ON TRAFFIC ANALYSIS OF 4G/LTE TRAFFIC

SEAN W. CALDWELL

Bachelor of Science in Computer Engineering

Cleveland State University

December 2017

Submitted in partial fulfillment of the requirements for the degree

Master of Science in Electrical Engineering

at the

CLEVELAND STATE UNIVERSITY

August 2021

© 2021 SEAN W. CALDWELL

We hereby approve this thesis for

SEAN W. CALDWELL

Candidate for the Master of Science in Electrical Engineering

degree for the

Department of ELECTRICAL AND COMPUTER ENGINEERING

and the CLEVELAND STATE UNIVERSITY'S

College of Graduate Studies by

Thesis Committee Chairperson, Dr. Ye Zhu

Department & Date

Committee Member, Dr. Yongjain Fu

Department & Date

Committee Member, Dr. Sui-Tung Yau

Department & Date

Student's Date of Defense: 06/22/2021

DEDICATION

To my loving and adventurous family...

ACKNOWLEDGMENTS

I would like to thank the following people:

Dr. Ye Zhu for giving me such a challenging opportunity, for his inspiring direction for the research and for his great support in all aspects of my study.

Dr. Yongjain Fu, and Dr. Siu-Tung Yau, who are on my committee, for their time in reviewing and evaluating this dissertation.

Dr. Ye Zhu for my improvement in finding ways to be efficient in how I work and how I should solve research problems.

Thank you to my friends and coworkers for their kind help and friendship.

ON TRAFFIC ANALYSIS OF 4G/LTE TRAFFIC

SEAN W. CALDWELL

ABSTRACT

In this thesis, we draw attention to the problem of cross-service attacks, that is, attacks that exploit information collected about users from one service to launch an attack on the same users on another service. With the increased deployment and use of what fundamentally are integrated-services networks, such as 4G/LTE networks and now 5G, we expect that cross-service attacks will become easier to stage and therefore more prevalent. As running example to illustrate the effectiveness and the potential impact of cross-service attacks we will use the problem of account association in 4G/LTE networks. Account association attacks aim at determining whether a target mobile phone number is associated with a particular online account. In the case of 4G/LTE, the adversary launches the account association attacks by sending SMS messages to the target phone number and analyzing patterns in traffic related to the online account. We evaluate the proposed attacks in both a local 4G/LTE testbed and a major commercial 4G/LTE network. Our extensive experiments show that the proposed attacks can successfully identify account association with close-to-zero false negative and false positive rates. Our experiments also illustrate that the proposed attacks can be launched in a way that the victim receives no indication of being under attack.

TABLE OF CONTENTS

Page

LIST OF TABLES

Tables Page

LIST OF FIGURES

Figures Page

Chapter I

Introduction

In this paper, we study the problem of *cross-service attacks*, that is, attacks that exploit information collected about users from one service to launch an attack on the same users on another service. With the increased deployment and use of what fundamentally are integrated-services networks, such as 4G/LTE networks and now 5G, we expect that cross-service attacks will become easier to stage and therefore more prevalent. As multiple services share the underlying infrastructure, for example SMS and IP services on 4G/LTE, information gathered from one service on the network can disclose information about users of other services.

As running example to illustrate the effectiveness and the potential impact of cross-service attacks, we will use the problem of account association in 4G/LTE networks. The goal of account association is to determine whether a target mobile phone number is associated with a particular online account for an IP-based service such as Skype or Netflix. The demand for associating phone numbers with online accounts may emerge in a variety of settings. For example,any smartphone applications and services require users to provide their phone numbers during account registration. If adversaries are able to associate service accounts with their registered telephone numbers, they may be able to compromise the privacy of the application and that of their users. For example, an adversary may suspect a particular individual of anonymously broadcasting live videos through a smartphone application such as Periscope.

The adversary can infer the identity of the Periscope account owner by associating the account with the owner's phone number. Similarly, this particular form of account association can be used by an online retailer to prevent purchases made from unauthorized phones by detecting the association between the account in use and the phone number authorized during account registration.

4G/LTE networks are particularly susceptible to the type of cross-service attacks addressed in this paper because all the services provided by 4G/LTE are relying on the same IP-based communication channels. This includes services that one does not traditionally think of as IP based, such as voice calls and SMS messages, and services that are generally built on IP, such as high-definition mobile video, mobile augmented or virtual reality, mobile cloud computing, and video conferencing. We will see that the requirement for 4G/LTE to in many cases provide low-delay and high-bandwidth services generally renders cross-service attacks particularly effective. The opportunities for this type of attack will grow as well. In January 2017, the Global Mobile Suppliers Association (GSA) reported that there were 581 LTE or LTE-Advanced networks across 186 countries [1], and in December 2018 the GSA reported that there were 3.99 billion LTE subscribers world-wide [2]. Given the popularity of 4G/LTE and 5G, such attacks will likely become quite prevalent and impactful.

