The measurement procedure is complete when all iterations have terminated successfully for all recipients and their corresponding messaging applications. Algorithm 1 shows our measurement procedure.

Algorithm 1: Texting Thumb

input : A list of $messengers$ which are supported applications of the receivers
input : A list of $receivers$ according to the contact list
input : A list of $words$ which are sent to the receivers consecutively
output: $void$ function

```
sleep_time = 10;
num_of_messages = 5;
for receiver in receivers :
    for messenger in messengers :
        start_pcap ();
        start_app (messenger);
        open_chat (receiver);
        for i ← 0 to num_of_messages - 2 :
            send (words[i]);
            sleep (sleep_time);
        sleep (sleep_time);
        /* Send the long text          */
        send (words[num_of_messages - 1]);
        close_app (messenger);
        stop_pcap ();
```

We repeat this procedure over a period of several weeks in July and August 2021 for Round 1 and March to April 2022 for Round 2. Whereas the physical locations of receiving devices remain unchanged throughout the Round 1 measurements, we collect data for at least one week for each location a receiving device was placed at in Round 2. In total, we use more than 240,000 messages sent during the two rounds of experiments for evaluation.

IV. Descriptive Dataset Analysis

Using the setup described in Section III, we collected our dataset and use it in the further investigations.[2]

A. Data Processing

For each measurement iteration, we evaluate the recorded packet captures to determine the elapsed time between a message sent by the sender and the notifications (by the server and receiver) that return to the sender.

Since the messengers we consider use multiple layers of encryption (i. e., end-to-end encryption between the communication partners and TLS-encryption for connections between clients and servers on the transport layer), we are not able to access the contents of the communication. Yet to analyze the communication flow and identify the messages and confirmations, we develop heuristics from sample captures. We analyze characteristics of the network traffic such as packet sizes, their order and flow direction, which is a common technique, e. g., for traffic analysis [8], [48].

Within our analysis, we only consider traffic between the sender device and IP addresses within the IP address range(s) of the respective messaging service (cf. Section II). We are

[2]The raw data contain private location data we prefer not to share publicly.

TABLE IV: TCP packet lengths of notifications.

Messenger	Bytes (Server)	Bytes (Receiver)
Signal	123–124	773–828
Threema	38	158–390
WhatsApp	68–69	61–62

interested in sequences of packets of the form as illustrated in the information flow overview in Figure 3. The message sent by the sender usually consists of one or more outgoing TCP packets whose destination is one of the messenger servers. After a message has been sent, there is one incoming TCP packet containing the server notification, coming from the messenger server. Finally, once the receiver has confirmed that they have retrieved the message, there is another incoming TCP packet containing the delivery notification. From the sender's perspective, this packet is also coming from the messenger server. These observations are based on a first manual visual inspection of a small set of packet capture files.

Taking into account the aforementioned network traffic structure, we conduct our detailed packet capture analysis in two steps:

(1) Identifying typical packet sizes of server and receiver notifications.
(2) Matching sequences of TCP packets to determine round-trip times between sending a message and receiving the notifications.

1) Identifying Packet Sizes of Notifications: In the first step, we use a subset of $n = 1000$ randomly selected packet capture files and analyze the packet sizes of the two types of incoming packets (i. e., the notifications from server and receiver). To make sure that we only consider packets that contain these notifications, we limit our first analysis to sequences of packets that appear right after one another and right after the message has been sent.

We then analyze the lengths of the two inbound packets in all matched packet sequences across all packet capture files to identify the lengths of the packets containing the two types of notifications. We evaluate the frequencies of packet lengths, conducting an additional round of manual plausibility checks within the traces. The results are listed in Table IV. Most notably, the length of the packet containing the notification that a message has been delivered to its receiver in Threema is uniformly distributed between 158 and 390 bytes. In contrast, the other notifications have smaller variations in packet length: Signal's notifications range from 773 to 828, and WhatsApp's from 61 to 62.

2) Matching Packet Sequences to Determine RTTs: In the second step, we systematically analyze all packet captures we have collected during the two rounds of measurements. Since we now know the sizes of packets we are interested in, we omit the requirement of packets to appear right after one another in the correct order. This helps us to also identify messages whose delivery notification is delayed, or when the traffic patterns we are interested in interferes with other packets exchanged between the client application and the messenger server. We first identify the two inbound packets (i. e., the two notifications n_1 and n_2) based on their size and match them

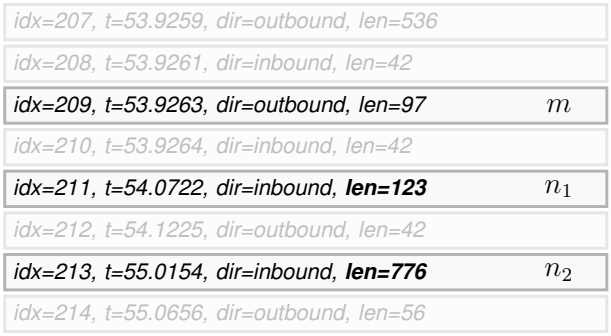


Fig. 4: Excerpt from an example packet capture with the three identified packets of interest highlighted.

with the latest outbound packet (i. e., the message m) sent before those two packets arrived. An example is illustrated in Figure 4. We use the timestamps of the three packets (i. e., $t(m)$ for message m) to determine the notification round-trip times (RTT) between (S)ender and (M)essenger Server, and (S)ender and (R)eceiver:

$$\begin{aligned} RTT_{S,M} &= t(n_1) - t(m) \\ RTT_{S,R} &= t(n_2) - t(m). \end{aligned} \tag{1}$$

Finally, we calculate the hypothetical RTT between (M)essenger Server and (R)eceiver:

$$RTT_{M,R} = RTT_{S,R} - RTT_{S,M}. \tag{2}$$

Additional Notes on Signal in the UAE: In the Signal data collected in Round 2 in the UAE, we observed different traffic characteristics. In particular, there is only one specific packet returned from the server – presumably containing both confirmations from server and receiver. Thus, we cannot determine the difference between the two but only consider $RTT_{S,R}$ for our analysis.

B. Delivery Notification Timings

We now present a first view into our delivery notification timing dataset. We start by analyzing the measured times in relation to the traveled distance, and later continue with distributions of timings to different receivers.

1) Timings and Distances: We are first interested in the relation between the timings we observe and the traveled distances between sender, messenger server, and receiver. To this end, we analyze what messenger servers have been picked on the sender's side and leverage the findings from our messenger infrastructure analysis (cf. Section II) to determine the distances from the server to sender ($dist_{GCD}(S, M)$) and receiver ($dist_{GCD}(M, R)$), respectively. We emphasize that the receiving device might be connected to a different server (location) than the sender – however, from the attacker's position (i. e., the sender), this information cannot be further resolved. We can then analyze the relation between timings and distances for the two segments.

In Figure 5a, we see a slight tendency for minimum timings to increase for longer distances between sender and server (for Threema and Whatsapp), even though timings are largely

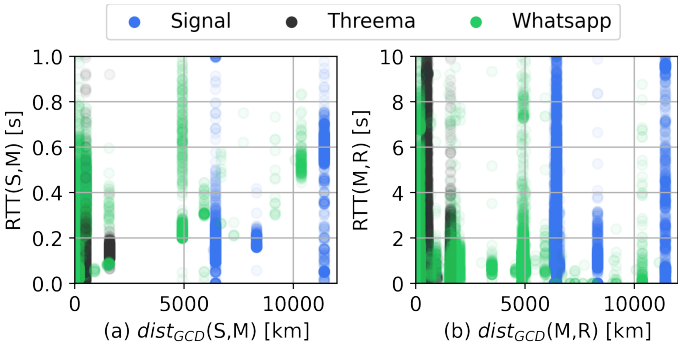


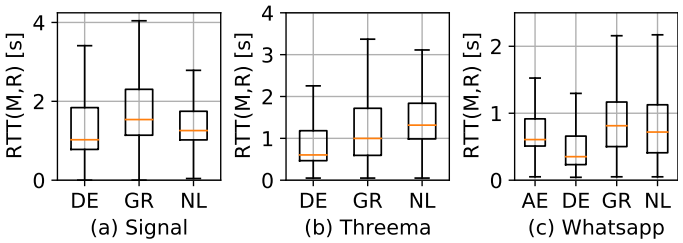
Fig. 5: Round trip time distributions of distance splits for (a) sender to server and (b) server to receiver – (each with 10,000 randomly sampled timings per measurement round)

scattered for similar distances. In Figure 5b, there is, again, a comparably small set of distances between servers and receivers, and timings being scattered a lot without clear trends. Since our experiments only cover a small set of distances between devices, and only consider Great Circle Distances (GCD) between entities, without taking into account the actual routing through the Internet topology, our dataset does not allow to develop a generalized model to put timings in direct relation to the traveled distances. To reduce the noise introduced into our data at this stage, we continue with focusing on the timings between messenger server and receiver, i. e., we use $RTT_{M,R}$ in subsequent analyses.

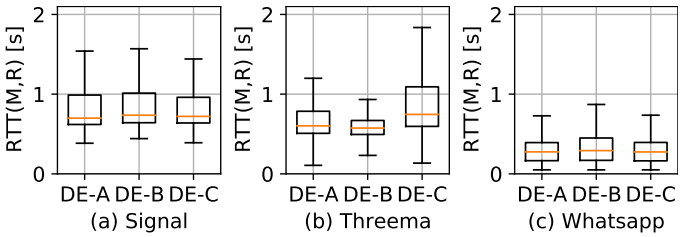
2) Differences between Receiver Characteristics: In the next step, we analyze to what extent timings we collected comprise differences between receivers, or their characteristics, respectively.

