FBS-Radar: Uncovering Fake Base Stations at Scale in the Wild

Zhenhua Li [1], Weiwei Wang [2], Christo Wilson [3], Jian Chen [1], Chen Qian [4],
Taeho Jung [5], Lan Zhang [6], Kebin Liu [1], Xiangyang Li [6], Yunhao Liu [1]
[1] *Tsinghua University* [2] *Baidu Mobile Security* [3] *Northeastern University* [4] *UCSC*
[5] *Illinois Institute of Technology* [6] *University of Science and Technology China*
{*lizhenhua1983, ww.wang.cs*}*@gmail.com, cbw@ccs.neu.edu, softwarecj@163.com, cqian12@ucsc.edu*
tjung@hawk.iit.edu, {*zhanglan03, liukebin2006*}*@gmail.com, xiangyangli@ustc.edu.cn, yunhaoliu@gmail.com*

***Abstract*—Base stations constitute the basic infrastructure of today's cellular networks. Unfortunately, vulnerabilities in the GSM (2G) network protocol enable the creation of *fake base stations* (FBSes) that are not authorized by network operators. Criminal gangs are using FBSes to directly attack users by sending spam and fraud SMS messages, even if the users have access to 3G/4G networks. In this paper, we present the design, deployment, and evolution of an FBS detection system called *FBS-Radar*, based on crowdsourced data of nearly 100M users. In particular, we evaluate five different metrics for identifying FBSes in the wild, and find that FBSes can be precisely identified without sacrificing user privacy. Additionally, we present a novel method for accurately geolocating FBSes while incurring negligible impact on end-user devices. Our system protects users from millions of spam and fraud SMS messages per day, and has helped the authorities arrest hundreds of FBS operators.**

I. Introduction

Base stations (BSes), also known as base transceiver stations (BTSes) or cell towers, constitute the basic infrastructure of today's cellular networks. They connect end-user cellular devices to a wide-area network (*e.g.,* a mobile carrier network and the Internet) by forwarding voice streams, short message service (SMS) messages, and IP data packets. At present, millions of 2G/3G/4G and hybrid-mode BSes co-exist all over the world, serving billions of mobile devices. Recent trends suggest that cellular networks will be the dominant access method for the Internet in the near future; in fact, they already are in many parts of the world [1], [2].

Unfortunately, vulnerabilities in the GSM (2G) network protocol enable the creation of *fake base stations* (FBSes) that are not authorized by network operators. Specifically, the GSM standard does not require that devices authenticate BSes [3]. Furthermore, typical cellular devices support 2G, 3G, and 4G networks, and in the presence of multiple available networks, tend to choose the one with the highest signal strength [4]. This allows unauthorized third-parties to set up their own high signal-strength 2G cell towers, and nearby clients are likely to attach to them. Attackers can even send jamming signals that force nearby GSM-compatible 3G/4G cellphones down to the GSM mode [5], [6]. FBSes are not hypothetical: to date, they have been observed in the US, China, India, Russia, Israel, and the UK [7], [8], [9], [10]. Although many mobile carriers are planning to abandon GSM, it will take years to upgrade cell towers [7] and phase out legacy end-user devices.

Criminal gangs are now using FBSes to directly attack users. Using an FBS, an attacker can send SMS messages to users from spoofed phone numbers, including privileged numbers associated with mobile carriers, government agencies, public services, banks, *etc.* These messages can contain spam advertisements, phishing links, and solicitations for high-fee premium services. In China alone, users received over 2.9B (B = billion), 4.2B, and 5.7B spam/fraud messages from FBSes in 2013, 2014, and 2015, respectively, causing estimated losses of billions of dollars [11], [12], [13]. Surprisingly, an attacker with a $700 FBS that is small enough to mount inside a car [4] can earn up to $1400 a day [14].

Although government agencies have undertaken efforts to detect and take down FBSes, these efforts are hampered by several challenges. For example, in China the Ministry of Public Security (MPS) and mobile network operators have attempted to detect FBSes by deploying static and mobile sensors in metropolitan areas that scan for BSes with unexpectedly high signal strengths. However, this approach incurs huge infrastructure costs and covers limited geographic areas. Additionally, the MPS and mobile carriers encourage users to report suspicious BS signals and SMS messages. Unfortunately, this crowdsourced detection approach yields poor results, since users are not trained to identify and report FBSes.

In this paper, we present the design, deployment, and evolution of an FBS detection system called *FBS-Radar*. The client side of FBS-Radar is integrated into Baidu PhoneGuard [15], a mobile security app available on Android and iOS. PhoneGuard is currently used by over 100M (M = million) users, mostly in China. The goals of FBS-Radar are fourfold:

1) FBS-Radar should detect as many FBSes as possible with few false positives, based on little ground truth (since we have no insight into criminals' activities), and without any specialized hardware.
2) To protect users, FBS-Radar should automatically filter spam and fraud SMS messages sent by FBSes

NDSS '17, 26 February - 1 March 2017, San Diego, CA, USA
 ISBN 1-891562-46-0
http://dx.doi.org/10.14722/ndss.2017.23098

from users' devices, with a high precision.

3) To aid law enforcement agencies, FBS-Radar should provide actionable intelligence about the physical locations of FBSes, so that they can be taken down.

4) To encourage adoption, FBS-Radar should use minimal resources on the client side (*e.g.,* CPU, traffic, and battery), minimize collection of sensitive data (to preserve user privacy), and not require root privileges.

Since beginning this project, we have learned a great deal about FBSes, and made major changes to FBS-Radar as a consequence. For ease of exposition, we will discuss the *alpha* and *beta* phases of FBS-Radar's development as follows.

Alpha Phase. When we released FBS-Radar in Aug. 2014, the goal was to explore whether it was possible to detect FBSes automatically, with high precision and accuracy, based on data passively collected from end-user devices. To this end, we asked PhoneGuard users to opt-in to the following data collection process: whenever a *suspicious* SMS message arrived at their device, the message and accompanying meta-data were sent to a cloud service maintained by Baidu, as well as information about the device's recent BSes (*e.g.,* IDs and signal strengths) and nearby WiFi Access Points (*e.g.,* MAC addresses). We define an SMS message to be *suspicious* if the sender's number is not in the recipient's contact list, or the number is an *authoritative number* used by a mobile carrier, a government agency, a public service, a bank, and so forth.

Throughout 2015, FBS-Radar examined a total of 70B SMS messages, of which 6.4B (9.1%) were labeled as suspicious and collected for further analysis on the cloud side. Using this data, we comprehensively explored different metrics/methods for detecting FBSes, ultimately isolating five sets of SMS messages that are highly likely to have been sent by FBSes:

- Set-1: 0.23% of suspicious messages were sent by BSes with unreasonably high-strength cellular signals (> -40 dBm versus -113 dBm to -51 dBm for legitimate BSes; see § V-A).
- Set-2: 0.15% of suspicious messages were sent by BSes with invalid IDs that violate the syntax of legitimate BS IDs (see § V-B).
- Set-3: 0.16% of suspicious messages came from authoritative phone numbers and were determined to contain fraudulent text content (*e.g.,* phishing) using a bag-of-words SVM classifier trained on 200,000 hand-labeled SMS messages (see § V-C).
- Set-4: 4.1% of suspicious messages were sent by BSes that were not in their correct geolocation, *i.e.,* they were spoofing the ID of a legitimate, but distant, BS. We determined this by geolocating the user based on nearby WiFi APs, and comparing the user's location to authoritative coverage maps of BSes provided by mobile carriers (see § V-D).
- Set-5: 0.39% of suspicious messages were sent by BSes that were also in incorrect geolocations. In this case, we use BS-handover speed to estimate the user's geolocation when WiFi AP data is too sparse to be accurate (see § V-E).

FBS-Radar conducts all analysis on the cloud side (employing around 30 commodity servers), so the client-side resource consumption is negligible.

Taking the union of all five sets, we found that over 0.8M SMS messages were sent to users by FBSes each day in our dataset, covering 4.7% of all messages marked as suspicious. Furthermore, we found that the union of Set-1, 2, 4, and 5 had over 98% overlap with Set-3. This was a critical observation, since it meant that **FBS-Radar did not need to collect the text content of SMS messages to identify FBSes**.

Beta Phase. In Jan. 2016, we deployed an updated and more feature-complete version of FBS-Radar. The most major change is that we automatically opted-out 99% of FBS-Radar users from the collection of suspicious SMS message content. We continue to collect the text content of suspicious SMS messages from the remaining 1% of users to conduct A/B testing of new features, although we eventually plan to opt these users out as well. As a result of this privacy-friendly change, many more PhoneGuard users have enabled FBS-Radar, causing the number of suspicious SMS messages flagged by the system to increase from 17.5M per day in 2015 to 32M in 2016.

FBS-Radar protects users from FBS-originated SMS messages by quarantining them in an "FBS message folder". After the client-side app forwards a suspicious SMS message to the cloud, the cloud-side software analyzes the message meta-data to determine if it was sent from an FBS. If so, the client-side app quarantines the SMS message and notifies the user. FBS-Radar allows users to manually flag messages in the "FBS message folder" as valid, indicating that the system produced a false positive. We observe that only 0.05% of flagged messages are marked as valid by users, indicating that FBS-Radar has a very low false positive rate.

Using crowdsourced data, FBS-Radar is able to quickly provide accurate location estimates of FBSes to law enforcement (see § VII). We use a novel method to pinpoint FBSes with a median (mean) accuracy of 11 meters (149 meters). This achievement is non-trivial, since FBSes frequently move and change their IDs, so we must take temporal and spatial locality into account. Overall, we estimate that there are hundreds of active FBSes around China at any point in time. Between Jul. 2015 and Jun. 2016, the police were able to arrest 455 FBS operators and take down 1109 FBSes thanks to the data provided by FBS-Radar.

Finally, given the popularity of FBS-Radar, it is possible that FBS attackers may attempt to avoid our detection methods. However, to avoid the detection of FBS-Radar, attackers must adopt strategies that decrease their ability to conduct attacks. First, an attacker may be able to avoid detection by reducing the signal strength of their FBS. Nonetheless, this reduces the effectiveness and radius of their attack, and consequently reduces the attacker's income. Second, an attacker may avoid detection by choosing a BS ID for their FBS that corresponds to a nearby legitimate BS tower. But in this case, user devices will maintain their existing connections to the legitimate BS tower, rather than switch over to the FBS. Both of these outcomes are positive for users, and support our goal of making FBS-Radar an effective deterrence against FBS attacks.