We will therefore study a class of account association attacks specifically designed for the 4G/LTE networks. An adversary launches the attacks by sending SMS messages to the target phone number and by analyzing the traffic related to the account. If the analysis can find traffic patterns corresponding to the SMS messages, the attack assumes that the target phone number is associated with the account.

We performed a number of attacks within a local 4G/LTE testbed. The experiment results were very encouraging, indicating that the proposed attacks can successfully identify account associations with negligible false-positive and false-negative rates. Our local experiments also show that the attacks can be "silent" to the victim, meaning that the victim receives no indication that it is the target of an account association attack. These "silent" attacks are possible because existing smartphones have no abilities to process messages in some specific formats, such as CPIM [3]. We will show that although the victim does not know that a "silent" attack is under way, the proposed attacks can actually achieve better identification performance when CPIM or similar message formats are being used compared to attacks with user-visible SMS messages.

Our experiments show, however, that these attacks in their basic forms are not effective when deployed in a commercial 4G/LTE network. The main challenge comes from the fact that the SMS service center and the uplink bandwidth are shared among many service subscribers. The scheduling algorithm used in this sharing tend to spread the SMS messages in order to prevent the batching of SMS messages to any particular subscriber. This makes it hard for the attacker to find the correspondence between the traffic patterns and SMS messages. To overcome this challenge, we design a new class of account association attacks that is particularly well suited to be deployed against commercial 4G/LTE networks. These new attacks leverage the knowledge about how spreading and throttling is realized by network operators, and take advantage of the spreading and throttling caused by the scheduling for identification. We evaluate this new class of attacks using an extensive suite of experiments over a major commercial 4G/LTE network. The results illustrate the effectiveness of these attacks and show that we can identify account associations with very high accuracy on large commercial 4G/LTE networks as well.

The success of the proposed account association attacks should encourage us rethink the architecture of 4G/LTE networks and integrated-services architectures[1] in

[1] We use the term "integrated services" loosely here, and we include architectures in general that use the same underlying platform to provide a diverse set of services, much as 4G/LTE uses that same IP-based communication channels to carry a diverse set of apparently unrelated services for the user.

general. As mobile-network providers transitioned from 3G to 4G/LTE networks they also transitioned to a fully IP-based underlying platform. As a result, a highly diverse set of services, including voice calls and SMS messages, are now provided over a shared, IP-based network. Integrating these services over a shared IP-based platform brings many advantages, such as better scalability and richness of features. It does expose the services to attacks, primarily side-channel attacks, that span individual services. As we will show, such attacks are particularly effective against services that provide Quality-of-Service (QoS) guarantees, such as voice, video, and various forms of augmented or virtual reality services. As a result, attention must be paid as we transition to next-generation architectures for mobile network to the importance preventing cross-service covert channel attacks, much as the attacks proposed in this paper.

The remainder of this paper is organized as follows. Chapter II reviews background on the network architecture for 4G/LTE communications. The threat model is described in Chapter III. Chapter IV presents details of the account association attacks in our local 4G/LTE testbed and performance of the proposed attacks. Chapter V presents details of the account association attacks in a major 4G/LTE network and performance of the proposed attacks. We discuss countermeasures and experiments in the major commercial 4G/LTE network in Chapter VI. Chapter VII reviews related work on previous attacks on SMS and 4G/LTE networks and related work on traffic analysis. We conclude the paper in Chapter VIII.

Chapter II

Background

2.1 Network Architecture for 4G/LTE Communications

4G/LTE networks provide all their services over a flat, all-IP architecture. This is in contrast to the hierarchical structures used in previous architectures, such as 2G and 3G. The flat, all-IP architecture of 4G/LTE enables constantly higher bandwidths with significantly lower data-transfer delays than the architecture of previous 2G and 3G networks.

As show in Figure 1, a *User Equipment* (UE), such as a smartphone, connects to a 4G/LTE network through one of the base stations, also called *Evolved Node B* devices (eNodeBs). The eNodeB devices are elements of the *Evolved Universal Terrestrial Radio Access Network* (E-UTRAN), which is responsible for keeping the UEs wirelessly connected and which is designed to help improve overall wireless connectivity. The eNodeBs are connected to the *Evolved Packet Core* (EPC), which provides 4G/LTE services to a subscribed UE.