We first compare the measured $RTT_{M,R}$ between receivers the different countries involved in the first round of experiments. Figure 6a illustrates distributions of these timings of messages sent from device *DE-11* to receivers in different countries for each messenger. For all messengers, we observe that timings to Germany are shorter (lower medians) and tighter distributed (smaller boxes). Shorter timings for Germany are the result we expect in this case, since all messages have also been sent from a device in Germany. Whereas the differences between the medians are smaller for the other countries, distributions have different widths (heights of boxes) or are differently shifted (position of boxes).

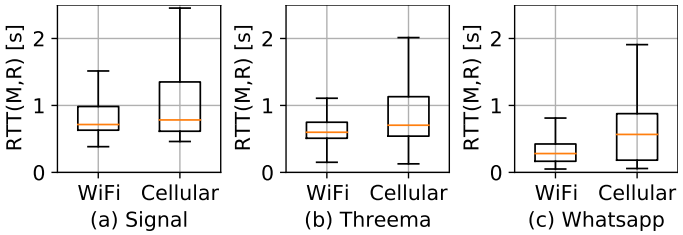
While differences between the distributions of notification timings to receivers in different countries can be easily identified, we also analyze if such differences also exist at smaller scale. Moreover, we cannot entirely exclude that these differences are partially grounded in the devices itself, since in the first round of measurements, each country location corresponds to a different device. In this regard, we now compare notification timings of messages sent to device DE-22 at its different locations in Germany during the second round of measurements. Figure 6b shows the distributions of timings to the three locations. Differences appear to be much smaller than those on the per-country level, we can only observe small variations in, e. g., medians or ranges of timing distributions, indicated by ranges and shapes of boxes.



(a) Messages sent from device *DE-11* to receivers in different countries. Y-axes have different ranges since we only intend to highlight differences within each messenger.



(b) Messages sent to device *DE-22* separated by the device's location.



(c) Messages sent to device *DE-22* separated by its network connection.

Fig. 6: Empirical distributions of $RTT_{M,R}$ [s] for messages sent at different stages of our experiments.

In the last step, we also compare notification timings sent to the same device depending on its network connection. In this case, differences appear to be larger again, with distributions of timings of messages received over cellular data showing a higher variance (larger box) and being slightly slower, indicated by a higher median (cf. Figure 6c).

V. Delivery Notification Timing Classification

Classifying the timing measurements collected in the experiments can help to determine certain characteristics of the receiver of a message, such as their location. We demonstrate at what scale it is feasible to classify different targets based on delivery notification timing measurements and to distinguish these characteristics from each other.

A. Classification Tasks

To evaluate and demonstrate at what scale the classification of receivers and their characteristics works, we specify a set of classification tasks at different granularity levels as follows:

(1) *Country*: We distinguish our measurements by the country a receiving device is located in (out of the set of countries we have measurements for).
(2) *Within Country*: We only distinguish whether or not a receiving device is located within a specific country.
(3) *City Location*: We distinguish timings to different locations within the same city. We repeat this classification task for devices individually and conjointly.
(4) *Network Connection*: We distinguish whether a device is using a WiFi or a cellular Internet connection.

According to the designs of our measurement setups (cf. Section III-B), we use data from the first round of measurements for classification tasks (1) and (2), whereas classification tasks (3) and (4) are based on data from the second round of measurements.

B. Classification Setup and Parameter Tuning

We use *sequences* of delivery notification timings for classification. A sequence is a set of notification timings derived from up to five subsequently sent messages (cf. Section III-C). We repeat the classification with different sequence lengths, starting with $n = 1$, i. e., a single notification round-trip time from a single message.

For each classification task, we analyze the measurement data for each sender device and for each messenger independently. We randomly sample k notification timing sequences from each class, whereas k is the number of timing sequences of the class with the lowest number of sequences. This way, we reach an evenly weighted set of samples per class.

We use convolutional neural networks (*CNN*) as classifiers, train them with sequences of delivery notification timings from different classes and then measure their accuracy in predicting newly observed timing sequences. This selection is grounded in our own preliminary parameter tuning evaluation and builds upon findings by Rimmer et al. [40], who extensively evaluate the performance of different types of neural networks for a similar network traffic analysis task (Website Fingerprinting) and report CNNs to perform best when compared to Long Short-Term Memory (LSTM) networks and Stacked Denoising Autoencoder (SDAE) networks. Before we start the actual classifications, we repeatedly run the first classification task with varying parameters to find the optimal classification setup for each of the three types of neural networks, i. e., CNN, LSTM, and SDAE and compare the results. Details about the parameter tuning configurations can be found in Appendix A.

C. Classification Procedure and Evaluation Metrics

For each classification task, we randomly split the respective data into five portions and use all but one of these portions as training set for the neural network. The remaining portion serves as test set from which all samples are to be classified. For each sample in the test set, the neural network output comprises a softmax result, i. e., assigning each candidate class a probability that the classified sample belongs to this class. Based on the softmax output, we assign each sample the class with the maximum probability, considering this as the classification decision. To avoid model over-fitting, we repeat this procedure until each of the five data portions has served as test set and merge the five classification results, effectively implementing 5-fold cross-validation.

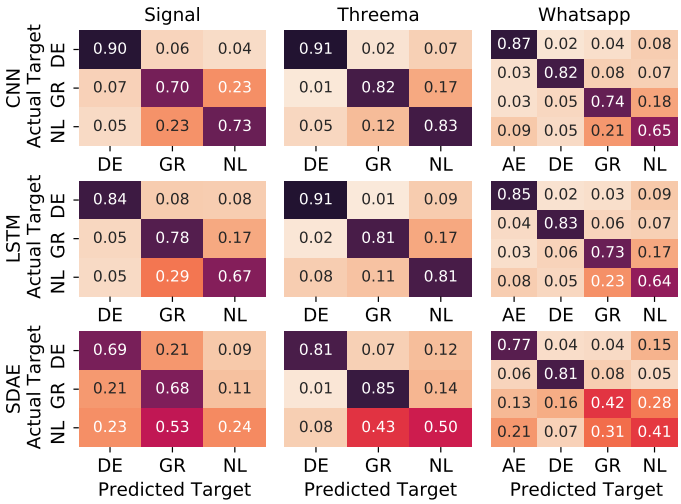


Fig. 7: Detailed classification results for the receiver *country* based on measurements from sender *DE-11* with three different neural network types. For each classification, numbers report *precision* values, i. e., the fractions of predicted classes (x-axis) given the actual class (y-axis).

The performance of the classification is determined by the numbers of classifications that identify the correct class (*precision*) and the number of samples in each class that are correctly classified (*recall*). In our evaluation, we focus on precision, i. e., we are interested in the fraction of samples per class that can be correctly identified and how the classifications are distributed for all samples of a particular class. We also analyze changes in classification performance when we vary the sequence length.

We report these detailed results for the first classification task (i. e., distinguishing receiver countries) to provide detailed insights into our evaluation and how it works. For subsequently presented classification tasks, we report overall accuracy results for a large number of different classifications using the maximum delivery notification sequence length (i. e., 5 messages). We do so to provide a broad overview of the varying effectiveness of leveraging the timing side channel in different scenarios.

Finally, we also analyze the convergence of the classification accuracy depending on the sample size, i. e., we repeat a selected set of classifications multiple times with increasing numbers of samples per class, and measure the resulting performance.

D. Receiver Classification by Country

In the first step, we present the results of the receiver *country* classification for one sender device in Germany (*DE-11*). For WhatsApp, the receiver can be one of four countries (*AE*, *DE*, *GR*, *NL*). For the two other messengers, we cannot present data for *AE* due to the messenger not being available at all in the country (Threema) or too little successful delivery notification measurements (Signal). To this end, we are restricted to the three remaining countries for Signal and Threema.

TABLE V: Detailed precision results for the classification of receiver countries (CNN-based classification).

Sender	DE-11			DE-12			GR-11	
Messenger	SIG	THR	WA	SIG	THR	WA	THR	WA
Classification Task: Country								
AE	–	–	84 %	–	–	94 %	–	95 %
DE	90 %	94 %	81 %	73 %	70 %	77 %	71 %	89 %
GR	77 %	84 %	79 %	53 %	68 %	64 %	–	–
NL	70 %	80 %	63 %	61 %	68 %	53 %	66 %	88 %
Samples/Class	177	527	825	66	60	135	187	168
Overall Acc.	79 %	86 %	77 %	62 %	69 %	72 %	68 %	90 %
Classification Task: Within Germany								
DE	92 %	91 %	90 %	86 %	84 %	92 %	90 %	90 %
NOT-DE	91 %	94 %	92 %	78 %	85 %	88 %	51 %	94 %
Samples/Class	559	1135	1888	250	180	605	187	349
Overall Acc.	91 %	92 %	91 %	82 %	85 %	90 %	70 %	92 %

Detailed results are presented in confusion matrices in Figure 7, separated by messenger (columns) and neural network type (rows). The numbers indicate the fractions of predicted classes for each actual class (*precision* values). A darker principal diagonal in each matrix indicates higher accuracy since numbers on this axis refer to correct predictions. Figure 8 illustrates corresponding overall accuracy for this classification tasks for all three messengers depending on the length of the notification timing sequence.

For Signal (left column matrices), the receivers located in Germany can be distinguished from receivers in the two other countries quite well. We observe false classifications mostly between devices in GR and NL. This result is not surprising since timing distributions for GR and NL largely overlap, whereas timings of messages to receivers in DE are lower (cf. Figure 6a). The overall classification accuracy rises from 60 % for a single timing per sample to 79 % for 5 timings per sample (cf. Figure 8) in the case of a CNN classification. For Threema, there is a quite similar outcome. Again, classification works best for receivers in *DE* with most false classifications between *GR* and *NL*. For longer timing sequences, Threema reaches a better overall accuracy of 86 % for 5 timings per sample, compared to 60 % for single-time samples. In the case of WhatsApp, receivers in *DE* and *AE* can be distinguished best from the others and performance increases for longer timing sequences. The overall accuracy is a bit lower for the other two messengers (i. e., 47 % to 77 %).