Contributions. In summary, our work on FBS-Radar makes the following contributions:

- Using extensive crowdsourced data, we evaluate five different metrics for identifying FBSes in the wild, and find that FBSes can be precisely identified without sacrificing user privacy.
- We present a novel method for accurately geolocating FBSes based on crowdsourced measurements. Our method only relies on the data about the geolocations of WiFi APs, and therefore has negligible impact on end-user devices' batteries.
- As of Jun. 2016, FBS-Radar is in use by over 92M people. Our system protects users from millions of spam and fraud SMS messages per day, and has helped the authorities arrest hundreds of FBS operators.

Limitations. Despite its large-scale deployment and solid real-world impact, FBS-Radar bears two-fold limitations. First, as mentioned before, we have little ground truth about FBSes, thus limiting the recall rate of our FBS and spam/fraud SMS message detection. We do not know how many FBSes have evaded the detection of FBS-Radar, nor do we know the structure of the criminal organization(s) behind FBS attacks.

Second, though FBS-Radar is effective in detecting FBSes that send spam/fraud SMS messages, it cannot detect FBSes that do not send SMS messages. The latter are exemplified by surveillance devices like IMSI-catchers [16], [17], [18]. We focus on spamming devices since they are a large-scale problem in China, and there have been few public reports of illegal IMSI-catcher attacks in China. Although we have not evaluated the use of FBS-Radar in detecting surveillance attacks, many of these devices actively interrogate user devices [19], which suggests that some of the features used by our system (*e.g.,* crowdsourced measurements of BS IDs, cell tower locations, and signal strengths) could be useful towards detecting these devices. We leave this investigation as future work.

Consequently, we view FBS-Radar as a first step towards practically addressing the threat posed by FBSes. Our results provide a first-of-its-kind window into the activities of criminals that exploit FBSes to attack users.

II. Background and Related Work

In this section, we first introduce the operating principles of a typical FBS. Next, we present examples of spam and fraud SMS messages sent by FBSes, and discuss state-of–the-art FBS detection approaches that are used in practice. Finally, we review related work on detecting and frustrating FBSes.

A. Operating Principles of a Typical FBS

It is reported that the system architecture of FBSes has evolved over several generations [20], where each generation uses simpler and less costly hardware and communication processes. Below, we explain the operating principles of a widely reported type of FBS [4].

Hardware and Software. As shown in Fig. 1 (a), an FBS is generally made up of three components: 1) a GSM wireless transceiver, 2) an engineering laptop, and 3) an engineering cellphone. The three components are portable and require a relatively small amount of electrical power, making it easy to accommodate them in a van, a minibus, the trunk of a car, or even a backpack. The mobility of modern FBSes makes them difficult to localize and take down.

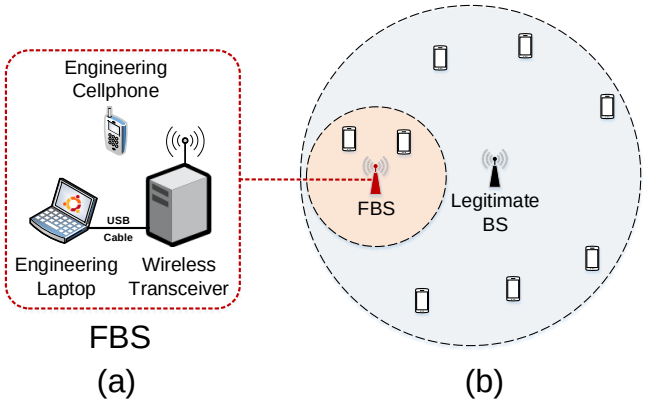

Fig. 1. Operating principles of a typical FBS. Compared with a legitimate BS, an FBS usually possesses a smaller signal coverage area but a higher signal strength.

The GSM wireless transceiver is the core component of an FBS, mainly consisting of a main board, a GSM radio frequency duplexer, an external antenna, a signal power amplifier, and a power supply. These devices can simulate most of the BS-to-cellphone functions of a legitimate GSM base station, but not the BS-to-carrier functions (*i.e.,* it cannot forward voice, SMS, and data packets to a mobile carrier's network). These devices typically include support for BCCH (Broadcast Control Channel) allocation, location updating, identification requests, user identification (particularly IMSI and IMEI) acquisition, SMS message sending, and so on.

The engineering laptop is usually connected to the wireless transceiver via a USB cable, through which it controls the transceiver, *e.g.,* tuning the radio frequency, adjusting the signal strength, setting the BS ID, and forging the sender's phone number. The FBS operator typically installs a suite of GUI-driven control software onto the laptop, to facilitate sending commands to and observing statistics from the transceiver.

The engineering cellphone is employed to search for nearby, legitimate BSes and record their wireless parameters (*e.g.,* their BS IDs). Based on this information, the operator can reconfigure the parameters of the FBS to mimic legitimate BSes, so as to maximize the number of affected users.

Communication Process. When an FBS operator wants to send SMS messages to nearby user devices, s/he first uses the engineering cellphone to find the closest legitimate BS, as depicted in Fig. 1 (b). Suppose this legitimate BS is using the BCCH radio frequency f_0 and identifier ID_0, and the signal strength perceived by the engineering cellphone is s_0 (s_0 usually lies between -113 and -51 dBm). The FBS operator will configure their transceiver to reuse frequency f_0 or use some other frequency f_1, and set a different identifier ID_1.

In order to interrupt the existing connections between nearby user devices and legitimate BSes, the FBS must satisfy

at least two requirements. First, ID_1 must contain a different LAC code (Location Area Code, as explained in § V-B) compared with ID_0, so that nearby user devices might (but not necessarily) believe that they have entered a new cellular coverage area. However, for energy concerns, user devices are generally insensitive to a small BS ID difference [3] — in this case, they may well preserve existing connections unless the connectivity is intolerably poor. As a consequence, most FBS operators would select BS IDs (more specifically, LAC codes) that are significantly different from those of nearby legitimate BSes, to make the FBS affect as many users as possible and maximize their profits. Second, following the same economic motivation, the operator would adjust the signal strength of the FBS to guarantee that it offers a much higher signal strength ($s_1 \gg s_0$) than the legitimate BS.

Once the operator has properly configured the FBS, the following sequence of events begin to occur, culminating in SMS messages being sent to all users in close proximity:

- The FBS broadcasts its system information using the configured BCCH radio frequency.
- When a user device receives the system information of the FBS, it detects it as a new BS in a new coverage area with a higher signal strength than alternate BSes. Thus, the user device sends a Location Updating Request to the FBS.
- On receiving the Location Updating Request, the FBS first sends an Identity Request to the user device to acquire its IMSI information, and then sends another Identity Request to acquire its IMEI information. When both types of information are obtained, the FBS returns a Location Updating Accept to it.
- Subsequently, the FBS sends an SMS message to the user device using a spoofed phone number. This action can be repeated multiple times.
- When the FBS finishes sending messages, it cuts off the cellular connections with its user devices by lowering its signal strength, changing its BS ID, or simply shutting down the signal. After that, the affected user devices will re-connect to a legitimate BS.

B. Spam and Fraud Messages Sent by FBSes

FBSes can send spam and fraud SMS messages to connected user devices from arbitrary (spoofed) phone numbers. In this work, we define *spam* as messages containing advertisements. Example spam sent from FBSes include:

> "*We are selling excellent, cheap goods and food from Jul. to Aug. 2016. Visit our shops at the People's Square as soon as possible!*" – sent from a (usually not well-known) mart or grocery.

> "*We provide very cheap and legal invoices that can help you quickly make a big fortune. Don't hesitate, dial us via the phone number: 010-12345678!*" – sent from a (usually not well-known) company.

In contrast, we define *fraud* as messages that attempt to maliciously deceive users. Fraud messages often cause severe losses to mobile users, in violation of the law. Typical examples of fraud messages include:

> "*Dear user, you are lucky to be the winner of this month's big award! You will be offered 10-GB FREE 4G traffic by clicking on this URL: http://www.10086award.com.*" – sent from China Mobile (10086). If the user clicks this URL, they are taken to a page that attempts to phish their account credentials. Some URLs also lead to pages containing drive-by download attacks.

> "*Dear customer, you have failed to pay for this year's management fee of 100 dollars. If you do not pay for it before Jul. 30th, you will face a fine of 500 dollars. You should pay it by transferring money to the following bank account: 0000000000123456789.*" – sent from a well-known bank.

We present quantitative results on the spam and fraud messages sent by FBSes in our dataset in § V-C.

C. State-of-the-Art Detection Approaches

It is well-known that FBSes use a much higher signal strength than legitimate BSes [4], [20]. Guided by such knowledge, the MPS and mobile carriers in China have adopted signal-based approaches to detect FBSes. Three methods have been deployed in practice:

Static Electronic Fence. The MPS of China often deploys a static "electronic fence" within a specific geographic area to capture cellular signals from FBSes [4]. The basic units of the electronic fence can be a number of networked cellular signal sensors, or a number of low-power feature cellphones [21], typically deployed at street corners. While static fences can be effective tools, they incur high infrastructure costs, and it is not feasible to deploy them at scale across whole cities, let alone whole countries.

FBS-signal Detection Car. Mobile network operators in China employ dedicated FBS-signal detection cars to patrol along major streets [22]. Unfortunately, this random-walk method is unable to provide detection coverage over a large area. In addition, FBSes can easily be moved away from major streets or simply shut down when approaching major streets.

Active User Reporting. Both the MPS and mobile carriers in China encourage users to actively report suspicious BS signals and SMS messages, *e.g.,* by dialing certain phone numbers like `12321` [23]. However, the vast majority of mobile users in China do not realize the existence of FBSes, making this detection method ineffective.

Comparison with FBS-Radar. Like these signal-based approaches, FBS-Radar also makes use of signal-strength information and user reports. However, there are two fundamental differences. First, we do not use signal information as the major, or only, factor for FBS detection. As we show in § V-A, signal-strength examination can only detect a small portion ($4.9\% = 0.23\%/4.7\%$) of all FBS messages. Second, as we discuss in § VI, FBS-Radar only uses active user engagement to detect misclassified FBS messages, *i.e.,* false positives.

All other data is collected passively by the client-side app after users opt-in. Thus, FBS-Radar achieves wide detection coverage and quick responsiveness all in one system.

D. Other Related Work

In academia, a few preliminary technologies have been recently proposed to detect or frustrate FBSes by monitoring BSes on the carrier and client sides.

Like the static electronic fence method, Dabrowski *et al.* propose to deploy a network of signal measurement units in a geographical area, constantly scanning frequency bands and fingerprinting network parameters of nearby BSes [24]. Do *et al.* propose to utilize machine learning-based anomaly detection of carrier contexts to discover FBSes [25]. Unfortunately, neither study addresses the practical issue of how to scale the detectors to cover large areas.