The EPC consists of the following components: (1) The *Mobile Management Entity* (MME) is responsible for tracking the UE and for providing the initial connection and authentication for the UE device. As the MME's purpose is to maintain a connection, it is responsible for most control-plane functions. Particularly important for the proposed attacks is the MME's role in activation and deactivation of *bearers*, which uniquely identify traffic flows with specific Quality of Service (QoS) require-

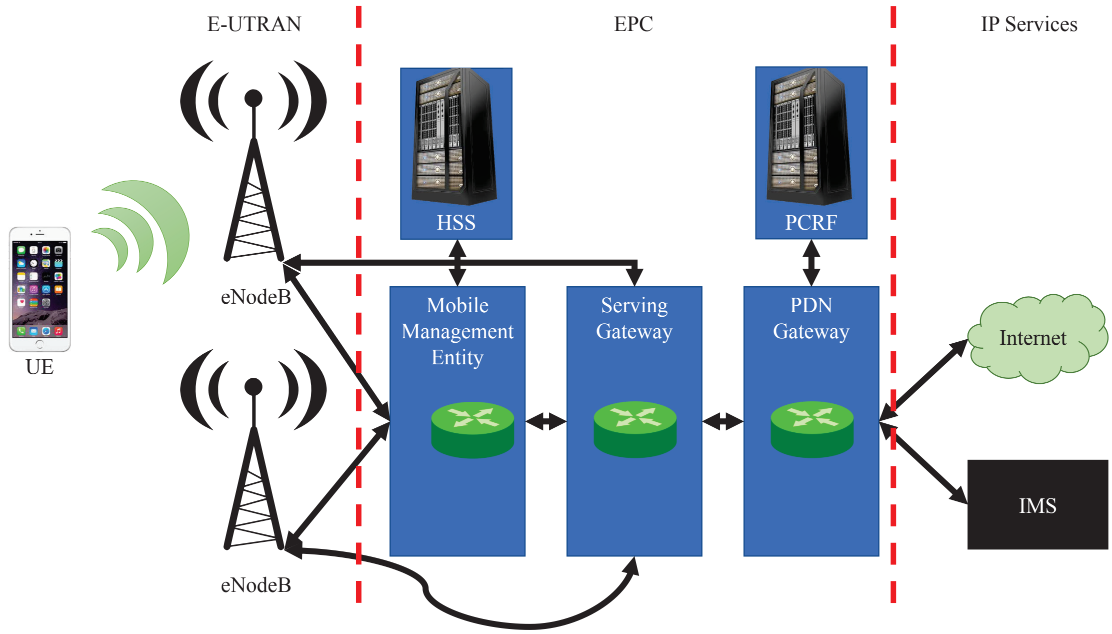


Figure 1: Architecture of 4G/LTE Mobile Networks

ments. We give more details on bearers in 4G/LTE networks in Figure 2. (2) the *Home Subscriber System* (HSS) is a database that maintains user profiles and location information. It acts as a source for name and address resolution. It is also responsible for providing the appropriate authentication and authorization information required for a UE to access the 4G/LTE network services. (3) The *Serving Gateway* (SGW) is responsible for managing all IP packets that flow through the network. It is also responsible for handling handovers whenever UEs move between eNodeBs. (4) The *Packet Data Network Gateway* (PGW) is responsible for allocating IP addresses to the UEs. It provides an interface towards the Internet and to the *IP Multimedia Subsystem* (IMS). A PGW often implements a *Policy and Charging Rule Function* (PCRF), which is responsible for determining in real-time whether a particular traffic is to be allowed in the network. It also is responsible for tracking network usage for billing purposes. Particularly related to the proposed attacks is PGW's role in setting up the appropriate bearers to establish the corresponding connections to IMS services. We note that the EPC is – differently than in 2G/3G mobile networks – an *IP-only core network* that supports packet-switching.

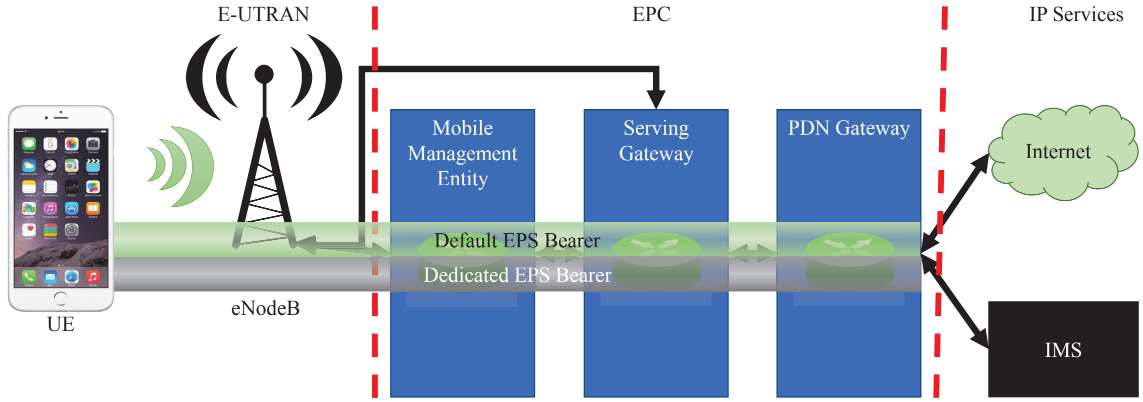


Figure 2: A Typical Scenario of VoLTE

2.2 Voice over LTE

The *IP Multimedia Subsystem*, or IMS, is the current designated solution for offering multimedia services in 4G/LTE mobile networks. It shifts the voice communications of mobile devices from the legacy circuit-switching technology to the IMS-based, packet-switching design used in 4G/LTE. In comparison to 3G networks and even Voice over IP (VoIP), voice-over-LTE (VoLTE) packets have smaller packet headers and therefore save bandwidth [4].