Regarding the classifier type, CNN and LSTM perform with similar quality with CNN reaching slightly higher performances in most cases. SDAE results are noticeably worse. Therefore, and taking into account previous findings [32], [40], we continue with CNN throughout the remaining evaluations.

Table V lists precision results per class for the country classification for messages sent with all three messengers from three sender devices (*DE-11*, *DE-12*, and *GR-11*). We also report sample sizes of notification timing sequences there. All results listed in the table refer to the maximum notification timing sequence length, i. e., timings of *five* subsequently sent messages. The results in the top left block of the table correspond to the numbers presented in Figure 7 with each table column corresponding to the principal diagonal axis in the respective confusion matrix.

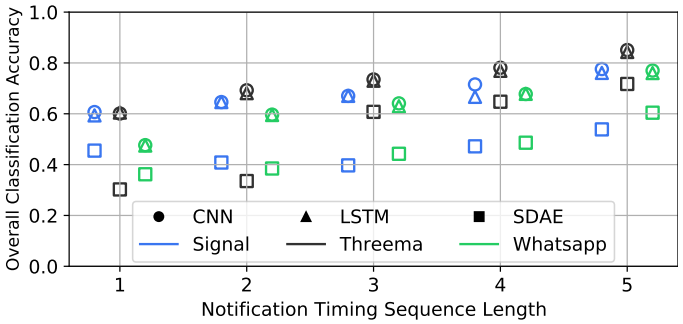


Fig. 8: Overall classification accuracy (y-axis) for the receiver classification per country, depending on delivery notification timing sequence length (x-axis) and NN type (icon shape)

.

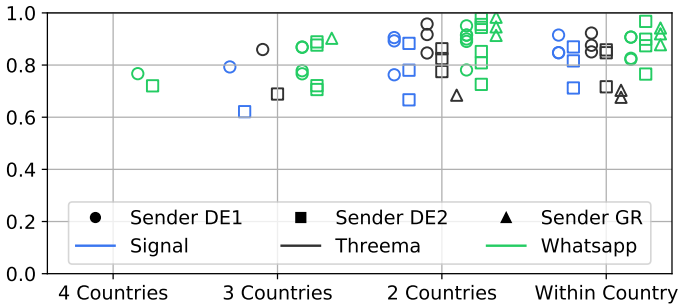


Fig. 9: Overall accuracy of receiver country classification separately for all possible country subsets for each sender device (icon shape) and messenger (colors).

1) Country Subsets: We repeat the classification of delivery notification timing sequences with the other devices and for every subset of countries in our data set. The resulting set comprises one more classification of four countries (sender device *DE-12*) and multiple evaluations of all possible pairs and triplets of countries including measurements from all three sender devices. In this context, we only consider the maximum sequence length, i. e., delivery notification timings of $n = 5$ subsequently sent messages.

Figure 9 shows the overall accuracy values of the receiver country classification for all combinations of countries in our data set. For smaller target sets, classifications perform better, with overall classification accuracy mostly between 70 % and 90 %. In the case of two countries, some classifications even perform with more than 95 % accuracy. Such nearly perfect results can only be achieved when timings can be clearly distinguished, which is mostly the case when the candidate locations are far from each other (one receiving device located in the UAE and the other one in a European country). However, also for distinguishing notification timings of messages sent to Germany and to the Netherlands (*DE11-2countries1*), we achieve a classification accuracy of more than 90 % (92 % for Threema and 91 % for Signal and WhatsApp).

2) Within Country: In the second classification task, we are interested in whether or not a receiver is located in a specific country. Different from the previous task, we are not interested in determining the exact location but only in

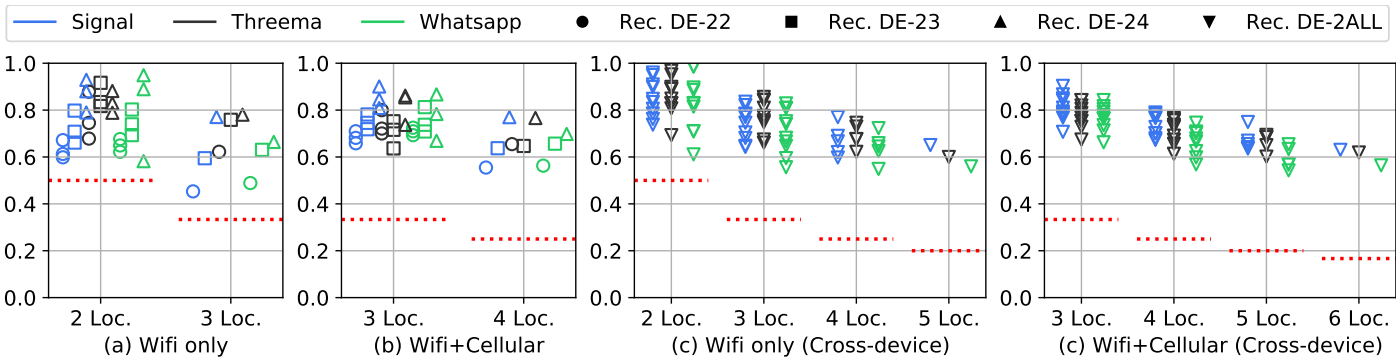


Fig. 10: Overall accuracy of receiver location classification separately for all possible combinations of locations in Germany. Colors indicate messengers and icon shapes indicate different receiver devices (we refer to Rec. DE-2ALL for the cross-device analysis). Dotted lines indicate the probability of randomly guessing the correct location out of the set of known locations.

a binary decision about a specific location. Therefore, we only distinguish notification timing sequences of messages sent to the country we are interested in (e. g., *DE*) from timings to any of the other countries, effectively summarizing timing sequences of all other countries into one class (e. g., *NOT-DE*). Technically, this type of prediction is similar to the classification of two countries.

Figure 9 also includes accuracy results for all such classifications, with the majority being very similar to the two-country classification. As an example, we provide more detailed precision results for the *Within Germany* classification task in Table V for all three sender devices.

E. Receiver Locations Within the Same City

We now present classification results for receivers at different locations within the same city to demonstrate that the timing side channel provided by delivery confirmations also persists at smaller scale. In this case, the end-to-end distances between sender devices, messenger servers, and receiver devices remain roughly the same across all measurements. Similar to the per-country classification, we consider all possible combinations of WiFi locations and subsets and evaluate the classification performance for each of them. Subsequently, we repeat the analysis also including the timing data retrieved from receivers operating on a cellular connection as a separate class. We repeat these analyses for receiver devices individually and across all devices within the same setup, i. e., the Round 2 measurements in Germany and in the UAE (cf. Table II). Whereas cross-device analyses provide first insights towards the generalizability of receiver location classification models (i. e., whether or not the classification requires training for each individual device), the individual analyses ensure that the classification is not biased by timing artifacts introduced by characteristics of the different devices.

1) Individual Receivers: The classification results for the three receiving devices in Germany are illustrated in Figure 10a+b. The accuracy highly varies between messengers, devices, and the respective combination of locations. Across all combinations of two locations, in each of which the device is connected via WiFi (a), the prediction performance can reach more than 90 % in some cases, e. g., when distinguishing locations *DE-A* and *DE-B* for the receiver device *DE-24* (*2wloc1-DE-24*). On the other side of the spectrum, there are also combinations of two locations which cannot be distinguished at all – a classification accuracy of around 60 % is hardly better than randomly guessing one of the two location candidates, e. g., when distinguishing locations *DE-B* and *DE-C* for device *DE-22* (*2wloc5-DE-22*). For distinguishing three WiFi locations, accuracy is lower with a maximum of 77 % for Signal, 78 % for Threema, and 66 % for WhatsApp (*3wloc3-DE-24*). However, the chance of randomly guessing one location is also lower in this case (33 %).

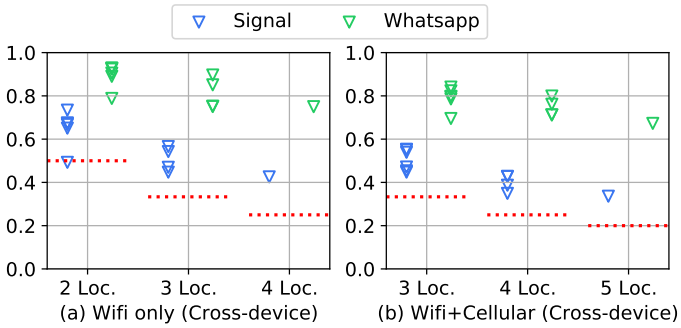


Fig. 11: Overall accuracy of receiver location classification separately for all possible combinations of locations in the UAE. Colors indicate messengers and dotted lines indicate the probability of randomly guessing the correct location.

Identifying the correct location becomes easier when the receiving device operates on a cellular connection in one of them (cf. Figure 10b). For distinguishing two WiFi locations and one on mobile data, the classification accuracy is mostly between 60 % and 80 %. Such a scenario could, for example, model home and work locations of the device owner, whereas the cellular connection represents any other place in which the phone is not connected to a WiFi network.

2) Cross-device Analysis: When distinguishing locations across different devices (cf. Figure 10c+d), classification performs similar to the case of individual devices, with accuracy increasing slightly. Such differences might come from individual devices introducing specific timing characteristics into the dataset that facilitate distinguishability of locations.