To mitigate the attacks from FBSes, Broek *et al.* suggest replacing the IMSIs of user devices with changing pseudonyms [26]. The key limitation of this method is that it requires changes to the SIM (Subscriber Identity Module) and the authentication server, at great cost to carriers and users.

SRLabs developed two client-side tools for FBS detection, called SnoopSnitch [27] and CatcherCatcher [28]. SnoopSnitch is an Android app that warns users about FBS threats by leveraging in-depth and fine-grained analysis of received cellular signals. However, SnoopSnitch only works on Qualcomm-based Android phones and requires root privileges. Similarly, CatcherCatcher attempts to identify irregularities in mobile networks to detect FBS activity, but it only works on Osmocom phones. Lastly, the ongoing Android IMSI-Catcher Detector [29] (AIMSICD) project attempts to detect FBSes through a variety of client-side heurstics, *e.g.,* monitoring signal strength, checking BS information consistency, and so forth. Nevertheless, AIMSICD is still an alpha-version Android app, and its effectiveness in practice has not been evaluated.

Comparison with FBS-Radar. FBS-Radar is conceptually similar to many of these efforts, in that it turns end-user devices into a crowdsourced network of sensors. However, FBS-Radar is able to operate at scale (as of 2016, it is used by 92M users) since it runs on generic, non-rooted Android and iOS devices.

III. Threat Model

Before describing the design of FBS-Radar, we first discuss the threat model under which we are operating.

In this work, our goal is to detect SMS messages sent by, and the geographic locations of, FBSes. We assume that these FBSes are operated by *active attackers* who attempt to disseminate spam and fraud SMS messages. We make no assumptions about the FBSes' underlying technology (*i.e.,* they can use any network protocol (2G/3G/4G), frequency, *etc.*) and we assume that FBSes may spoof identifiers (*i.e.,* BS IDs and phone numbers). Furthermore, we assume that FBSes may change locations, and go on and offline at any time.

We make no attempt to detect *passive eavesdropping* on phone calls, SMS messages, and data packets. Passive eavesdropping is mainly conducted using IMSI-catchers [16], [17], [18], commonly known as "stingrays". Although these devices are controversial, they also have legitimate law enforcement uses [19], [30], [31], [32]. Besides, as mentioned at the end of § I, illegal IMSI-catcher attacks have rarely been reported in China. For the above reasons, we leave the detection of passive attackers as future work.

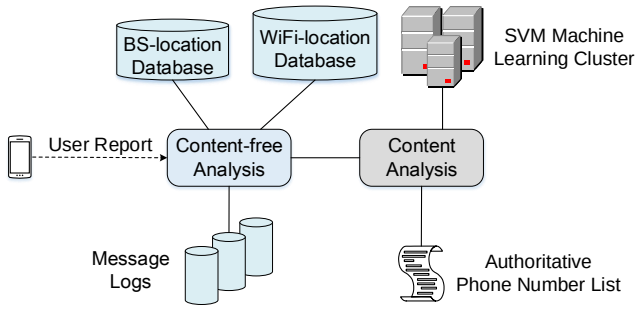

Fig. 2. Architectural overview of FBS-Radar.

IV. Data Collection

Fig. 2 depicts the system architecture of FBS-Radar, made up of three functional components (Content-free Analysis, Content Analysis, and SVM Machine Learning Cluster) and four data components (*i.e.,* Authoritative Phone Number List, BS-location database, WiFi-location database, and Message Logs). In this section, we present how each data component collects and parses its concerned data. We will explain how the three functional components work in § V and § VII .

User Reports. As mentioned in § I, FBS-Radar is implemented as a component of Baidu PhoneGuard, and leverages the platform provided by PhoneGuard to collect data from users. Once a user opts-in to data collection, the client-side mobile app forwards *suspicious* SMS messages and associated device meta-data to the cloud (recall that we define SMS messages as suspicious if the sender is not in the recipient's contact list, or the sender is using an authoritative phone number given on a predefined list). When a *user report* arrives at the cloud, it is first processed by the functional components and extended with additional meta-data from the data components, and finally backed up into the Message Logs storage cluster.

The mobile app submits user reports in JSON (JavaScript Object Notation) format. We choose to use JSON because it is easy to read and analyze for both humans and machines. In detail, each user report includes the following fields:

- Reception time of the suspicious SMS message (t_1) in the UNIX time format.
- Perceived signal strength (s_1) and BS identifier (ID_1) of the device's current BS;
- Perceived signal strengths (s_2, s_3) and identifiers (ID_2, ID_3) of the two previously connected BSes, coupled with two timestamps (t_2, t_3) when the signal strengths of the two BSes were measured by the mobile app;
- Sender's phone number, and content of the SMS message (note that as of Jan. 2016, 99% of FBS-Radar users no longer submit SMS message content);

TABLE I. AN EXAMPLE USER REPORT.

Field	Value
t_1	1452869570549
s_1	-79 dBm
ID_1	460-00-39185-21492
t_2	1452865343627
s_2	-84 dBm
ID_2	460-00-39185-52921
t_3	1452865278412
s_3	-95 dBm
ID_3	460-00-39185-52112
Sender's phone number	+86-135-5281-9836
Content of the message	$\leq$ 140 characters
MAC_1	ec:26:ca:26:f6:c0
MAC_2	d0:c7:c0:aa:6a:fc
...	...
MAC_n	6a:3e:34:03:d8:13

- MAC address (MAC_1) of the WiFi AP connected to the user device, as well as the MAC addresses (MAC_i, $i \in \{2, \cdots, n\}$) of the other $n-1$ WiFi APs perceived by the user device. FBS-Radar uses these MAC addresses to enhance the accuracy of WiFi localization (as explained in § VII-A). We do not collect users' GPS coordinates, since most users turn this feature off to preserve battery life. Meanwhile, we note that for those users who turn the GPS feature on, their GPS coordinates could potentially help FBS-Radar achieve better localization. Inevitably, this will increase the complexity of our system design.

An example user report is shown in Table I. Note that the user report does not include the recipient's phone number. We do not collect this information to help preserve users' privacy.

Authoritative Phone Number List. FBSes often send spam or fraud messages using spoofed authoritative phone numbers, such as those used by mobile carriers, public services, and banks. Although *real* authoritative phone numbers are sometimes used to broadcast advertisements, they never send fraud messages to intentionally cheat users. Therefore, a fraud message coming from an authoritative phone number is a clear signal that an FBS sent the message.

We collaborate with both the MPS and major mobile carriers of China to maintain an authoritative phone number list. This list is updated on a monthly basis, and its latest version (in Feb. 2016) contains 1446 phone numbers.

BS-location Database. FBSes have the capability to spoof arbitrary BS IDs. To help distinguish legitimate from spoofed BS IDs, we need a database of legitimate BS IDs and their respective coverage areas.

The mobile network operators in China provide us with a confidential database of all BSes in China, and their coverage areas. This database is updated on a weekly basis. When the BS-location database is queried with a valid BS ID, it returns a four-tuple $< lat,\ lon,\ radius,\ tag >$, where lat denotes the latitude and lon denotes the longitude of the BS; $radius$ is the *theoretical* signal coverage radius of the BS (measured in a laboratory environment); and tag represents whether the queried BS ID can be found in the database. Hence, the signal coverage area of the BS is roughly taken as $\pi \times (radius)^2$ (as demonstrated in Fig. 1 (b)). The values of lat and lon are accurate to six decimal places, *e.g.,* $lat = 24.800947$ and $lon = 113.598193$. That is to say, the localization accuracy of a BS is about 0.1 meter in theory. However, according to the mobile carriers who provide the BS-location database, the localization error of a BS can be up to 10 meters in practice.

WiFi-location Database. To geolocate users, each WiFi MAC address in a user report must be mapped to a geographic location $(lat,\ lon)$. We perform this mapping using a nationwide WiFi-location database maintained by Baidu. When the WiFi-location database is queried with a MAC address, it returns a three-tuple $< lat,\ lon,\ tag >$, where tag represents whether the queried MAC address can be found in the database. The values of lat or lon are accurate to six decimal places. According to Baidu, the localization error of a single WiFi AP can reach tens of meters, *i.e.,* the WiFi-location database is less accurate than the BS-location database.

Different from the BS-location database, the WiFi-location database does not provide the signal coverage area of a WiFi AP because of two issues. First, (1) most WiFi APs do not provide this information. Second, (2) the signal coverage area of a WiFi AP is highly unstable, as it is significantly influenced by the environment [33]. Advanced WiFi APs can even adaptively re-scale the signal coverage area by tuning the signal strength [34]. Accordingly, the WiFi-location database does not provide the signal strength information of a WiFi AP. Furthermore, the WiFi-location database is updated on a daily basis (more frequently than the updating of the BS-location database) for two reasons. First, (1) the number of WiFi APs (∼450 M in our current WiFi-location database) is much larger than that of BSes (∼15 M in our current BS-location database). Second, (2) WiFi APs are more dynamic than BSes. For example, a WiFi AP can be moved to several different places in one day, while moving a legitimate BS to multiple (> 2) different places in one week is almost impossible. Overall, we observe that between 71–75% of the WiFi MAC addresses in user reports are present in our WiFi-location database.

Message Logs. Each user report is extended by FBS-Radar to a full-fledged message log using data from the the authoritative phone number list, BS-location database, and WiFi-location databases. In addition to the original fields in the user report, a message log also includes a field $isAuth$ to indicate whether the sender's phone number is authoritative, the geographic location and signal coverage radius of the three most recent BSes, the geographic location all nearby WiFi APs. As of Jun. 30th, 2016, we have archived more than 600 days of message logs, corresponding to petabytes of data. To facilitate our research in this paper, we select the message logs during one week (between Sep. 15–21, 2015) as the studied dataset, referred to as MsgLogs. We select this dataset because it contains abundant (122M) message logs with complete information (particularly the text content of every message).

Ethics. Throughout this project, we took the utmost care to protect users of FBS-Radar and their sensitive data. Although Baidu does not have an internal Institutional Review Board (IRB), we adopted fundamental ethical principals throughout this project, *i.e.,* obtaining informed consent from users, and

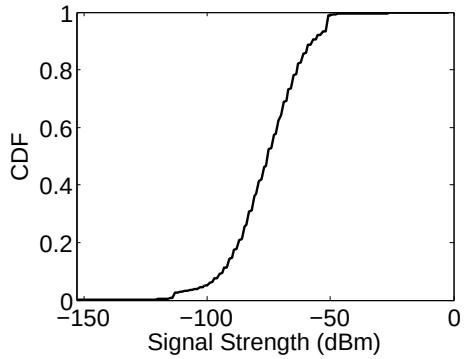

Fig. 3. Distribution of user-perceived signal strengths recorded in MsgLogs.

causing no harm to PhoneGuard users. First, users must choose to install PhoneGuard, and explicitly opt-in to data collection. Users are clearly informed about what data will be collected, and they are free to opt-out at any time in the PhoneGuard settings (or by uninstalling the PhoneGuard app). Second, the MsgLogs dataset used in our experiments was securely stored on Baidu servers, and at no time did user data leave Baidu's systems. We are cognizant of the fact that FBS-Radar collects sensitive data from users, most notably the content of suspicious SMS messages. As we describe in § V-F, we found that SMS message content is not necessary to identify FBS messages, and thus we opted 99% of FBS-Radar users out of this data collection in Jan. 2016.