A typical scenario of VoLTE communications is shown in Figure 2. A VoLTE-capable phone is connected to the 4G/LTE network with two bearers. The *default bearer* is established when a UE connects to a 4G/LTE network. It remains established to provide the UE with always-on IP connectivity. The default bearer is typically setup without any QoS requirements. It is primarily used for general IP traffic. A *dedicated bearer* is used for VoLTE communications. It can be setup during the call setup or when the UE is attached to 4G/LTE network. Since voice communications are delay-sensitive, the dedicated bearers established for voice communications have specific QoS requirements.

SMS is a store-and-forward service with a long history. In most current 4G/LTE networks, the SMS service is based on IMS. Due to their demand for timeliness, SMS packets are usually sent in a dedicated bearer with QoS requirements that are higher than those of general IP packets. The IMS service utilizes the Session Initiation Protocol (SIP) [5] to handle SMS delivery. A SIP session is maintained between the phone's SMS application and the IMS server. In turn, the IMS server is responsible for bridging the SIP session and the SMS center. For compatibility with 2G/3G networks and protocols, a 4G/LTE UE may also support circuit-switching fallback [6] for SMS message delivery. In this paper we focus on the IMS-based SMS message delivery.

Chapter III

Threat Model

In the following, we assume that the adversary's goal is to associate a target mobile phone number with a particular user account of a given IP-based service. As introduced in Section I, this approach can also be used for legitimate purposes, such as to confirm whether an account is accessed from a registered phone number. Figure 3 shows the threat model. We assume that both the adversary's and victim's smartphones are connected to commercial 4G/LTE networks. To launch the attack, the adversary is assumed to have the following capabilities:

1. The attacker can send SMS messages to the suspected phone number. Our experiments show that if the adversary can choose a message format incompatible with smartphones, for example CPIM [3, 7], the messages sent by the adversary will not be shown on the victim's phone. In other words, the victim will likely not be aware of the messages that enable the adversary's attack.

2. The adversary can collect traffic generated by the application associated with the online account. We make no assumptions about whether the application uses encryption or whether it pads the packets to show a fixed packet size. As a result, the adversary has no access to either content of the victim's communication or payload size.

3. We make no assumptions where the traffic is observed. The attacker can collect the traffic anywhere along the path from the victim's smartphone. Obviously,

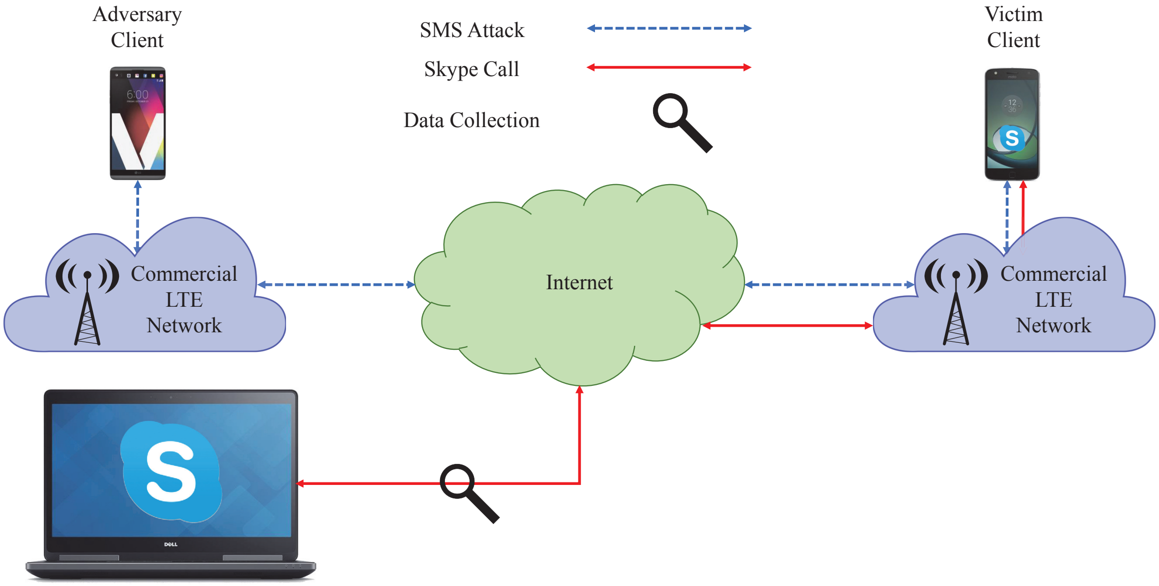


Figure 3: Threat Model

the closer to the victim's phone the traffic is collected, the less interference the collected traffic data show from other traffic in the network.