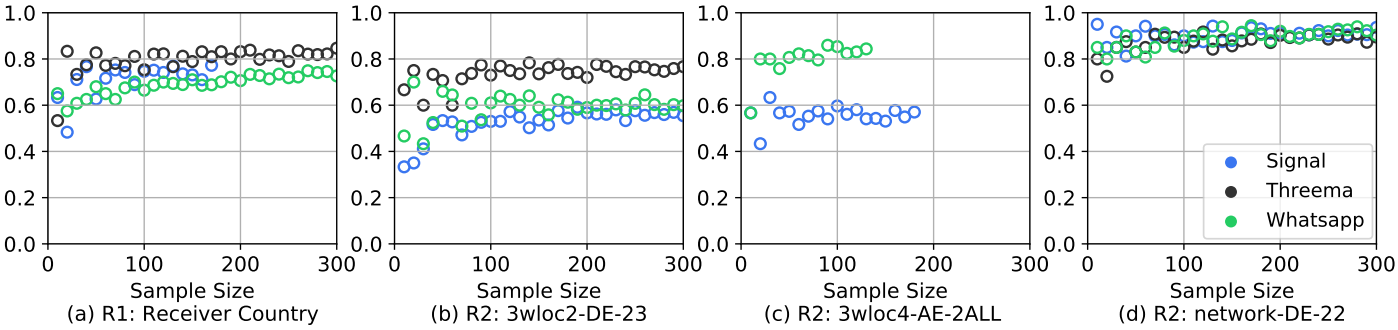


Fig. 12: Overall accuracy of four different classification tasks, depending on the number of samples per class (x-axis).

For the data collected in the UAE setup, the picture is more diverse. As the results in Figure 11 show, both two and three WiFi locations can be distinguished with up to more than 90 % accuracy in WhatsApp, which resembles better performance than comparable classifications in the German setup. However, for Signal, the classification of locations does not seem to work at all, which we attribute to the different structure of message exchange (and in particular the presence of only one confirmation packet) as described in Section IV-A.

F. Receiver Network Connections

Since different locations can apparently be better distinguished when the receiving device operates on a mobile data in one of them, we also analyze if we can generally detect whether a phone is connected via WiFi or using a cellular connection. Being able to distinguish these two cases allows us to determine whether a target is currently in one of their usual locations (i. e., we assume that they are connected to the respective WiFi network there) or not (mobile data).

The results for the evaluation of this classification task are listed in Table VI. In the setup in Germany, we can detect the receiver's Internet connection type with high accuracy for all devices for all messengers, both for individual devices and also across different ones. Classifications reach an overall accuracy of 90 % or even above, with only one prediction performing worse (Device *DE-23*, Threema). In the setup in the UAE, predicting the network connection performs on a similar level for WhatsApp. In the case of Signal, results do not seem convincing (50 % corresponds to randomly guessing the connection type), which is in line with results of the WiFi location distinguishability.

TABLE VI: Classification accuracy for receiving devices' network connections (WiFi vs. mobile data)

	Germany				UAE		
Receiver	SIG	THR	WA		Receiver	SIG	WA
DE-22	92 %	90 %	94 %		AE-22	54 %	91 %
DE-23	90 %	75 %	90 %		AE-23	61 %	89 %
DE-24	95 %	94 %	92 %		AE-24	77 %	90 %
DE-2ALL	91 %	85 %	88 %		AE-2ALL	62 %	87 %

G. Classification Accuracy Convergence

Whereas the results reported for the classification so far always refer to the maximum number of notification timing sequences available for all classes, we are also interested in how many samples are actually required for an accurate classification. To this end, we repeatedly run four specific classifications representing different classification tasks with increasing numbers of notification timing samples. We start with 10 samples per class and increase this number in steps of 10 until we reach 300 or the maximum number of available samples for all classes (if it is lower than 300). Figure 12 illustrates the results of these evaluations. We include (a) the receiver country classification based on the first round of measurements, two classifications of three WiFi locations, both (b) device-specific in Germany (device *DE-23*) and (c) cross-device in the UAE (referred to as *AE-2ALL*), and (d) a receiver network classification for one of the devices (*DE-22*) in Germany. Whereas the overall classification accuracy is varying for smaller sample sizes, there are only minor improvements for more than around 100 sequences of 5 delivery confirmation timings. This observation seems to hold for all three messengers and across the different classification tasks.Thus, we can already reach considerable classification results with sample sizes of around 100 delivery confirmation timing sequences per class – for some cases, e. g., the network connection detection, even with lower sample sizes.

H. Experimental Factors

While we are mostly interested in differences between receiver characteristics such as their location or network connection type, there are many dynamic features that can influence the RTTs of delivery confirmations, including network, device, and server characteristics. We now carefully discuss how such features are reflected in our measurements, and to what extent they can affect our experiments.

Network Characteristics: Varying network loads, both in terms of general Internet traffic and messenger use, may affect the time required to send a message and receive the confirmations. However, such circumstances cannot be influenced by our setup. In general, network loads are mostly varying depending on the time of day, with higher loads during mornings and evenings [16], [51]. Since we continuously collect data for at least one week per receiving device and location, all relevant load levels should be covered by our measurements. When looking into our timing dataset, we do

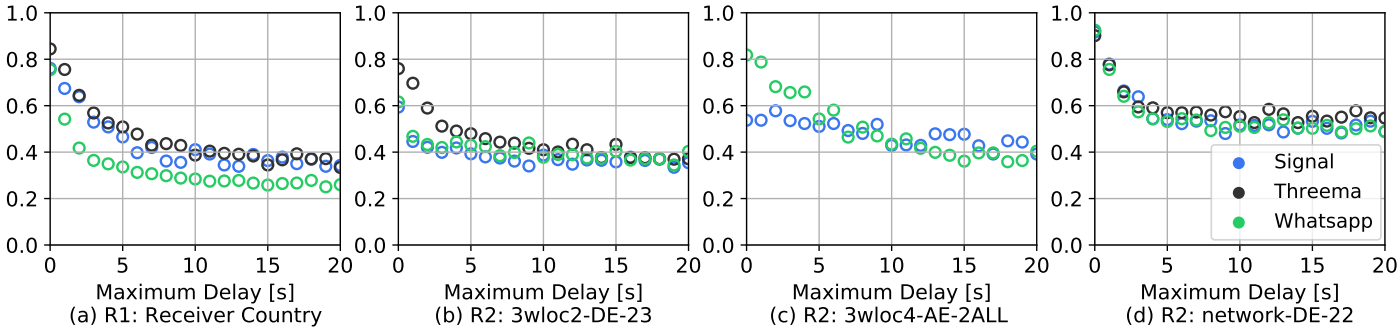


Fig. 13: Overall accuracy of four different classification tasks with increasing random delays (x-axis) added to message delivery confirmation timings. For higher delays, the accuracy approximates the chance of randomly guessing the receiver's location.

not observe large deviations or suspicious patterns depending on the time of day. Thus, the influence of network load on our dataset should be negligible. Figure 15 in Appendix C serves as an example and shows detailed distributions of $RTT_{M,R}$ for messages sent from device *DE-11* to receiving devices in their respective countries per hour of day.

Timings may also depend on the routes taken between sender, messenger server, and receiver, which could vary depending on the provider of the devices' Internet connections, making WiFi locations easier distinguishable when different connection providers are involved. In our measurements in Germany, only locations DE-C and DE-E were using the same connection provider but, unfortunately, our dataset does not include measurements of the same device in both locations. In the UAE, all Internet connections were provided by the same operator, with timings being fairly distinguishable (e. g., 82 % accuracy for *2wloc1-AE-22*). However, our dataset is too small, to adequately measure the effect of using the same provider at multiple locations vs. using different ones.

Device Behavior: During our measurements, receiving devices were idling at each location while receiving messages. This comprises a limitation of our setup, since active interaction with the devices and parallel processes may affect the timings we measure while sending messages, with potential consequences for classification accuracy.

To overcome this issue, we conducted additional experiments over one week sending messages to one author's private smartphone while it was in everyday use and continuously recorded its network connection type (i. e., WiFi or mobile data). We then used the data to predict its network connection following the procedures described in Section V. Classification reaches overall accuracy of 82 % for Signal, 80 % for Threema, and 74 % for WhatsApp. These numbers are fairly lower than the ones in our original and fully controlled setup (cf. Table VI) and shows that the threat vector still persists in a realistic usage profile, although with lower accuracy.

Server Behavior: Through the experiments, the sender devices were connected to different servers when sending WhatsApp messages. We only consider WhatsApp here, since Threema only has one server location and Signal's actual infrastructure remains unclear. While the same sender connected to up to 34 different WhatsApp IP addresses (*AE-21*), 3 servers (4 for *DE-21*, respectively) make up at least 95 % of connections used when sending messages. Additionally, server selection follows similar distributions for all receiver locations. Thus, the selected server should have little unintended influence on our measurements. While our data does not contain meaningful differences in round-trip times depending on the selected server, it may be possible that strategic server selection could help the attacker (e. g., by locally changing DNS resolution) to make timings better distinguishable, i. e., further improve classification accuracy. We leave the required data collection and evaluation an open task for future work.

VI. Countermeasures

We now shed light on possible countermeasures that can be applied to make the receiver location classification harder to better protect clients' location privacy. We consider countermeasures on the messenger's and on the user's side.

A. Randomizing Delivery Confirmation Times

Since timing measurements are a noisy source of information used for the attack, randomly delaying the delivery confirmation might be a viable solution to make timings to receivers in different locations harder to distinguish. While adding random delays must be implemented by messenger providers to come into effect, we can evaluate the impact of such a mechanism through a simulation based on the timing data we collected.

Timings can be perturbed by adding a delay sampled uniformly at random *between 0 and a specific maximum delay*. We systematically repeat the evaluation of the same four classification tasks (cf. Section V-G) and increase the maximum delay in every iteration by 1 second from 0 s to 20 s. Our goal is to find a threshold value that is sufficient to make the delivery confirmation timings to receivers in different locations indistinguishable. In addition, the maximum delay should be as small as possible to keep the impact on user experience low.