V. Identifying FBS Messages

In § IV, we describe the large-scale, comprehensive datasets that are available to FBS-Radar. In this section, we investigate five different methods for identifying SMS messages sent by FBSes (which we simply refer to *FBS messages*). This corresponds to the "alpha" phase of FBS-Radar's deployment. We describe each method in detail, and summarize with high-level findings based on our analysis of identified FBS messages.

A. Signal Strength Examination

As mentioned in § II, unreasonably high signal strength is the most obvious characteristic of an FBS. This motivated us to try and use signal strength information as our first method to identify FBS messages. Fig. 3 illustrates the distribution of all user-perceived signal strengths recorded in MsgLogs. The signal strengths lie between -153 dBm and -1 dBm, and over 95% of signal strengths are between -113 dBm and -51 dBm (*i.e.,* the common range of user-perceived signal strengths).

For a legitimate BS, the maximum signal strength perceived by a user device can reach -40 dBm if the user device is placed just below the BS [3]. As a consequence, any message with a signal strength higher than -40 dBm is highly likely to have come from an FBS. Guided by this rule, we find that 0.23% of suspicious messages have a signal strength higher than -40 dBm, We refer to these FBS messages as Set-1. Note that -40 dBm is a very conservative threshold, and thus we expect many false negatives. For example, a device that is relatively far from an FBS will observe signal strength $<$-40 dBm, even if the FBS has a high-powered transceiver.

B. BS ID Syntax Checking

The BS ID, also known as the BS CGI (Cell Global Identifier), is the globally unique identifier of a cell tower. It is the concatenation of four codes:

$$\text{BS ID} = \text{MCC} + \text{MNC} + \text{LAC} + \text{CID}.$$

Each code has a different size, meaning, and configuration rules. First, MCC is a 3-digit Mobile Country Code ranging from 000 to 999, where the values 0XX, 1XX, and 8XX are reserved. It is allocated by the ITU (International Telecommunication Union) [35]. One country or geographical area can be allocated with multiple MCCs, *e.g.,* China only owns MCC = 460 while the US owns MCC = 310–316.

Second, MNC is a 2-digit Mobile Network Code ranging from 00 to 99. It is jointly allocated and maintained by the government of a country and the ITU. A specific mobile carrier can be allocated multiple MNCs, *e.g.,* China Mobile owns MNC = 00, 02, 07. As of Aug. 2016, no country has used up its 100 MNCs, so the number of valid MCC + MNC combinations is much smaller than 1000×100. Specifically, valid MCC + MNC combinations are publicly available at [36].

Third, LAC is a 16-bit location area code ranging from 0 to 65535. It is assigned by a specific mobile carrier or network. The former three codes together (MCC + MNC + LAC) are also known as the LAI (Location Area Identification) of a BS. As mentioned in § II-A, an FBS usually uses a different LAC code from that of the closest legitimate BS, in order to break nearby cellphones' connections to the legitimate BS.

Finally, CID is a n-bit cell identity code, where $n = 16$ is for 2G/3G BSes (ranging from 0 to 65535) and $n = 28$ is for 4G BSes (ranging from 0 to 268,435,455). CID is also assigned by a specific mobile carrier or network. There should not be two identical CID codes in a given area except in special cases (*e.g.,* re-deployment of a BS by the mobile carrier).

In MsgLogs, we observe many BSes with *syntactically invalid* BS IDs. For example, "460-00-21880-25975" is a valid BS ID, while "460-80-21880-25975" is an invalid BS ID (non-existent MCC + MNC combination). Following the above syntax rules, we find that 0.15% of suspicious messages in MsgLogs were sent by FBSes with invalid BS IDs. We refer to these FBS messages as Set-2. More specifically, in Set-2 2.25% of BS IDs have problems with MCC + MNC, and 97.75% of BS IDs have problems with LAC and/or CID.

C. Message Content Mining

As mentioned in § IV, a fraud message coming from an authoritative phone number should be an FBS message. Given that we have a list of authoritative phone numbers, this motivates us to try using automatic text classification techniques to determine the *functionality* (*i.e.,* legitimate, spam, or fraud) of suspicious messages to identify FBS messages. FBS-Radar determines the functionality of a message by mining its content with SVM (Support Vector Machine), a classical supervised machine learning model for data and text classification [37]. Specifically, we conduct SVM classification by following the seven steps below:

1) Labeling Suspicious Messages. We manually labeled the content of 200,000 suspicious messages as legitimate, spam, or fraud. Specifically, five experts were hired to do the labeling, and each expert independently labeled all the 200,000 messages. For >94% of the messages, the five experts' labels are consistent; for the remainder, we apply the majority rule to determine their functionality [1]. Among these labeled messages, we randomly pick 160,000 messages as the training set, and the remainder are used as the test set.

2) Word Segmentation. We make use of a classical Chinese Word Segmentation (CWS) tool to divide the content of each message in the training set into individual Chinese words. Afterwards, we remove frequently occurring *stop words* that have little discriminatory power [39].

3) Feature Extraction. After eliminating stop words, we take the remaining words in the training set as *features* for message content classification. Accordingly, each message is represented by a few features (words). However, the overall feature set is too large to effectively process (over 200,000 features). Thus, we select the top-10000 most discriminative features from the feature set, which is referred to as *feature extraction* or *feature dimension reduction* [40]. Several methods exist for feature extraction, such as Chi-square Statistics (CHI) [41], Principal Component Analysis (PCA) [42], and Linear Discriminant Analysis (LDA) [43]. We tried all three approaches and found CHI to yield the most precise results [44]. Thus, we employ CHI to extract a 10000-element *feature vector* from the feature set.

4) Quantizing the Feature Vector. Since SVM requires the feature vector to be represented by a series of numbers rather than words, we quantize the feature vector by transforming it from $< word_1, word_2, \cdots, word_n >$ to $< N_1, N_2, \cdots, N_n >$, where N_i is a positive integer. Then, the content of message k is quantized as $< F_k, N_{k_1} : w_{k_1}, N_{k_2} : w_{k_2}, \cdots, N_{k_m} : w_{k_m} >$, where F_k is the functionality and w_{k_j} is the TF-IDF weight [45] of N_{k_j}.

5) Training the SVM Model. Based on the training set and the feature vector, we construct the SVM model using LIBLINEAR [46], a widely used open-source library that implements linear SVM. We manually configure key parameters for the SVM model, *e.g.,* the soft margin parameter C and the tolerance of termination criterion ϵ.

6) Preprocessing the Test Set. For each message in the test set, we segment it into Chinese words, remove the stop words, and quantize its extracted features with the feature vector.

7) SVM Classification of the Test Set. We use the constructed SVM model to classify the content of each message in the test set. After that, we calculate the classification *precision* as the major metric for performance evaluation, as well as *recall*. We use the standard machine learning definitions of precision and recall in our analysis [47].

[1]For the few exceptional cases where the majority rule cannot be applied, *e.g.,* when a message was labeled with {legitimate, spam, spam, fraud, fraud}, we discussed with the five experts in person to determine their functionality. Recently, we note that besides manual labelling in an explicit manner, it is also possible to label spam/fraud content by leveraging implicit user behaviors [38].

Following standard procedures, we performed 10-fold cross validation of our SVM model. We construct 10 different SVM models by randomly generating 10 different pairs of training and test sets (from the 200,000 manually labeled messages), and then calculate their respective precisions and recalls.

Using a commodity 8-core server it takes 20–25 minutes to finish a complete round of SVM classification (including the above Steps 1–7) with a given combination of parameters on a given pair of training and test data. Thus, when the search space includes multiple parameters with hundreds of combinations, we need tens of servers (*i.e.,* the SVM Machine Learning Cluster in Fig. 2) to parallelize the above process. Parallelization reduces the total optimization time of the SVM model to within one day. The precision achieved by the final, tuned SVM model is 98%, while the recall is 91%.

After classifying the content of all messages in MsgLogs, we find that 93.53% are legitimate messages, 6% are spam, and 0.47% are fraud. However, not all of the fraud messages were sent from authoritative numbers; indeed, many originate from numbers that we know nothing about. To be conservative, *we can only attribute fraud messages sent from authoritative numbers to FBSes.* Thus, we find that 0.16% of all suspicious messages were sent from authoritative numbers *and* are classified as fraud by the SVM model. We refer to these FBS messages as Set-3.

D. BS-WiFi Location Analysis

The next detection method we investigate relies on the location of BSes and users. Intuitively, if a user observes a BS ID at a location that does not match its true location (as specified by the mobile network operator), then it is likely to be an FBS with a spoofed BS ID. We refer to all such FBS messages in MsgLogs as Set-4.

Since FBS-Radar does not collect GPS coordinates from users, we instead rely on WiFi information to geolocate users. For a given message log, as long as the MAC addresses of nearby WiFi APs are available, FBS-Radar can usually estimate the geographic location $(lat_{User},\ lon_{User})$ of the user device. We refer to this calculation as *localization* of the user device; we present the details of the localization algorithm used by FBS-Radar in § VII-A.

At a high-level, we conclude that a BS is actually an FBS if $(lat_{User},\ lon_{User})$ does not lie within the *practical* signal coverage area of the BS. For a given BS that sends a suspicious message, we can determine its coverage area by looking up its BS ID in our BS-location database, which will return the true geographic location $(lat_{BS},\ lon_{BS})$ and signal coverage area $(\pi \times r^2)$ of the BS. To calculate the distance between a user and a BS, we apply a series of transformations to their coordinates. In both the BS-location and WiFi-location databases, data are stored in Google Maps format [48]. Given a location $(lat,\ lon)$ like $(24.800947,\ 113.598193)$, we first transform it into radian form:

$$rad(lat) = \frac{\pi \times lat}{180}, \quad rad(lon) = \frac{\pi \times lon}{180}. \qquad (1)$$

Then, we calculate the distance $d_{i,j}$ between two geographical locations $(lat_i,\ lon_i)$ and $(lat_j,\ lon_j)$ as:

$$d_{i,j} = 2 \times r_{earth} \times$$

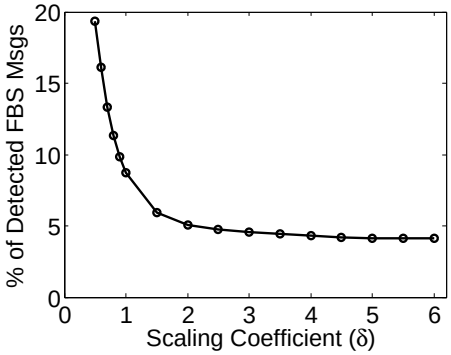

Fig. 4. Ratio of detected FBS messages through BS-WiFi location analysis with different δ.