4. We assume that the traffic collected by the adversary is *aggregated*. There are many reasons (the use of VPN is one of them) why the adversary cannot filter the collected traffic to gain access to the traffic flow of interest. In other words, the collected traffic includes not only the traffic generated by the application of interest, but other traffic as well. We note that if the adversary has control of the traffic destination, such as a Skype party in the call with the victim, the adversary can separate the traffic generated by the application and so gain access to the traffic of the flows of interest. In this case, the attack will of course be more effective. For the sake of generality, we do not assume that the adversary has control of the traffic destination. So the traffic collected by the adversary is aggregated.

In the rest of this paper, we will be using Skype as an example for the IP-based service. In this example attack, the adversary is suspecting that a given Skype name is being used on a smartphone with a suspected phone number. Skype is a particularly attractive IP-based service for cross-service attacks because on one hand it is susceptible to timing-based side-channel attacks because of its need to provide quality-of-service. In the other hand, Skype prides itself to having a unique set of features to protect privacy of Skype calls, such as strong encryption [8], proprietary protocols [8], unknown codecs [9], dynamic path selection [10], and constant packet rates [11].

Chapter IV

Identifying Account Association on A Local Testbed

In this section, we present our investigation on account association in a local testbed with 4G/LTE connectivity. We first introduce the setup of the local testbed. We then describe an approach that an attacker can use to identify if a target phone number is associated with a given user account for an IP service, in this case Skype. In the following we will use the term "identification" to denote the process of identifying if there is an association between a target phone number and a given user account. The approach to identify whether an association exists is therefore called an *identification appraoch*. At the end of this section we will present the performance of the described identification approach.

4.1 Local Testbed

Figure 4 illustrates the setup of the local testbed. The testbed is built around the Keysight LTE test solution, including an Agilent PXT E6621A LTE wireless communications test set, the E6966B IMS-SIP Network Emulator Software, and the N6061A LTE Protocol Logging and Analysis application. The victim's phone is connected to the PXT E6621A through 4G/LTE connections. The VoLTE service, including the SMS message service, is provided by the IMS server running the Keysight E6966B-1FP IMS-SIP Server Emulator Software. The adversary sends SMS text messages through the IMS client running Keysight E6966B-2FP IMS-SIP Client Emulator Soft-

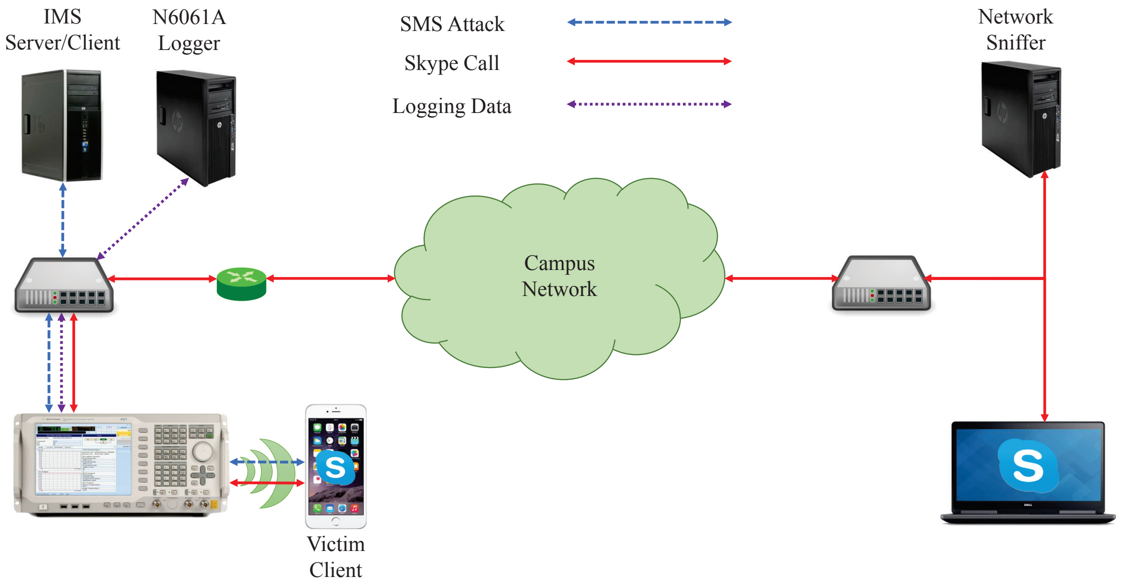


Figure 4: A Local 4G/LTE Testbed

ware. The Skype call comes in through a campus network, and the adversary observes Skype traffic originating at the victim's phone by collecting traffic on the farthest hop of the path of the Skype connection.