Figure 13 shows the overall accuracy values for four classification tasks with maximum random delays between 0 s and 20 s. We selected the same classification tasks as for the classification accuracy convergence analysis, again, to cover different types of classifications (cf. Section V-G). A maximum delay of 0 s corresponds to the original classification results. When we increase the maximum delay, the overall

classification accuracy continuously decreases and approximates the chance of randomly guessing the location, which depends on the number of location candidates. Depending on the classification task, the random guessing accuracy is reached for a maximum delay of between 5 s and 10 s, as for example for determining the network connection of receiving device *DE-22* (cf. Figure 13d). Messenger servers randomly delaying delivery confirmations by up to 6 s seems to be sufficient to render the timings indistinguishable and, thus, to disable the timing side channel in message delivery confirmations. We emphasize that there is a graceful degradation of accuracy with increasing delays – introducing maximum delays of as little as 1 or 2 seconds will already have a positive and measurable impact on users' location privacy under our attack.

If and to what extent the maximum delay can be further decreased or even flexiblized, e. g., different delays for different groups of contacts, or depending on dynamic parameters should be subject to extensive further evaluations. The best option from a user perspective would actually be the possibility to disable sending (and receiving) delivery confirmations at all – exactly as it is already offered for *read receipts* (verbatim a privacy option) in all messengers we analyzed in this paper.

B. User-side countermeasures

Users' means to reduce the effects of the timing side channel are limited, since delivery confirmations cannot be turned off in the messengers we analyzed – randomly delaying these timings can only be applied by the messenger providers. However, the use of VPN services or Tor routing all traffic through dedicated servers at distant and changing geographical locations may be a promising mitigation strategy that can be applied by users. The overhead of additional servers may perturb the delivery notifications in a similar fashion like adding random delays.

We run a small additional experiment to get a preliminary estimate of the effects of using a VPN as a countermeasure. To this end, we send messages to one receiver phone (*DE-23*) in one location (*DE-B*) both on WiFi and cellular Internet connections – in both cases connected to a US-based VPN server provided by a commercial VPN provider. Whereas without VPN, the network connection of this device can be distinguished with up to 90 % accuracy (cf. Table VI, classifications perform worse when using a VPN. For Threema (51 %) and WhatsApp (62 %), performance is hardly better than random guessing (50 %). However, for Signal, we reach a surprisingly high overall accuracy of 77 %. When repeating the same small experiment with using Tor instead of a VPN, WiFi and cellular connections can be distinguished better (Signal: 72 %, Threema: 58 %, WhatsApp: 82 %).

Without investigating these issues more systematically, we can only speculate about the reasons. One explanation could be that Signal's servers are US-based and, therefore, the routing overhead introduced by using the VPN server is too small to adequately perturb timings. For the case of Tor, the set of circuits selected in either sample may have biased the comparably small sets of timings we measured. However, since conclusive statements require more systematic and extensive measurements to allow a thorough evaluation, we leave this issue an open task for future work.

Since users' means to perturb timings and, thus, to disable the side channel seem ineffective in practice, another option could be to totally block delivery confirmations, e. g., by filtering the related packets based on their size out of their local network traffic by means of a firewall. While this might be a viable solution for technically adept users or in specifically security-sensitive use cases, it does, however, not apply to the vast majority of the 2 billion WhatsApp users.

VII. Related Work

Security of Messengers: A systematization of knowledge by Unger et al. [52] provides an extensive overview of security features in many instant messaging applications. Similarly, also other studies have analyzed security features of specific subsets of messengers and their cryptographic foundations [1], [20], [21], protocols [9], [17], [24], [42], or exploited specific features such as contact discovery to crawl millions of American phone numbers [19].

Additionally, the analysis of encrypted messenger traffic has served as a side channel to identify the language used in iMessage conversations [10], specific user actions in KakaoTalk [36], and users in various messengers [4]. Our paper complements such works in providing empirical evidence for a similar side channel under real-world conditions. Different from these works, though, our attack is conducted from one participant and directed at a specific target.

Despite such attacks, messengers specifically designed to improve the privacy of contact discovery [22] or to resist traffic analysis [53] are, however, not widely in use.

Analysis of Timings and Internet Traffic: There is a large body of work studying the connection between timings and distances and taking into account the Internet topology for the purpose of localization [6], [13], [23], [26] and distance bounding [3], [30], [37], [39] on the Internet. Our work is different, in that we do not directly relate timings to traveled distances, but instead use recurring timing characteristics to re-identify previously seen, expected locations. Similarly, traffic analysis [12], [46] is regularly used to analyze encrypted network traffic in various other domains. Purposes include website fingerprinting [27], [35], [40] or deanonymizing users and their end-to-end connections in anonymity networks such as Tor [5], [18], [28], [32], [33], [41], [43] with the most recent ones technically reaching accuracy of up to 100 % using deep learning techniques.

VIII. Conclusion

We presented a novel timing side-channel in popular instant messengers, allowing to distinguish different receivers and their locations by sending them instant messages. We have demonstrated how measuring the time between sending a message and receiving the notification that the message has been delivered enables clients to spy on each other, e. g., to determine whether or not they are at their usual location. While making use of this side channel is mostly limited to people who are in each others' contact lists and have already started a conversation before, it yet comprises an unexpected and privacy-infringing act with low technical requirements that is equally hard to detect and to mitigate for a potential victim.

Acknowledgement

This research was supported in parts by the Research Center Trustworthy Data Science and Security, one of the Research Alliance centers within UA Ruhr, and by the Center for Cyber Security at NYU Abu Dhabi. The authors would like to thank Marvin Kowalewski, Leona Lassak, Philipp Markert, Sarah Pardo, and Lena Schnitzler for their help with data collection.

References

[1] Puneet Kumar Aggarwal, P.S. Grover, and Laxmi Ahuja. Security Aspect in Instant Mobile Messaging Applications. In *Recent Advances on Engineering, Technology and Computational Sciences*, RATECS '18, Allahabad, India, February 2018. IEEE.

[2] Ryan Ariano. 'What do the check marks mean on WhatsApp?': How to determine the status of your message on WhatsApp, January 2020. https://www.businessinsider.com/what-do-the-check-marks-mean-on-whatsapp, as of October 20, 2022.

[3] Gildas Avoine, Muhammed Ali Bingöl, Ioana Boureanu, Srdjan Čapkun, Gerhard Hancke, Süleyman Kardaş, Chong Hee Kim, Cédric Lauradoux, Benjamin Martin, Jorge Munilla, et al. Security of Distance-Bounding: A Survey. *ACM Computing Surveys*, 51(5):94:1–94:33, September 2018.

[4] Alireza Bahramali, Amir Houmansadr, Ramin Soltani, Dennis Goeckel, and Don Towsley. Practical Traffic Analysis Attacks on Secure Messaging Applications. In *Network and Distributed System Security Symposium*, NDSS '20, San Diego, CA, USA, February 2020. The Internet Society.

[5] Alex Biryukov, Ivan Pustogarov, and Ralf-Philipp Weinmann. Trawling for Tor Hidden Services: Detection, Measurement, Deanonymization. In *IEEE Symposium on Security and Privacy*, SP '13, pages 80–94, San Francisco, CA, USA, May 2013. IEEE.

[6] Massimo Candela, Enrico Gregori, Valerio Luconi, and Alessio Vecchio. Using RIPE Atlas for Geolocating IP Infrastructure. *IEEE Access*, 7(1):48816–48829, April 2019.

[7] Check-host.net. API Overview. https://check-host.net/about/api, as of October 20, 2022.

[8] Heyning Cheng and Ron Avnur. Traffic Analysis of SSL Encrypted Web Browsing, 1998.

[9] Katriel Cohn-Gordon, Cas J. F. Cremers, Benjamin Dowling, Luke Garratt, and Douglas Stebila. A Formal Security Analysis of the Signal Messaging Protocol. In *IEEE European Symposium on Security and Privacy*, EuroS&P '17, pages 451–466, Paris, France, April 2017. IEEE.

[10] Scott E Coull and Kevin P. Dyer. Traffic Analysis of Encrypted Messaging Services: Apple iMessage and Beyond. *ACM SIGCOMM Computer Communication Review*, 44(5):6–11, October 2014.

[11] Nils Dalmeijer and Vlad Niculescu-Dinca. What's up with WhatsApp Neighbourhood Watch?, February 2019. https://www.leidensecurityandglobalaffairs.nl/articles/whats-up-with-whatsapp-neighbourhood-watch, as of October 20, 2022.

[12] George Danezis. The traffic analysis of continuous-time mixes. In *Privacy Enhancing Technologies Workshop*, PET '04, pages 35–50, Toronto, Canada, May 2004. Springer.

[13] Ben Du, Massimo Candela, Bradley Huffaker, Alex C Snoeren, and kc claffy. RIPE IPmap Active Geolocation: Mechanism and Performance Evaluation. *ACM SIGCOMM Computer Communication Review*, 50(1):4–10, April 2020.

[14] Chris Duckett and Steven J. Vaughan-Nichols. AWS EC2 North Virginia outage resolves but some issues linger, September 2021. https://www.zdnet.com/article/aws-ec2-north-virginia-outage-resolves-but-some-issues-linger/, as of October 20, 2022.

[15] Facebook, Inc. Data Centers, 2020. https://sustainability.fb.com/data-centers/, as of October 20, 2022.

[16] Anja Feldmann, Oliver Gasser, Franziska Lichtblau, Enric Pujol, Ingmar Poese, Christoph Dietzel, Daniel Wagner, Matthias Wichtlhuber, Juan Tapiador, Narseo Vallina-Rodriguez, Oliver Hohlfeld, and Georgios Smaragdakis. The Lockdown Effect: Implications of the COVID-19 Pandemic on Internet Traffic. In *ACM Internet Measurement Conference*, IMC '20, pages 1–18, Virtual Event, October 2020. ACM.