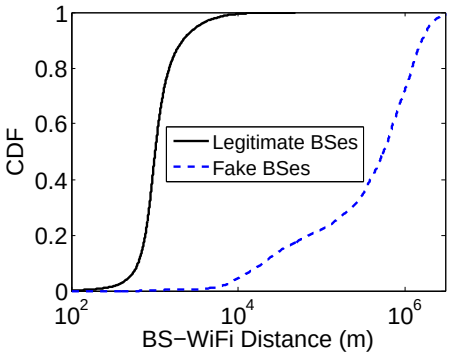

Fig. 5. Distribution of the BS-WiFi distance for legitimate and fake BSes using $\delta = 5.0$.

$$\arcsin\sqrt{(\sin(\frac{rad(lat_i) - rad(lat_j)}{2}))^2 + B}, \quad (2)$$

where $r_{earth} = 6,378,137$ m, and

$$B = \cos(rad(lat_i)) \times \cos(rad(lat_j)) \times (\sin(\frac{rad(lon_i) - rad(lon_j)}{2}))^2.$$

Finally, the BS is taken as an FBS if

$$d_{BS-WiFi} > \delta \times r, \quad (3)$$

where $d_{BS-WiFi}$ is the BS-WiFi distance between (lat_{BS}, lon_{BS}) and (lat_{User}, lon_{User}), and δ is a positive *scaling coefficient*. Theoretically, the signal coverage area of a BS is $\pi \times r^2$ and $\delta = 1.0$. But in practice, the signal coverage area of a BS can become smaller, larger, or directional for a variety of reasons (*e.g.,* weather or obstructions), so we must "scale" the signal coverage area/radius of the BS with δ.

To quantify the impact of δ on the size of Set-4, we investigate different values of δ ranging from 0.5 to 6.0 as to MsgLogs. The results in Fig. 4 indicate that the size of Set-4 decreases as δ increases. When $\delta = 1.0$, Set-4 includes 8.75% of suspicious messages; when $\delta \geq 5.0$, Set-4 becomes quite stable, indicating that this is a "safe", conservative threshold that incurs few false positives. Therefore, we conclude that Set-4 includes at least 4.1% of suspicious messages, corresponding to $\delta = 5.0$.

Additionally, Fig. 5 plots the distribution of BS-WiFi distance ($d_{BS-WiFi}$) for legitimate and fake BSes using the safe threshold $\delta = 5.0$. Obviously, the BS-WiFi distance for FBSes is orders of magnitude greater than that for legitimate BSes. Quantitatively, the median (mean) BS-WiFi distance for legitimate BSes is 1.0 km (1.35 km), while the median (mean) BS-WiFi distances for FBSes is 548 km (729 km).

E. BS-Handover Speed Estimation

For some message logs, WiFi AP information may not be available, and thus BS-WiFi location analysis is not applicable. In these cases, FBS-Radar attempts to identify FBSes by detecting anomalies in *handover speed*. When a user device moves away from the area covered by one BS and enters the area covered by another BS, it is handed over from the first BS to the second BS. If a user device is moving slowly, then we expect correspondingly slow handover speeds, since it will take some time for the user to move from one coverage area to another. Conversely, if a user device is moving quickly (*e.g.,* it is in a car) then we expect fast handovers. However, if a user is moving slowly but we observe a fast handover, this suggests that an FBS is overlapping the coverage area of a legitimate BS (see Fig. 1 (b)).

Suppose BS_1 is the user's currently connected base station, BS_2 and BS_3 are the user's previously connected base stations, and t_1, t_2, and t_3 are the corresponding timestamps when they were most recently observed (refer to Table I). If the handover speed from BS_2 to BS_1 is higher than a certain threshold speed, either BS_2 or BS_1 is fake. Further, FBS-Radar uses the estimated handover speed from BS_3 to BS_2 to determine whether BS_2 or BS_1 is the fake. All the FBS messages detected by this method are referred to as Set-5.

Since we do not know the velocity of the user device during the handover process (*e.g.,* the user may be walking or driving a car), we choose to estimate the maximum, minimum, and average handover speeds from BS_2 to BS_1 as follows:

$$V_{1,2-max} = \frac{d_{1,2} + r_1 + r_2}{t_1 - t_2}, \quad (4)$$

$$V_{1,2-avg} = \frac{d_{1,2}}{t_1 - t_2}, \quad (5)$$

$$V_{1,2-min} = \begin{cases} \frac{d_{1,2} - r_1 - r_2}{t_1 - t_2} & \text{when } d_{1,2} > r_1 + r_2, \\ 0 & \text{when } d_{1,2} \leq r_1 + r_2, \end{cases} \quad (6)$$

where $d_{1,2}$ is the distance between BS_1 and BS_2. The estimation of $V_{2,3-max}$, $V_{2,3-avg}$, and $V_{2,3-min}$ is similar. With regard to the threshold speed, we have two choices: the first is a very conservative threshold

$$threshold_{CRH} = 350\,km/h, \quad (7)$$

which corresponds to the highest operational speed of China Railway High-speed (the fastest railway in China), The second choice is a slightly less conservative threshold

$$threshold_{Highway} = 150\,km/h, \quad (8)$$

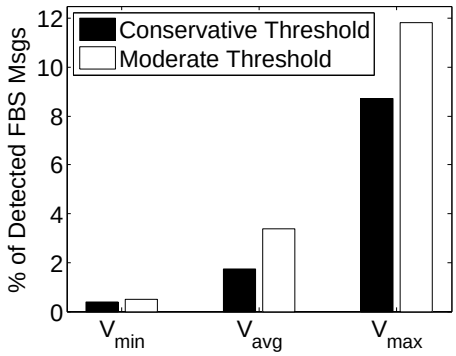

Fig. 6. Ratio of detected FBS messages through BS-handover speed estimation.

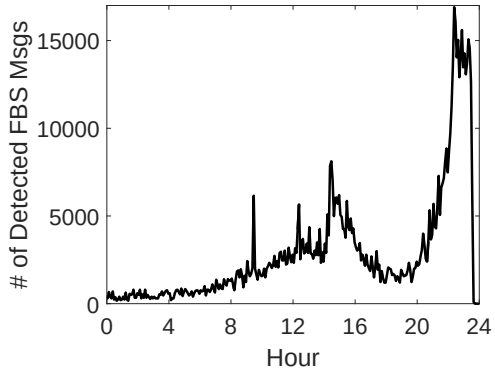

Fig. 7. Number of detected FBS messages in a whole day, where the time interval is 5 minutes.

which is above the speed limit on highways in China.

Consequently, to detect an FBS based on the handover speed, we have three choices for estimating the handover speed and two choices for setting the threshold speed. Obviously, selecting V_{max} and $threshold_{Highway}$ can detect the most FBSes while incurring the most false positives. Conversely, selecting V_{min} and $threshold_{CRH}$ can detect the least FBSes while incurring almost zero false positives. To quantify the impacts of handover and threshold speed selections on Set-5, we apply their different combinations to MsgLogs. The results in Fig. 6 indicate that using V_{max} detects more FBS messages (than using V_{avg} or V_{min}), however this set may contain many false positives. Ultimately, we conclude that Set-5 includes at least 0.39% of suspicious messages, using the most conservative combination V_{min} + $threshold_{CRH}$.

F. Performance Summary

All the FBS messages in MsgLogs detected by the above five methods are referred to as Set-all (= Set-1 ⋃ Set-2 ⋃ Set-3 ⋃ Set-4 ⋃ Set-5). By carefully analyzing all these sets, we make the following major observations:

- *At least 4.7% of suspicious messages are likely to be FBS messages with a high precision.* In § V-A and § V-B, we use strict rules to detect FBS messages, so we regard the detection precision of Set-1 and Set-2 as 100%. As we note in § V-C, the precision of our SVM model is ∼98%, so the detection precision of Set-3 is also ∼98%. On the other hand, in § V-D and § V-E, the detection precision heavily depends on our choice of key parameters (see Fig. 4 and Fig. 6). In these cases, we always make the most conservative choice: $\delta = 5.0$ and V_{min} + $threshold_{CRH}$.

  Overall, this means that 4.7% is a lower bound on FBS messages. Furthermore, we observe that this fraction is stable (between 4.2% and 5.5%) over time, indicating that FBS attackers are not altering their methods in general. We plot the number of detected FBS messages in every 5 minute interval on a typical day (Sep. 18, 2015) in Fig. 7, and observe that most FBS messages are sent in the afternoon and evening.

- *FBSes are pervasive throughout all provinces and municipalities of China.* Fig. 8 plots a heatmap of locations around China where users received FBS messages on Sep. 18, 2015, while Fig. 9 focuses just on FBS messages in Beijing. From both heat maps, we discover that the density of detected FBS messages is proportional to population density.

- *Set-3 is >98% covered by the other four sets, especially Set-4.* This reveals that it is unnecessary for FBS-Radar to collect and analyze the content of SMS messages. To improve privacy for our users, starting in 2016 we automatically opted 99% of FBS-Radar users out of this data collection. Subsequently, FBS-Radar experienced a surge in adoption — the daily average number of suspicious messages increases from 17.5M in 2015 to 32M in 2016, and the daily average number of involved users increases from 8.5M in 2015 to 15.6M in 2016 accordingly.

- *Different from Set-3, only 1.3% of Set-1 is covered by Set-4.* To discover why, we examined the message logs in Set-1, and found the vast majority were reported via a cellular (not WiFi) connection. Given that all messages in Set-1 were delivered by unreasonably high-strength cellular signals, we infer that most user devices in Set-1 were outside, and thus did not have WiFi connections. Thus, their received FBS messages cannot be detected through BS-WiFi location analysis. Additionally, *Set-2 has no intersection with Set-4 and Set-5*, because all messages in Set-2 were sent by FBSes with invalid BS IDs, which cannot be looked up in our BS-location database.