4.2 Identifying Account Association

4.2.1 Rationale

The identification of the account association is feasible because of the differences in the bearers (and their QoS levels in particular) used to transport Skype packets *vs.* SMS text messages. As described in Section II, SMS text messages are usually sent in a bearer with higher Quality of Service (QoS) requirements such as the bearer for VoLTE calls [12, 13, 14, 15] because of SMS's close relationship to voice, while usual IP packets including the Skype packets are sent in the default bearer, which usually has lower QoS requirements [12, 13, 14, 15]. The adversary identifies the account association by sending SMS text messages to the victim's phone. Since the

SMS messages are sent with higher QoS, the traffic of other IP services with lower QoS (Skype in our case) will be disturbed. The IP traffic will therefore display a inter-packet-time pattern that has been caused by the SMS messages. The adversary can therefore identify the account association by correlating the timing of the SMS messages with the timing pattern of IP traffic (Skype in our case) generated by victim's phone.

Figure 5 shows an example of the effect on Skype traffic caused by the interfering SMS messages. The graph shows the throughput of Skype traffic over time. In this example, three bursts of SMS messages are generated. We observe that each burst affects the rate of the Skype traffic. The results in this figure are obtained from the testbed shown in Figure 4. The length of the sampling window for the computation of the throughput curve is 2.5 seconds. These results illustrate the two primary challenges with identification: (1) The Skype traffic throughput fluctuates over time, and some decreases in the throughput curve may not be caused by the interfering bursts of SMS messages. (2) The size of SMS message bursts may be limited, and therefore may not be generating easily-detectable interference patterns. For illustrative purposes, the results in Figure 5 use a large number of SMS messages (425 messages) in each burst. Obviously, in real 4G/LTE networks it is not possible to send such large a number of SMS text messages in a burst because the network operators have limits in place on SMS message sending rates.

4.2.2 Identification Algorithm

To start the identification, the adversary sends a sequence of bursts with n messages each. The adversary identifies the account association by detecting the pattern using Algorithm 1. The algorithm can be divided into three steps: data extraction, cross correlation, and decision, respectively. In the data extraction step, we first store the number of SMS messages sent within each burst in the array b_{len}. We then store the

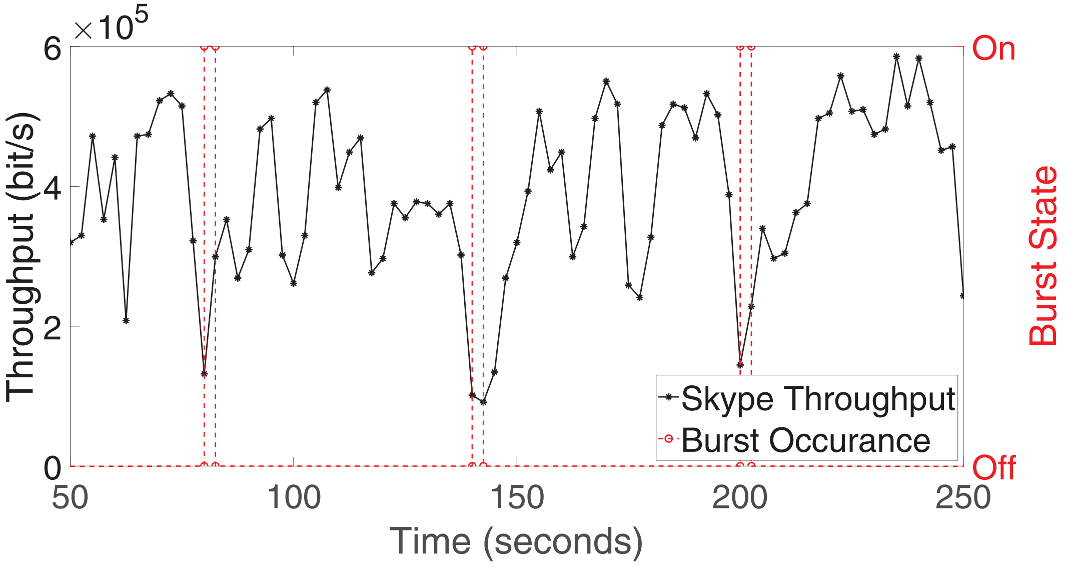


Figure 5: Effect of SMS Messages on Skype Traffic

number of packets collected during the corresponding time slots at the data collection point in array $traf$. We also randomly pick time slots that do not overlap with the burst periods. The number of SMS messages sent during the randomly picked time slots is zero and kept in array b_{len} as well. Similarly, the number of packets collected during the corresponding time slots is kept in array $traf$.