[17] Tilman Frosch, Christian Mainka, Christoph Bader, Florian Bergsma, Jörg Schwenk, and Thorsten Holz. How Secure is TextSecure? In *IEEE European Symposium on Security and Privacy*, EuroSP '16, pages 457–472, Saarbrücken, Germany, March 2016. IEEE.

[18] John Geddes, Rob Jansen, and Nicholas Hopper. How Low Can You Go: Balancing Performance with Anonymity in Tor. In *Privacy Enhancing Technologies Symposium*, PETS '13, pages 164–184, Bloomington, IN, USA, July 2013. Springer.

[19] Christoph Hagen, Christian Weinert, Christoph Sendner, Alexandra Dmitrienko, and Thomas Schneider. All the Numbers are US: Large-Scale Abuse of Contact Discovery in Mobile Messengers. In *Network and Distributed System Security Symposium*, NDSS '21, San Diego, CA, USA, February 2021. The Internet Society.

[20] Amir Herzberg and Hemi Leibowitz. Can Johnny Finally Encrypt?: Evaluating E2E-encryption in Popular IM Applications. In *Workshop on Socio-Technical Aspects in Security*, STAST '16, pages 17–28, Los Angeles, CA, USA, December 2016. ACM.

[21] Christian Johansen, Aulon Mujaj, Hamed Arshad, and Josef Noll. Comparing Implementations of Secure Messaging Protocols (long version). Technical report, University of Oslo, 2017.

[22] Daniel Kales, Christian Rechberger, Thomas Schneider, Matthias Senker, and Christian Weinert. Mobile Private Contact Discovery at Scale. In *USENIX Security Symposium*, USENIX '19, pages 1447–1464, Santa Clara, CA, USA, August 2019. USENIX Association.

[23] Ethan Katz-Bassett, John P. John, Arvind Krishnamurthy, David Wetherall, Thomas Anderson, and Yatin Chawathe. Towards IP Geolocation Using Delay and Topology Measurements. In *ACM Internet Measurement Conference*, IMC '06, pages 71–84, Rio de Janeiro, Brazil, October 2006. ACM.

[24] Nadim Kobeissi, Karthikeyan Bhargavan, and Bruno Blanchet. Automated Verification for Secure Messaging Protocols and Their Implementations: A Symbolic and Computational Approach. In *IEEE European Symposium on Security and Privacy*, EuroS&P '17, pages 435–450, Paris, France, April 2017. IEEE.

[25] Daniela Köhler. Department of Health Abu Dhabi In collaborations with Abu Dhabi Public Health Center launches COVID-19 Home Isolation Program Smart service on WhatsApp, July 2020.

[26] Katharina Kohls and Claudia Diaz. VerLoc: Verifiable Localization in Decentralized Systems. In *USENIX Security Symposium*, USENIX '22, Boston, MA, USA, August 2022. USENIX Association.

[27] Katharina Kohls, David Rupprecht, Thorsten Holz, and Christina Pöpper. Lost Traffic Encryption: Fingerprinting LTE Traffic on Layer Two. In *ACM Conference on Security and Privacy in Wireless and Mobile Networks*, WiSec '19, Miami, FL, USA, May 2019. ACM.

[28] Albert Kwon, Mashael AlSabah, David Lazar, Marc Dacier, and Srinivas Devadas. Circuit Fingerprinting Attacks: Passive Deanonymization of Tor Hidden Services. In *USENIX Security Symposium*, USENIX '15, Washington, DC, USA, August 2015. USENIX Association.

[29] Pam Loch. WhatsApp in the workplace, June 2019. https://www.hrmagazine.co.uk/content/features/whatsapp-in-the-workplace, as of October 20, 2022.

[30] Sjouke Mauw, Zach Smith, Jorge Toro-Pozo, and Rolando Trujillo-Rasua. Distance-Bounding Protocols: Verification without Time and Location. In *IEEE Symposium on Security and Privacy*, SP '18, pages 549–566, San Francisco, CA, USA, May 2018. IEEE.

[31] Daniel Miller, Laila Abed Rabho, Patrick Wondo, Maya de Vries, Marilia Duque, Pauline Garvey, Laura Haapio-Kirk, Charlotte Hawkins, Alfonso Otaegui, Shireen Walton, and Xinyuan Wang. *The Global Smartphone – Beyond Youth Technology*. UCL Press, London, UK, 2021.

[32] Milad Nasr, Alireza Bahramali, and Amir Houmansadr. DeepCorr: Strong Flow Correlation Attacks on Tor Using Deep Learning. In *ACM Conference on Computer and Communications Security*, CCS '18, pages 1962–1976, Toronto, Canada, October 2018. ACM.

[33] Milad Nasr, Amir Houmansadr, and Arya Mazumdar. Compressive Traffic Analysis: A New Paradigm for Scalable Traffic Analysis. In

ACM Conference on Computer and Communications Security, CCS '17, pages 2053–2069, Dallas, TX, USA, October 2017. ACM.

[34] Open Whisper Systems. Signal, May 2010. https://signal.org/, as of October 20, 2022.

[35] Andriy Panchenko, Fabian Lanze, Andreas Zinnen, Martin Henze, Jan Pennekamp, Klaus Wehrle, and Thomas Engel. Website Fingerprinting at Internet Scale. In *Network and Distributed System Security Symposium*, NDSS '16, San Diego, CA, USA, February 2018. The Internet Society.

[36] Kyungwon Park and Hyoungshick Kim. Encryption is Not Enough: Inferring User Activities on KakaoTalk with Traffic Analysis. In *International Workshop on Information Security Applications*, WISA '15, Jeju Island, Korea, August 2015. Springer.

[37] Christian Peeters, Hadi Abdullah, Nolen Scaife, Jasmine Bowers, Patrick Traynor, Bradley Reaves, and Kevin Butler. Sonar: Detecting SS7 Redirection Attacks with Audio-Based Distance Bounding. In *IEEE Symposium on Security and Privacy*, SP '18, pages 567–582, San Francisco, CA, USA, May 2018. IEEE.

[38] Michelly Purz. How Governments Worldwide are Using Messaging Apps in Times of COVID-19, March 2020. https://www.messengerpeople.com/governments-worldwide-covid-19/, as of October 20, 2022.

[39] Kasper Bonne Rasmussen and Srdjan Čapkun. Realization of RF Distance Bounding. In *USENIX Security Symposium*, USENIX '10, Washington D.C., USA, August 2010. USENIX Association.

[40] Vera Rimmer, Davy Preuveneers, Marc Juarez, Tom Van Goethem, and Wouter Joosen. Automated Website Fingerprinting through Deep Learning. In *Network and Distributed System Security Symposium*, NDSS '18, San Diego, CA, USA, February 2018. The Internet Society.

[41] Vera Rimmer, Theodor Schnitzler, Tom Van Goethem, Rodríguez Romero, Wouter Joosen, and Katharina Kohls. Trace Oddity: Methodologies for Data-Driven Traffic Analysis on Tor. In *Privacy Enhancing Technologies Symposium*, PETS '22, pages 314–335, Sydney, Australia, July 2022. Sciendo.

[42] Paul Rösler, Christian Mainka, and Jörg Schwenk. More is Less: On the End-to-End Security of Group Chats in Signal, WhatsApp, and Threema. In *IEEE European Symposium on Security and Privacy*, EuroSP '18, London, UK, April 2018. IEEE.

[43] Theodor Schnitzler, Christina Pöpper, Markus Dürmuth, and Katharina Kohls. We Built This Circuit: Exploring Threat Vectors in Circuit Establishment in Tor. In *IEEE European Symposium on Security and Privacy*, EuroS&P '21, pages 319–336, Virtual Event, September 2021. IEEE.

[44] Ismail Sebugwaawo. WhatsApp most preferred tool for UAE parents: Survey, September 2018. https://www.khaleejtimes.com/uae/whatsapp-most-preferred-tool-for-uae-parents-survey, as of October 20, 2022.

[45] Simon Sharwood. AWS US East region endures eight-hour wobble thanks to 'Stuck IO' in Elastic Block Store, September 2021. https://www.theregister.com/2021/09/28/aws_east_brownout/, as of October 20, 2022.

[46] Vitaly Shmatikov and Ming-Hsiu Wang. Timing Analysis in Low-Latency Mix Networks: Attacks and Defenses. In *European Symposium on Research in Computer Security*, ESORICS '06, pages 18–33, Hamburg, Germany, September 2006. Springer.

[47] Statista Inc. Most Popular Global Mobile Messenger Apps as of July 2021, Based on Number of Monthly Active Users , July 2021. https://www.statista.com/statistics/258749/most-popular-global-mobile-messenger-apps/, as of October 20, 2022.

[48] Qixiang Sun, Daniel R. Simon, Yi-Min Wang, Wilf Russell, Venkata N. Padmanabhan, and Lili Qiu. Statistical Identification of Encrypted Web Browsing Traffic. In *IEEE Symposium on Security and Privacy*, SP '02, pages 19–30, Berkeley, CA, USA, May 2002. IEEE.

[49] Threema GmbH. Threema, December 2012. https://www.threema.ch/, as of October 20, 2022.

[50] Threema GmbH. Security and Privacy FAQ, 2020. https://threema.ch/en/security, as of October 20, 2022.

[51] Martino Trevisan, Danilo Giordano, Idilio Drago, Maurizio Matteo Munafò, and Marco Mellia. Five Years at the Edge: Watching Internet From the ISP Network. *IEEE/ACM Transactions on Networking*, 28(2):561–574, April 2020.

[52] Nik Unger, Sergej Dechand, Joseph Bonneau, Sascha Fahl, Henning Perl, Ian Goldberg, and Matthew Smith. SoK: Secure Messaging. In *IEEE Symposium on Security and Privacy*, SP '15, pages 232–249, San Jose, CA, USA, May 2015. IEEE.