VI. Deployment Experience

In Jan. 2016, we used the results from § V to improve FBS-Radar and deploy additional features. This marks the transition from the "alpha" to the "beta" phase of FBS-Radar deployment. In this section, we describe some of our deployment experiences, including the empirical false positive rate of FBS-Radar, and the practical overhead of FBS-Radar on the client and cloud sides.

False Positive Analysis. As mentioned in § I, if an SMS message is determined to be an FBS message by FBS-Radar, the client-side mobile app quarantines it from the normal SMS message list and puts it to a separate FBS message folder. The

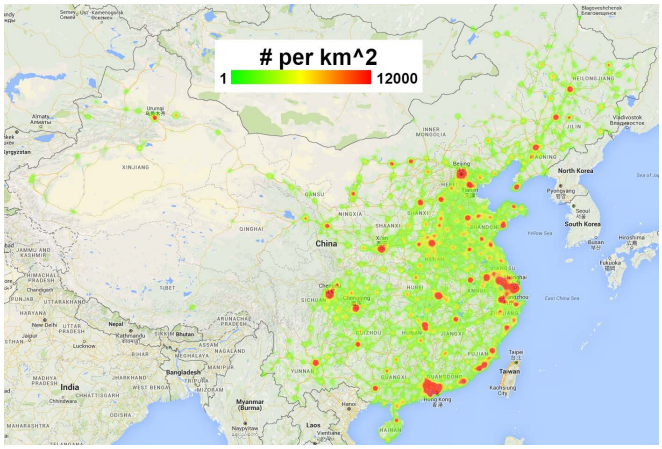

Fig. 8. Heat map of FBS messages detected in China in a typical day.

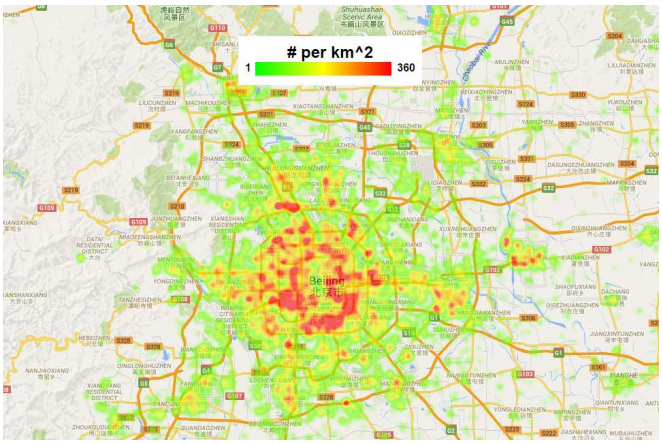

Fig. 9. Heat map of FBS messages detected in Beijing in a typical day.

user is notified about this operation, and she is allowed to mark quarantined messages as valid to recover them from the FBS message folder. When a user marks an FBS message as valid, the app reports this to the cloud.

This user feedback mechanism gives us an opportunity to evaluate the false positive rate of FBS-Radar in practice. Evaluating the false positive rate of FBS-Radar would not be possible without user feedback, since we lack ground truth data about which SMS messages were actually sent by FBSes.

According to their recovery actions, the false positive rate of FBS-Radar is only 0.05%. Delving deeper into the false positive FBS messages, we find that 6% of them come from Set-3, 10% from Set-5, and the remainder (84%) from Set-4. Set-4 (BS-WiFi Location Analysis) is the largest of our five sets, so it is not surprising that it produces the most false positives. Furthermore, inaccuracies and incompleteness in our WiFi-location database can also produce false positives.

System Overhead. FBS-Radar incurs both client- and cloud-side overhead. First and foremost, we need to limit client-side resource consumption in terms of CPU, memory, network traffic, and battery. To this end, all computation-intensive operations, especially FBS message identification and FBS localization, are carried out in the cloud. The client side only reports suspicious messages and receives the returned result (*i.e.,* whether the reported SMS comes from an FBS), so its CPU, network traffic, and battery consumption is trivial. With regard to memory consumption, the client side needs to maintain the Authoritative Phone Number List in memory for determining suspicious messages, which contains 1446 phone numbers amounting to merely 40 KB of memory.

As described at the beginning of § IV, the cloud side consists of three functional components (*i.e.,* Content-free Analysis, Content Analysis, and SVM Machine Learning Cluster) and four data components (*i.e.,* Authoritative Phone Number List, BS-location database, WiFi-location database, and Message Logs). A 10-Gbps network connection handles all traffic to the cloud-side components. All of the functional components use homogeneous commodity servers (HP ProLiant DL380). The configuration of each server is: 2×4-core Xeon CPU E5-2609 @2.50GHz, 4×8-GB memory, and 6×300-GB 10K-RPM SAS disk. Content-free Analysis employs three servers for three different tasks: content-free FBS message detection (§ V-A, § V-B, § V-D, § V-E), user-device localization (§ VII-A), and FBS localization (§ VII-B). Content Analysis uses one server while its affiliated SVM Machine Learning Cluster uses 25 servers (§ V-C). However, note that Content Analysis components are being phased out.

Among the four data components, the Authoritative Phone Number List is only 40 KB in size and thus it is simply loaded into the memory of the Content Analysis server. Both the BS-location and WiFi-location databases are provided by external parties, so FBS-Radar accessed them via web APIs. In addition, petabytes of Message Logs are stored on cheap tapes for backup.

VII. Localizing Fake Base Stations

Above we have described the methods employed by FBS-Radar to identify FBS messages, and quarantine them on the client side. This fulfills our goal of helping to protect users from the attacks of FBSes. In this section, we address another major goal of this project: localizing FBSes so that law enforcement agencies can take them down. Our approach to FBS localization includes two steps: first, FBS-Radar localizes the user device corresponding to each FBS message based on WiFi AP information (which is available for the vast majority of suspicious messages in our dataset). Second, FBS-Radar localizes each identified FBS based on the locations of its affected user devices. The latter step is particularly challenging, since FBSes frequently move and change their IDs, so we must take temporal and spacial locality into account.

The main purpose of FBS-Radar is not to localize FBSes with an extremely high spatial accuracy. Instead, FBS-Radar provides *approximate* location information based on the limited information available in our message logs. Although we might be able to improve accuracy by collecting WiFi signal information like CSI (Channel State Information [49]) and RSSI (Received Signal Strength Indicator [50]), this would require root privileges on user devices. Based on our conversations with police officers, the existing spatial accuracy of our results is enough for effective counter-measures against FBSes.

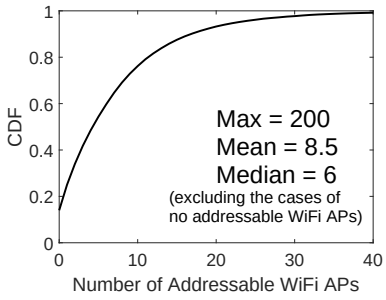

Fig. 11. Distribution of the number of addressable WiFi APs in each message log.

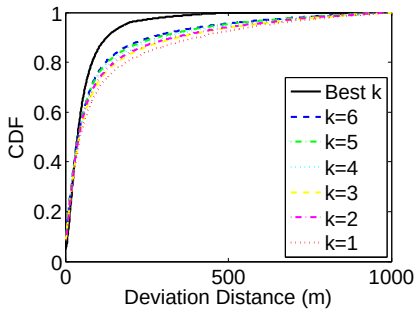

Fig. 12. Distribution of location errors for user devices, using the k-means clustering algorithm.

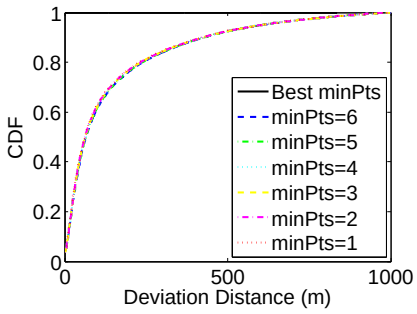

Fig. 13. Distribution of location errors for user devices, using the DBSCAN clustering algorithm.

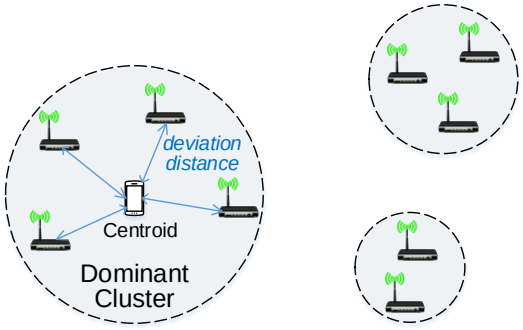

Fig. 10. The centroid of the dominant cluster is the estimated location of the user device.

A. Localizing User Devices based on the WiFi Information

As mentioned in § IV, to localize the user device for each suspicious SMS message, FBS-Radar collects the MAC addresses of all nearby (including both connected and perceived) WiFi APs. We then look up the locations of the WiFi APs in our WiFi-location database; we refer to APs that exist in the database as *addressable*. Finally, as illustrated in Fig. 10, we estimate the user's location by clustering the addressable WiFi APs (using either k-means [51] or DBSCAN [52]), and then selecting the centroid of the *dominant* (*i.e.*, largest) cluster as the user's location.

Fig. 11 illustrates the distribution of the number of addressable WiFi APs in every message log from MsgLogs. When a message log includes addressable WiFi APs, the average number of addressable WiFi APs is as high as 8.5, which provides ample samples for our clustering approach. On the other hand, we note that 14% of message logs do not include any addressable WiFi APs, either due to a lack of nearby WiFi APs, or coverage gaps in our WiFi-location database. We do not localize these 14% of message logs in this work.

Since FBS-Radar does not collect GPS information, it is impossible for us to know the exact location of a user device. In other words, we lack ground-truth user locations to evaluate the accuracy of our estimator. Instead, we use the *average deviation distance* from all WiFi APs in the dominant cluster to the centroid as an approximate measure of location error, where the deviation distance is the distance from a WiFi AP to the centroid of the cluster. Exceptionally, when the dominant cluster contains only one WiFi AP, we are unable to evaluate the corresponding localization accuracy, so we do not consider these cases in our accuracy evaluation. Specifically, the percentage of these cases amounts to 11%, and for these cases we roughly expect the location error to be around 55 meters (*i.e.,* the average location error of "Best k" in Fig. 12).

We tried two classical clustering algorithms to compute clusters of user devices: k-means [51] and DBSCAN [52]. When applying each of them to MsgLogs, we need to select appropriate parameters to achieve sound clusters. With regard to k-means, instead of using a fixed k (the number of clusters) to process all message logs, we dynamically select the best k for each message log after trying all possible values of k. With regard to DBSCAN, there are two key parameters: $minPts$ (the minimum number of points required to form a dense cluster) and ϵ (the scanning distance threshold). For $minPts$, we also dynamically select the best $minPts$ for each message log. But for ϵ, we use a fixed $\epsilon = 1000$ since trying all possible values of ϵ for each message log is practically impossible.