After the data extraction step, the values in array b_{len} and array $traf$ are cross-correlated. As show in Figure 5, the SMS message bursts can cause decreases in the rate of traffic sent from the victim's smartphone. Therefore, if the pattern caused by the SMS message bursts is detected, the two arrays will be highly negatively-correlated. The Cross-Correlation function of two vectors $A = [a_1, a_2, \cdots, a_m]$ and $B = [b_1, b_2, \cdots, b_m]$ used in Algorithm 1 is defined as follows:

$$CrossCorrelation(A, B) = \frac{\sum_{i=1}^{m}(a_i - \bar{a})(b_i - \bar{b})}{\sqrt{\sum_{i=1}^{m}(a_i - \bar{a})^2}\sqrt{\sum_{i=1}^{m}(b_i - \bar{b})^2}} \tag{IV.1}$$

The pattern caused by the bursts of SMS messages may take some time to appear at the data collection point. This is the case especially when the data collection point

```
input : $n$ - number of SMS message bursts, $len_{traf}$ - the length of traffic
        collected at the data collection point, array $b_{len}$ with $b_{len}[i]$ denoting
        the number of messages in the $i$th SMS message burst, $blen_{avg}$ -
        average length of SMS message bursts, array $b_{begin}$ with $b_{begin}[i]$
        indicating the start time of the $i$th SMS message burst, array $b_{end}$
        with $b_{end}[i]$ indicating the end time of the $i$th SMS message burst,
        $bound$ - the bound on the delay between the sending time of a burst
        of SMS messages and the arrival time of the corresponding pattern
        observed at the data collection point, $inc$ - step increase;
output: $dec$ - Detection decision, 1 means detected, 0 means not;
// counting messages sent in each burst
$shift \leftarrow 0$;
$t \leftarrow b_{end}[1]$;
$j \leftarrow 1$;
while $shift + b_{end}[n] < len_{traf}$ do
    for $i \leftarrow 1$ to $n$ do
        $traf[i] \leftarrow$ the number of packet arrivals at the data collection point
          during $[b_{begin}[i] + shift, b_{end}[i] + shift]$;
        // For simplification, we assume that the traffic collected
           at the data collection point is long enough so that the
           array $traf$ and $b_{len}$ can be of the same length.
        randomly pick one duration $[t_{random}, t_{random} + blen_{avg}]$ not overlapping
          with SMS burst durations;
        $b_{len}[n + i] \leftarrow$ the number of SMS messages in the duration
          $[t_{random}, t_{random} + blen_{avg}]$ ;
        $traf[n + i] \leftarrow$ the number of packets in the duration $[t_{random} +
          shift, t_{random} + blen_{avg} + shift]$ ;
    end
    $corrval[j] \leftarrow$ CrossCorrelation($b_{len}[1..2n]$, $traf[1..2n]$);
    $j \leftarrow j + 1$;
    $shift \leftarrow shift + inc$;
end
$mean_{corr} \leftarrow$ mean of array corrval;
$std_{corr} \leftarrow$ stand deviation of array corrval;
$min_{corr} \leftarrow$ the minimum of $corrval[1..\lceil \frac{bound}{inc} \rceil]$ ;
// $\lceil\rceil$ denotes the ceiling function
$dec \leftarrow$ Decision($min_{corr}, mean_{corr}, std_{corr}$);
```