[53] Jelle van den Hooff, David Lazar, Matei Zaharia, and Nickolai Zeldovich. Vuvuzela: Scalable Private Messaging Resistant to Traffic Analysis. In *Proceedings of the 25th Symposium on Operating Systems Principles*, SOSP '15, pages 137–152, New York, NY, USA, October 2015. ACM.

[54] WhatsApp LLC. How to check read receipts. https://faq.whatsapp.com/android/security-and-privacy/how-to-check-read-receipts/, as of October 20, 2022.

[55] WhatsApp LLC. WhatsApp, January 2009. https://www.whatsapp.com/, as of October 20, 2022.

[56] Mark Williams. Secure Messaging Apps Comparison, 2021. https://www.securemessagingapps.com/, as of October 20, 2022.

APPENDIX

A. Parameter Tuning Configuration Details

The tuned parameters for each neural network type are listed below, the best performing configuration is highlighted in **bold**.

Convolutional Neural Network (CNN):

- Activation function: tanh, **relu**
- Optimizer: SGD, **Adam**, RMSProp
- Dropout rate: 0, **0.1**, 0.2, 0.3
- Number of epochs: 20, 30, 40, 50, **60**
- CNN input filters: 8, 16, **32**, 64
- Number of fully connected layers: 1, **2**, 3, 4, 5
- Number of neurons on fully connected layers: **50**, 100, 200, 500

Long Short-Term Memory Recurrent Neural Network (LSTM-RNN):

- Activation function: tanh, Sigmoid, **relu**
- Optimizer: SGD, **Adam**, RMSProp
- Dropout rate: **0**, 0.1, 0.2, 0.3
- Number of epochs: 20, 30, 40, 50, **60**
- Number of LSTM layers: 1, 2, **3**, 4, 5
- Number of LSTM units: 50, **100**, 200, 500

Stacked Denoising Autoencoder (SDAE):

- Activation function: **tanh**, Sigmoid, relu
- Optimizer: SGD, Adam, **RMSProp**
- Dropout rate: **0**, 0.1, 0.2, 0.3
- Number of epochs: 20, 30, 40, **50**, 60
- Number of encoding layers: **1**, 2, 3

B. Messenger Infrastructures

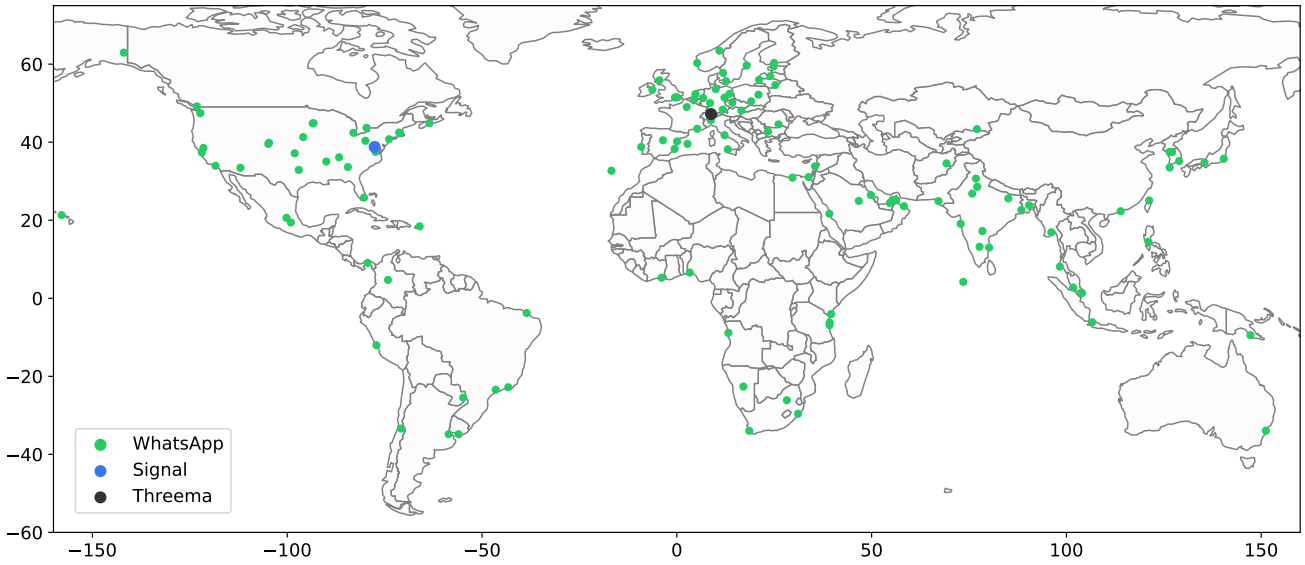


Fig. 14: Locations of Signal, Threema, and WhatsApp servers around the world. Signal is located at the US east coast, Threema is hosted in Switzerland, and WhatsApp instances are widely distributed across all continents.

C. Detailed Timing Data

More data included in the extended version available at https://arxiv.org/abs/2210.10523

1) Round 1 (Hour of day)

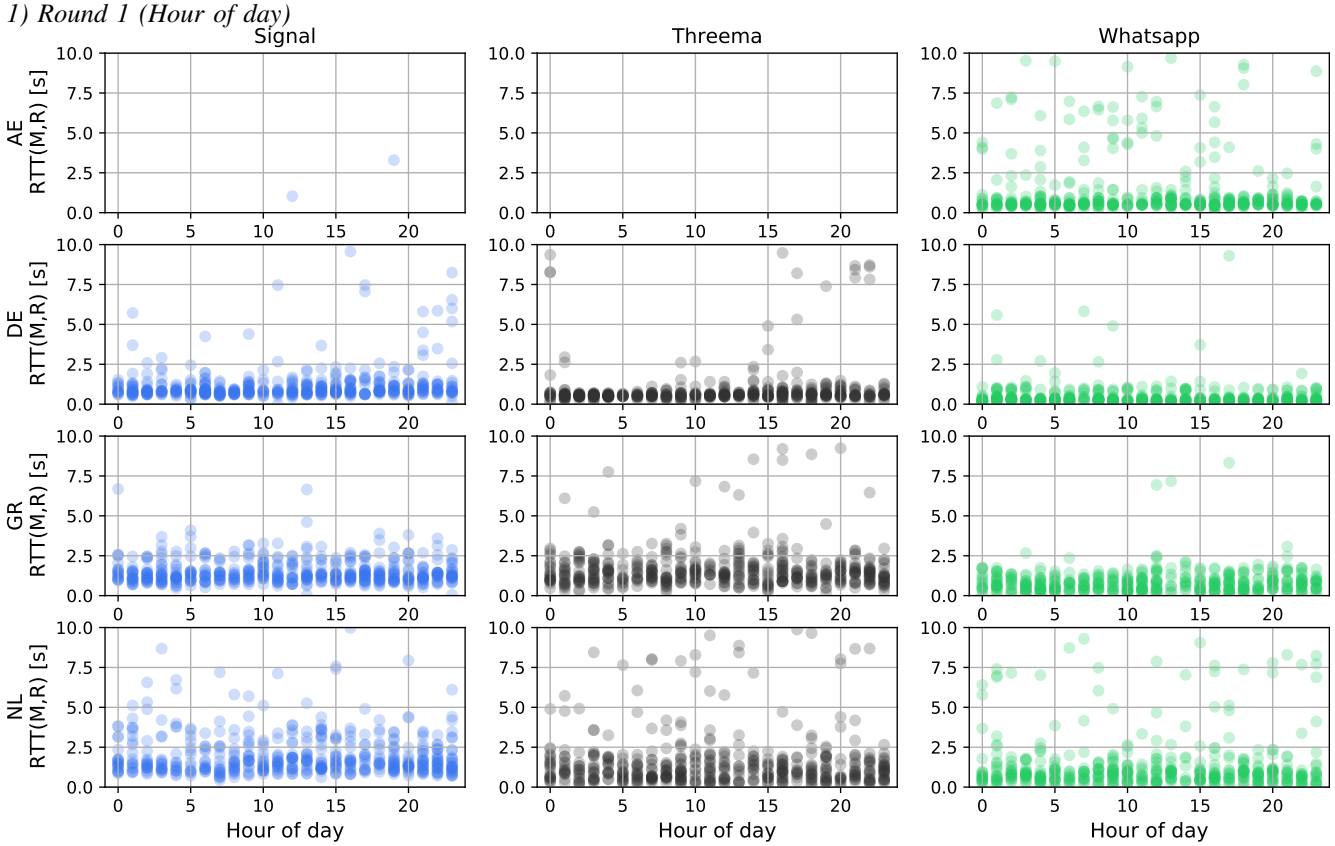


Fig. 15: Distributions of $RTT_{M,R}$ of messages sent from device *DE-11* to receivers in different countries per hour of day.

D. Detailed Classification Results

More data included in the extended version available at https://arxiv.org/abs/2210.10523

1) Round 1:

TABLE VII: Detailed classification results for the first round of measurements. Precision values for each class and overall classification accuracy values. Five values per messenger represent different notification sequence lengths.