Localization results corresponding to different clustering algorithms and parameters are shown in Fig. 12 and Fig. 13. First, we observe that dynamically selecting the value of k for each message log (shown as "Best k") obviously outperforms using a fixed k for all message logs, so we consider the additional computational overhead to be worthwhile. On the contrary, the performance of DBSCAN is independent of $minPts$. Moreover, dynamical k-means with the "Best k" outperforms dynamic DBSCAN with the "Best $minPts$," and thus we adopt k-means in practice. Fig. 12 reveals that the median (mean) location error for user devices is 36 (55) meters. We do not compare our errors rates with state-of-the-art WiFi localization algorithms from the literature [33], [53], [54], because we lack the necessary meta-data (*i.e.,* signal strength measurements) to perform a fair comparison.

B. Localizing FBSes based on User Device Locations

FBS-Radar localizes an FBS by clustering (using dynamical k-means with the best k) the locations of its affected user devices within a certain *time window*, as depicted in Fig. 14. Here, we need to consider the time window because FBSes may move and/or change their IDs. In other words, only those FBS message logs 1) using the same BS ID, 2) happening in the same time window, and 3) located in the same spatial cluster can be attributed to a specific FBS. Therefore, as demonstrated in Fig. 15, the centroid of every cluster is the estimated location of an FBS, because one ID can be

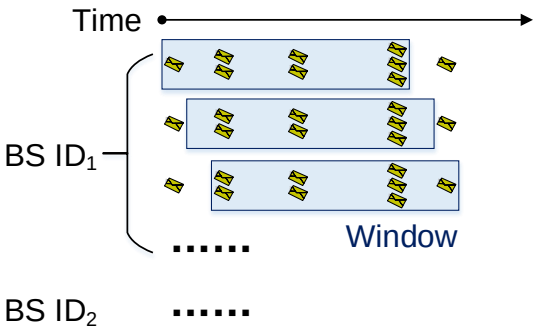

Fig. 14. FBS-Radar localizes an FBS by clustering the locations of its affected user devices in a certain time window. Each message log in the figure corresponds to an affected user device. Note that one ID can be simultaneously used by multiple FBSes at a same time.

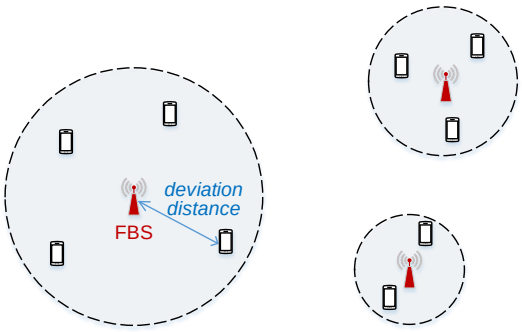

Fig. 15. The centroid of every cluster is the estimated location of an FBS. Note that this is different from Fig. 10 where only the centroid of the dominant cluster is the estimated location of the user device.

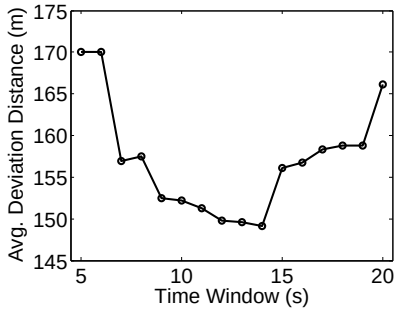

Fig. 16. Relationship between the time window (in seconds) and the accuracy of FBS localization.

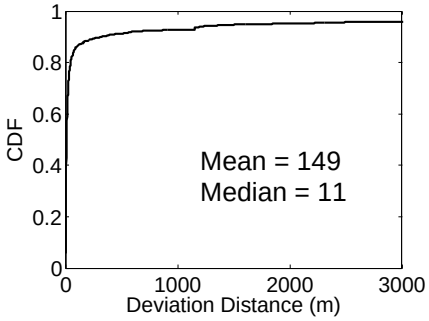

Fig. 17. Distribution of the errors of FBS localization, using the best 14-second time window.

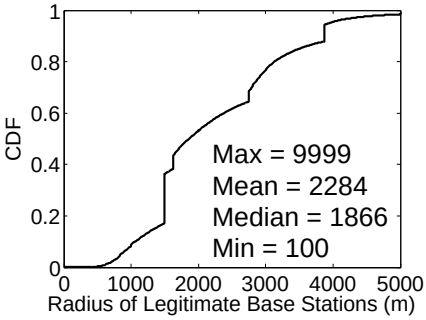

Fig. 18. Distribution of the radius of legitimate base stations.

simultaneously used by multiple FBSes. This is different from Fig. 10 where only the dominant cluster is taken into account.

Inside each cluster, we use the average deviation distance from all affected user devices to the centroid to measure the accuracy of FBS localization. However, when the cluster contains only one affected user device, we are unable to evaluate the corresponding accuracy of FBS localization, so we do not consider these cases in our accuracy evaluation.

To quantify the impact of the time window on the accuracy of FBS localization, we apply different time windows to MsgLogs. The results in Fig. 16 indicate that time windows that are too small or too large degrade the accuracy of FBS localization. The best time window seems to be 14 seconds, where the average deviation distance is 149 m. Although we might be able to improve this accuracy, this would impose onerous requirements on users, as explained in § IV.

Using the 14-second time window, we detail the distribution of deviation distances in Fig. 17. We observe that the median (mean) deviation distance of FBS localization is only 11 (149) meters. In comparison, Fig. 18 shows the distribution of the radius of legitimate base stations, where the average is 2284 meters ($\gg$ 149 meters) and the median is 1866 meters ($\gg$ 11 meters). Hence, such deviation distances are small compared to BS coverage ranges. Additionally, we note that the computations necessary to localize an FBS can be finished in around one second on a commodity server, demonstrating that our system is responsive and scalable.

Through the above FBS localization mechanism, we can estimate the number of active FBSes at any time point of a day. For example, using the 14-second time window, we estimate the numbers of active FBSes at 24 sampling time points on Jan. 16th, 2016. The results are plotted in Fig. 19. We observe an interesting time pattern of FBS deployment behavior, *i.e.,* most FBSes tend to be operational around 22:00 in the evening. We hypothesize this is to avoid being discovered by the police.

Aiding Law Enforcement. We are actively working with law enforcement agencies to put the data produced by FBS-Radar into practice. We have made a website available to the public (at http://shoujiweishi.baidu.com/static/map/pseudo.html) that shows the current locations of detected FBSes in real time. Fig. 20 presents a snapshot of the website's interface, showing the geolocations of detected FBSes at 19:00 on Jan. 16th, 2016.

In addition to the public website, we provide data to the Ministry of Public Security (MPS) and the three major mobile carriers of China. Specifically, once a suspicious SMS message is reported to FBS-Radar and determined to be an FBS message, the cloud side of FBS-Radar notifies specific staff in the MPS of the occurrence of the message, along with the estimated geolocation of the suspected FBS. All data sent to the MPS is stripped of personal information (*e.g.,* the recipient user's phone number) and is encrypted in transit.

Our collaboration with law enforcement is similar to past efforts by other researchers [55], [56], [57]. In these cases and in ours, the goal is to provide insight and intelligence to law

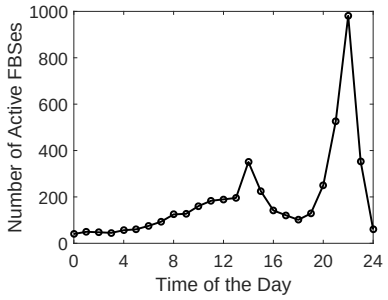

Fig. 19. Estimated numbers of active FBSes at 24 sampling time points on Jan. 16th, 2016, with one sampling time point per hour.

Fig. 20. A snapshot of detected FBSes in China at 19:00 on Jan. 16th, 2016, where each circle represents an active FBS.

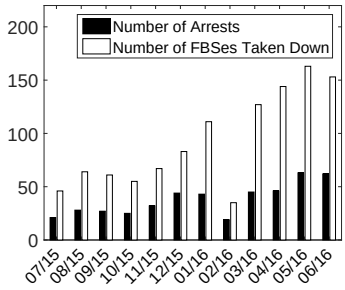

Fig. 21. Number of arrests made by the police and number of FBSes taken down during one year, with the help of FBS-Radar.

enforcement by pointing out activities that are unambiguously illegal, such as operating an unlicensed, high-power radio transceiver on regulated frequencies. However, we respect that this is fraught ethical terrain, as incorrect information could potentially implicate innocent people. In our case, the MPS uses data from FBS-Radar to guide their search for criminals, but only make arrests after conducting an independent investigation, which typically involves catching suspects in the act of operating an FBS. Thus, ordinary people are not placed at risk or directly implicated by data from FBS-Radar; instead, FBS-Radar is simply a tool that helps law enforcement narrow down the search for criminal activity.

According to conversations we have had with the MPS and mobile network operators, the FBS localizations provided by FBS-Radar are sufficiently timely and fine-grained to enable law enforcement activities. Fig. 21 plots the number of FBS take-downs and corresponding arrests made by the Chinese police between Jul. 2015 and Jun. 2016 with the help of FBS-Radar data. In total, the police made 455 arrests and took down 1109 FBSes during this time period. We also see that both arrests and take downs are generally on the rise. The only exception happened in Feb. 2016 ("02/16") when there was a sharp drop of both numbers, since the Chinese New Year Festival happened in this month.

Limitation. It is difficult for FBS-Radar to estimate the number of active FBSes during a relatively long period of time (*e.g.,* during a whole hour or day), as an FBS can frequently change its ID during the period. In other words, we still lack an effective method to attribute seemingly separate attacks back to a single physical FBS over long periods of time.

VIII. Conclusion

In this paper, we share our experiences on the design, implementation, and improvement of FBS-Radar, a large-scale FBS detection and localization system. As of Jun. 2016, FBS-Radar is in use by over 92M people; it successfully quarantines millions of spam and fraud SMS messages per day, and it has helped law enforcement arrest hundreds of FBS operators. FBS-Radar imposes minimal resource requirements on end-user devices, and does not require root privileges or active user intervention (in the vast majority of the time).

FBS-Radar relies on the automated collection of suspicious SMS messages from end-user devices to identify FBSes. We evaluate five different methods that leverage this data to detect FBSes, and find that FBSes can be identified with a high precision without needing to analyze the content of SMS messages. This is an important finding, as it opens the door for future research on FBS detection in a way that preserves users' privacy. Based on feedback from users, we find that the false positive classification rate of FBS messages is only 0.05%.

We have visualized the results of FBS-Radar and released them to the public, which we hope will facilitate further research on FBSes in the academic community, and promote better mobile security around the world. Furthermore, note that our detection methods are extremely conservative by design, since we lacked ground truth about FBSes when we began this project. In the future, it may be possible to develop better algorithms with a higher recall by leveraging the data produced by FBS-Radar as ground truth.

Acknowledgment

We wish to thank the following people for their contributions to the FBS-Radar system or paper. Changqing Han, Min Guo, Xuefeng Luo, Cheng Peng, Junyi Shao, and Xin Zhong (from Baidu Mobile Security) helped develop the system. Tianyin Xu (from UCSD) and Ennan Zhai (from Yale University) offered useful suggestions on the presentation of the submission. Our shepherd Christina Poepper (from New York University) guided the preparation of the camera-ready.

This work is supported by the High-Tech R&D Program of China ("863") under grant 2015AA01A201, the NSF under grants CNS-1563320, CNS-1464335, CNS-1526638, CMMI-1436786 and ECCS-1247944, the NSF of China under grants 61471217, 61472218, 61432002, 61632020, 61572281 and 61520106007, the Key Research Program of Frontier Sciences of CAS under grant QYZDY-SSW-JSC002, and the CCF-Tencent Open Fund under grant AGR20160105.

References

[1] "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update 2014-2019 White Paper," http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white_paper_c11-520862.html.

[2] Z. Li, W. Wang, T. Xu, X. Zhong, X.-Y. Li, Y. Liu, C. Wilson, and B. Y. Zhao, "Exploring Cross-Application Cellular Traffic Optimization with Baidu TrafficGuard," in *Proc. of NSDI*. USENIX, 2016, pp. 61–76.

[3] M. Mouly, M.-B. Pautet, and T. Foreword By-Haug, *The GSM System for Mobile Communications*. Telecom publishing, 1992.

[4] C. Zhang, "Malicious Base Station and Detecting Malicious Base Station Signal," *China Communications*, vol. 11, no. 8, pp. 59–64, 2014.

[5] "Demystifying Fake Base Stations," http://business.sohu.com/20160507/n448197405.shtml.

[6] "Can 4G Really Block the SMS Messages from Fake Base Stations?" http://mobile.163.com/15/0414/08/AN58G1KT0011179O.html.

[7] "Phony cell towers are the next big security risk," http://www.theverge.com/2014/9/18/6394391/phony-cell-towers-are-the-next-big-security-risk.

[8] "19 Fake Mobile Base Stations Found Across US – Are They For Spying or Crime?" http://ibtimes.co.uk/19-fake-mobile-base-stations-found-across-us-are-they-spying-crime-1464008.

[9] "Are your calls being intercepted? 17 fake cell towers discovered in one month," http://computerworld.com/article/2600348/mobile-security/are-your-calls-being-intercepted-17-fake-cell-towers-discovered-in-one-month.html.

[10] "Fake Stingray mobile base stations discovered spying on millions of Londoners," http://www.ibtimes.co.uk/fake-stingray-mobile-base-stations-discovered-spying-millions-londoners-1505368.

[11] "Mobile Security Reports by Baidu," http://shoujiweishi.baidu.com/safety.html#mobile.

[12] "Mobile Security Reports by Qihu 360," http://zt.360.cn/2015/reportlist.html?list=1.

[13] "A Report of Fake Base Stations in China by Tencent, 2016," http://m.qq.com/security_lab/news_detail_361.html.

[14] "Demystifying the Industrial Chain of Fake Base Stations," http://news.sohu.com/20160412/n443925430.shtml.

[15] "Baidu PhoneGuard," http://shoujiweishi.baidu.com.

[16] "IMSI-catcher," http://en.wikipedia.org/wiki/IMSI-catcher.

[17] "Stingray phone tracker manufactured by Harris Corporation," http://en.wikipedia.org/wiki/Stingray_phone_tracker.

[18] "Septier Law Enforcement Cellular Products," http://septier.com/law-enforcement.

[19] "Government Cellphone Surveillance Catalogue," https://theintercept.com/document/2015/12/17/government-cellphone-surveillance-catalogue.

[20] K. Wang, "A Case Study of Fake Base Stations," *China Radio*, pp. 34–36, 2013.

[21] "The 411: Feature phones vs. smartphones," http://cnet.com/news/the-411-feature-phones-vs-smartphones.

[22] "FBS-Signal Detection Cars Do Not Really Work," http://txwh123.com/html/2015/xgaj_1212/125.html.

[23] "12321 Warning: Be Cautious of the SMS Messages from Fake Base Stations," http://12321.cn/12321/warn_detail.php?id=9889.

[24] A. Dabrowski, N. Pianta, T. Klepp, M. Mulazzani, and E. Weippl, "IMSI-Catch Me If You Can: IMSI-Catcher-Catchers," in *Proc. of ACSAC*. ACM, 2014, pp. 246–255.

[25] T. van Do, H. T. Nguyen, N. Momchil *et al.*, "Detecting IMSI-Catcher Using Soft Computing," in *Soft Computing in Data Science*. Springer, 2015, pp. 129–140.

[26] F. van den Broek, R. Verdult, and J. de Ruiter, "Defeating IMSI Catchers," in *Proc. of CCS*. ACM, 2015, pp. 340–351.

[27] "SnoopSnitch project," http://opensource.srlabs.de/projects/snoopsnitch.

[28] "CatcherCatcher project on Osmocom phones," http://opensource.srlabs.de/projects/mobile-network-assessment-tools/wiki/CatcherCatcher.

[29] "AIMSICD: Android IMSI-Catcher Detector," https://cellularprivacy.github.io/Android-IMSI-Catcher-Detector.

[30] "Counterterrorism funds and tools are seeping into local policing," http://www.equalfuture.us/2015/08/26/counterterrorism-stingrays-local-policing.

[31] "The body-worn "IMSI catcher" for all your covert phone snooping needs," http://arstechnica.com/security/2013/09/the-body-worn-imsi-catcher-for-all-your-covert-phone-snooping-needs.

[32] Y. Ding, Z. Tang, and Y. Tang, "Study of Countermeasures Technology Based on Pseudo Base Station for CDMA Mobile Communication System," *Communication Countermeasures*, vol. 101, no. 2, pp. 41–43, 2008.

[33] P. Sapiezynski, R. Gatej, A. Mislove, and S. Lehmann, "Opportunities and challenges in crowdsourced wardriving," in *Proc. of IMC*. ACM, 2015, pp. 267–273.

[34] "10 Ways to Boost Your Wi-Fi Signal," http://www.pcmag.com/article2/0,2817,2372811,00.asp.

[35] "List of Mobile Country or Geographical Area Codes," http://itu.int/pub/T-SP-E.212A-2012/en.

[36] "Status of reserved or assigned ITU-T E.212 shared Mobile Country Codes (MCC) and associated Mobile Network Codes (MNC)," http://itu.int/net/ITU-T/inrdb/e212_901.aspx.

[37] S. Tong and D. Koller, "Support Vector Machine Active Learning with Applications to Text Classification," *The Journal of Machine Learning Research (JMLR)*, vol. 2, pp. 45–66, 2002.

[38] E. Zhai, Z. Li, Z. Li, F. Wu, and G. Chen, "Resisting Tag Spam by Leveraging Implicit User Behaviors," in *Proc. of VLDB*, 2017.

[39] D. Jurafsky and J. Martin, *Speech & Language Processing: An introduction to natural language processing, computational linguistics, and speech recognition*. Pearson Education India, 2000.

[40] "Text Classification for Sentiment Analysis – Eliminate Low Information Features," http://streamhacker.com/tag/feature-extraction.

[41] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 24, no. 7, pp. 971–987, 2002.

[42] H. Uğuz, "A Two-Stage Feature Selection Method for Text Categorization by Using Information Gain, Principal Component Analysis and Genetic Algorithm," *Knowledge-Based Systems*, vol. 24, no. 7, pp. 1024–1032, 2011.

[43] J. Ye and Q. Li, "A Two-Stage Linear Discriminant Analysis via QR-decomposition," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 27, no. 6, pp. 929–941, 2005.

[44] Y. Yang and J. O. Pedersen, "A Comparative Study on Feature Selection in Text Categorization," in *Proc. of ICML*, vol. 97, 1997, pp. 412–420.

[45] "tf-idf: A Single-Page Tutorial - Information Retrieval and Text Mining," http://tfidf.com.

[46] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: A Library for Large Linear Classification," *The Journal of Machine Learning Research*, vol. 9, pp. 1871–1874, 2008.

[47] "Wikipedia page for precision and recall," http://en.wikipedia.org/wiki/Precision_and_recall.

[48] "The Google Maps Geocoding API," http://developers.google.com/maps/documentation/geocoding.

[49] "CSI: Channel state information," http://en.wikipedia.org/wiki/Channel_state_information.

[50] "RSSI: Received signal strength indication," http://en.wikipedia.org/wiki/Received_signal_strength_indication.

[51] J. A. Hartigan and M. A. Wong, "Algorithm AS 136: A k-means Clustering Algorithm," *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 28, no. 1, pp. 100–108, 1979.

[52] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise," in *Proc. of KDD*, vol. 96, no. 34, 1996, pp. 226–231.

[53] P. Sapiezynski, A. Stopczynski, R. Gatej, and S. Lehmann, "Tracking Human Mobility using WiFi Signals," *PLoS ONE*, vol. 10, no. 7, p. e0130824, 2015.

[54] Z. Zhang, X. Zhou, W. Zhang, Y. Zhang, G. Wang, B. Y. Zhao, and H. Zheng, "I am the Antenna: Accurate Outdoor AP Location using Smartphones," in *Proc. of MobiCom*. ACM, 2011, pp. 109–120.

[55] B. Stone-Gross, M. Cova, L. Cavallaro, B. Gilbert, M. Szydlowski, R. Kemmerer, C. Kruegel, and G. Vigna, "Your Botnet is My Botnet: Analysis of a Botnet Takeover," in *Proc. of CCS*. ACM, 2009.

[56] K. Levchenko, A. Pitsillidis, N. Chachra, B. Enright, M. Félegyházi, C. Grier, T. Halvorson, C. Kanich, C. Kreibich, H. Liu *et al.*, "Click Trajectories: End-to-End Analysis of the Spam Value Chain," in *Proc. of S&P*. IEEE, 2011, pp. 431–446.

[57] D. Y. Wang, M. Der, M. Karami, L. Saul, D. McCoy, S. Savage, and G. M. Voelker, "Search + Seizure: The Effectiveness of Interventions on SEO Campaigns," in *Proc. of IMC*. ACM, 2014, pp. 359–372.