Classification Task	Receiver Loc.	Signal					Threema					Whatsapp				
		1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
Two countries measured with sender DE-11																
DE11-2countries1	Overall Acc.	0.81	0.83	0.85	0.85	0.91	0.73	0.81	0.84	0.86	0.92	0.77	0.85	0.89	0.9	0.91
DE11-2countries1	DE	0.77	0.81	0.88	0.86	0.89	0.83	0.82	0.83	0.89	0.91	0.84	0.83	0.88	0.89	0.9
DE11-2countries1	NL	0.84	0.86	0.82	0.85	0.92	0.63	0.79	0.84	0.84	0.93	0.7	0.87	0.91	0.91	0.93
DE11-2countries2	Overall Acc.											0.82	0.85	0.85	0.86	0.95
DE11-2countries2	AE											0.88	0.9	0.87	0.91	0.96
DE11-2countries2	DE											0.76	0.8	0.84	0.8	0.94
DE11-2countries3	Overall Acc.	0.76	0.85	0.87	0.88	0.89	0.87	0.9	0.92	0.95	0.96	0.77	0.83	0.84	0.86	0.9
DE11-2countries3	DE	0.75	0.81	0.86	0.9	0.88	0.89	0.91	0.93	0.94	0.96	0.76	0.85	0.88	0.9	0.91
DE11-2countries3	GR	0.76	0.88	0.87	0.87	0.9	0.84	0.89	0.91	0.95	0.96	0.77	0.81	0.81	0.83	0.89
DE11-2countries4	Overall Acc.											0.66	0.78	0.82	0.85	0.89
DE11-2countries4	AE											0.57	0.75	0.8	0.85	0.88
DE11-2countries4	NL											0.75	0.81	0.84	0.85	0.9
DE11-2countries5	Overall Acc.	0.64	0.65	0.7	0.71	0.76	0.68	0.75	0.77	0.82	0.85	0.57	0.65	0.7	0.73	0.78
DE11-2countries5	GR	0.83	0.74	0.78	0.73	0.71	0.76	0.67	0.76	0.83	0.89	0.83	0.63	0.68	0.74	0.79
DE11-2countries5	NL	0.45	0.57	0.61	0.7	0.82	0.6	0.83	0.78	0.8	0.8	0.3	0.67	0.72	0.73	0.77
DE11-2countries6	Overall Acc.											0.66	0.79	0.83	0.86	0.92
DE11-2countries6	AE											0.56	0.77	0.81	0.86	0.9
DE11-2countries6	GR											0.77	0.81	0.86	0.87	0.93
Three countries measured with sender DE-11																
DE11-3countries1	Overall Acc.											0.61	0.74	0.77	0.78	0.87
DE11-3countries1	AE											0.52	0.74	0.73	0.84	0.88
DE11-3countries1	DE											0.75	0.73	0.75	0.72	0.88
DE11-3countries1	NL											0.56	0.76	0.83	0.78	0.85
DE11-3countries2	Overall Acc.	0.6	0.65	0.69	0.7	0.79	0.6	0.7	0.73	0.79	0.86	0.54	0.63	0.69	0.73	0.77
DE11-3countries2	DE	0.76	0.75	0.76	0.87	0.9	0.81	0.81	0.8	0.88	0.94	0.77	0.78	0.85	0.86	0.83
DE11-3countries2	GR	0.6	0.65	0.71	0.7	0.77	0.77	0.77	0.78	0.79	0.84	0.71	0.52	0.51	0.65	0.73
DE11-3countries2	NL	0.44	0.54	0.59	0.54	0.7	0.23	0.52	0.6	0.69	0.8	0.14	0.6	0.69	0.69	0.75
DE11-3countries3	Overall Acc.											0.62	0.72	0.76	0.77	0.87
DE11-3countries3	AE											0.59	0.71	0.77	0.83	0.91
DE11-3countries3	DE											0.75	0.73	0.77	0.72	0.88
DE11-3countries3	GR											0.51	0.71	0.73	0.76	0.81
DE11-3countries4	Overall Acc.											0.45	0.61	0.68	0.73	0.78
DE11-3countries4	AE											0.49	0.67	0.76	0.79	0.84
DE11-3countries4	GR											0.57	0.58	0.69	0.69	0.77
DE11-3countries4	NL											0.28	0.58	0.59	0.7	0.73
Four countries measured with sender DE-12																
DE11-4countries	Overall Acc.											0.48	0.6	0.64	0.67	0.77
DE11-4countries	AE											0.5	0.64	0.72	0.74	0.84
DE11-4countries	DE											0.73	0.71	0.7	0.72	0.81
DE11-4countries	GR											0.55	0.53	0.57	0.58	0.79
DE11-4countries	NL											0.13	0.51	0.55	0.66	0.63
Two countries measured with sender DE-12																
DE12-2countries1	Overall Acc.	0.75	0.84	0.87	0.84	0.88	0.67	0.77	0.86	0.86	0.86	0.72	0.8	0.86	0.87	0.81
DE12-2countries1	DE	0.83	0.8	0.84	0.82	0.9	0.73	0.81	0.84	0.87	0.87	0.75	0.79	0.88	0.87	0.82
DE12-2countries1	NL	0.66	0.88	0.9	0.86	0.86	0.62	0.74	0.87	0.86	0.86	0.69	0.8	0.84	0.87	0.8
DE12-2countries2	Overall Acc.											0.86	0.92	0.94	0.96	0.98
DE12-2countries2	AE											0.95	0.97	0.98	0.97	0.97
DE12-2countries2	DE											0.77	0.88	0.9	0.95	0.99

Continued on next page

TABLE VII – continued from previous page

Classification Task	Receiver Loc.	Signal					Threema					Whatsapp				
		1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
DE12-2countries3	Overall Acc.	0.59	0.62	0.62	0.6	0.67	0.63	0.77	0.79	0.86	0.78	0.76	0.81	0.84	0.82	0.85
DE12-2countries3	DE	0.73	0.77	0.68	0.6	0.67	0.75	0.76	0.76	0.86	0.78	0.84	0.83	0.87	0.88	0.87
DE12-2countries3	GR	0.44	0.46	0.57	0.61	0.67	0.51	0.78	0.83	0.86	0.77	0.68	0.78	0.81	0.76	0.84
DE12-2countries4	Overall Acc.											0.77	0.87	0.91	0.92	0.94
DE12-2countries4	AE											0.85	0.88	0.95	0.92	0.94
DE12-2countries4	NL											0.7	0.85	0.87	0.93	0.94
DE12-2countries5	Overall Acc.	0.61	0.75	0.74	0.82	0.78	0.55	0.7	0.78	0.8	0.82	0.62	0.58	0.59	0.57	0.73
DE12-2countries5	GR	0.49	0.62	0.72	0.84	0.76	0.31	0.62	0.68	0.77	0.87	0.42	0.48	0.59	0.54	0.77
DE12-2countries5	NL	0.74	0.88	0.75	0.81	0.8	0.8	0.78	0.88	0.84	0.78	0.82	0.68	0.6	0.6	0.68
DE12-2countries6	Overall Acc.											0.82	0.89	0.95	0.94	0.96
DE12-2countries6	AE											0.87	0.89	0.96	0.93	0.96
DE12-2countries6	GR											0.78	0.9	0.95	0.94	0.96
Three countries measured with sender DE-12																
DE12-3countries1	Overall Acc.											0.66	0.76	0.78	0.84	0.88
DE12-3countries1	AE											0.77	0.89	0.87	0.91	0.94
DE12-3countries1	DE											0.71	0.74	0.75	0.86	0.86
DE12-3countries1	NL											0.51	0.65	0.73	0.74	0.83
DE12-3countries2	Overall Acc.	0.46	0.5	0.49	0.54	0.62	0.45	0.57	0.69	0.72	0.69	0.57	0.59	0.62	0.61	0.71
DE12-3countries2	DE	0.76	0.64	0.44	0.69	0.73	0.72	0.68	0.73	0.72	0.7	0.71	0.76	0.77	0.79	0.77
DE12-3countries2	GR	0.01	0.17	0.4	0.27	0.53	0.13	0.37	0.53	0.67	0.68	0.43	0.48	0.52	0.57	0.7
DE12-3countries2	NL	0.6	0.68	0.63	0.66	0.61	0.51	0.67	0.8	0.77	0.68	0.57	0.52	0.57	0.47	0.64
DE12-3countries3	Overall Acc.											0.7	0.77	0.85	0.85	0.89
DE12-3countries3	AE											0.84	0.84	0.94	0.94	0.95
DE12-3countries3	DE											0.69	0.71	0.82	0.84	0.86
DE12-3countries3	GR											0.57	0.76	0.79	0.79	0.86
DE12-3countries4	Overall Acc.											0.55	0.68	0.67	0.71	0.72
DE12-3countries4	AE											0.79	0.87	0.9	0.91	0.94
DE12-3countries4	GR											0.54	0.64	0.52	0.54	0.64
DE12-3countries4	NL											0.31	0.52	0.59	0.69	0.59
Four countries measured with sender DE-12																
DE12-4countries	Overall Acc.											0.57	0.59	0.66	0.67	0.72
DE12-4countries	AE											0.83	0.83	0.87	0.93	0.94
DE12-4countries	DE											0.72	0.64	0.73	0.84	0.77
DE12-4countries	GR											0.45	0.55	0.53	0.52	0.64
DE12-4countries	NL											0.3	0.35	0.48	0.4	0.53
Two countries measured with sender GR-11																
GR11-2countries1	Overall Acc.						0.59	0.61	0.66	0.68	0.68	0.75	0.84	0.83	0.87	0.91
GR11-2countries1	DE						0.85	0.91	0.7	0.8	0.53	0.8	0.86	0.8	0.84	0.86
GR11-2countries1	NL						0.32	0.3	0.62	0.55	0.83	0.7	0.81	0.87	0.9	0.97
GR11-2countries2	Overall Acc.											0.88	0.96	0.96	0.96	0.98
GR11-2countries2	AE											0.88	0.99	0.98	0.97	0.99
GR11-2countries2	DE											0.88	0.93	0.93	0.95	0.98
GR11-2countries3	Overall Acc.											0.68	0.79	0.86	0.89	0.95
GR11-2countries3	AE											0.87	0.92	0.92	0.91	0.95
GR11-2countries3	NL											0.48	0.66	0.8	0.86	0.95
Three countries measured with sender GR-11																
GR11-3countries	Overall Acc.											0.62	0.74	0.82	0.83	0.9
GR11-3countries	AE											0.81	0.88	0.93	0.87	0.95
GR11-3countries	DE											0.88	0.8	0.8	0.83	0.89
GR11-3countries	NL											0.18	0.54	0.74	0.79	0.88