Algorithm 1: Identification Algorithm for a Local Testbed

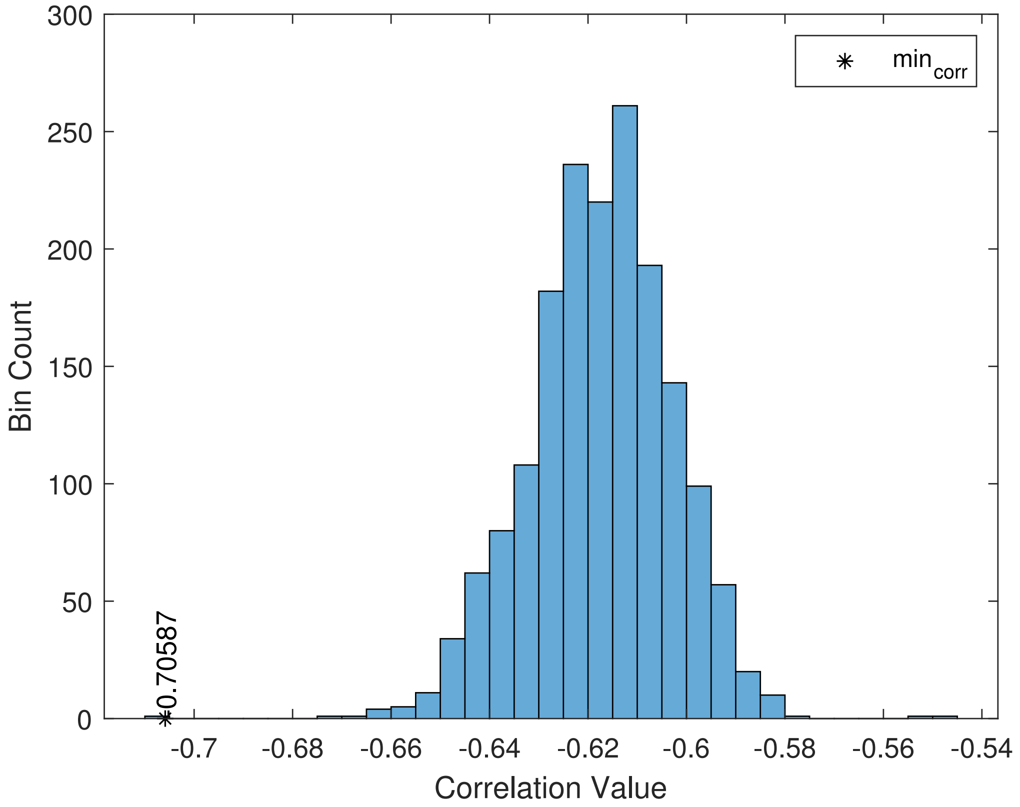


Figure 6: An Example Correlation

is far (in number of hops) from the origin of the Skype call. The delay includes (1) the delay between when the SMS messages are sent and when they are received by the victim's phone, (2) the delay between when the packets are sent by the victim's phone and when they are received at the data collection point. Because of this delay, the algorithm cross-correlates SMS message bursts, i.e., the array b_{len}, with the *delayed* versions of traffic, possibly containing the pattern, i.e., the $traf$ array. In Algorithm 1, the delay is added through the variable *shift*, and *shift* is incremented by inc seconds in each iteration. As a result, the traffic, which possibly contains the pattern, is essentially shifted in each cross-correlation. In our experiments, we set inc to be half the amount of time required to send out a burst.

Based on the cross-correlation results obtained in the previous step, the algorithm calls the function $Decision$ to determine whether the pattern is detected. Figure 6 shows an example histogram of the cross-correlation results. We can observe from the figure that the cross-correlation values are close to a normal distribution. We can also observe that the cross-correlation value is very far away from the mean of the normal distribution when the pattern is synchronized with the SMS message bursts. The decision logic is therefore simple: If within the bound of the delay between SMS message sending time and the arrival time of the corresponding pattern at the data collection point, the cross-correlation between the SMS message bursts and the traffic is less than the decision threshold, the algorithm declares the pattern to be found. In other words, the account association is confirmed. We define the decision threshold to be three standard deviations of the mean on the left side of the normal distribution as shown in Figure 6. In our experiments, the *bound* is roughly 1 second because the SMS message delay is around 0.3987 seconds and the packet delay between victim's phone and the data collection point is about 0.75 seconds. The *bound* is much smaller than the length of SMS message bursts, which are about one and a half seconds long.

```
Function Decision($min_{corr}$, $mean_{corr}$, $std_{corr}$):
    input  : $min_{corr}$ - minimum correlation within the delay bound, $mean_{corr}$
             - mean of correlation, $std_{corr}$ - standard deviation of correlation;
    output: $dec$ - Detection decision, 1 means detected, 0 means not;
    $dec \leftarrow 0$;
    if $min_{corr} < mean_{corr} - 3 * std_{corr}$ then
        $dec \leftarrow 1$
    end
    return dec;
```

Algorithm 2: Decision Function

4.3 Identification Performance

To evaluate the performance of the identification algorithm, we conducted experiments in the local testbed shown in Figure 4. In these experiments, SMS messages were sent in two different message formats: 3GPP2 [16] and CPIM [3, 7]. We choose the 3GPP2 message format because of its popularity [7]. The CPIM message for-

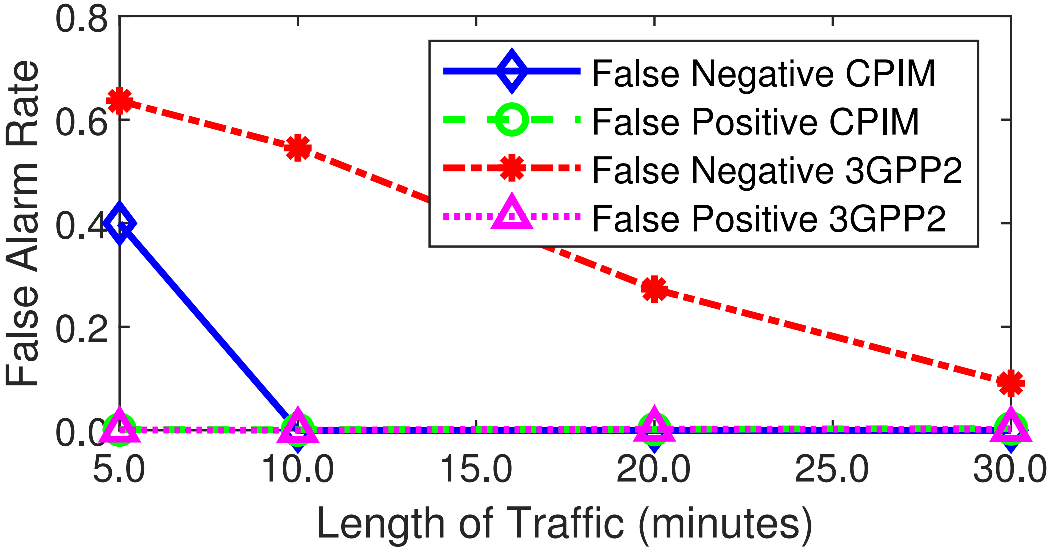
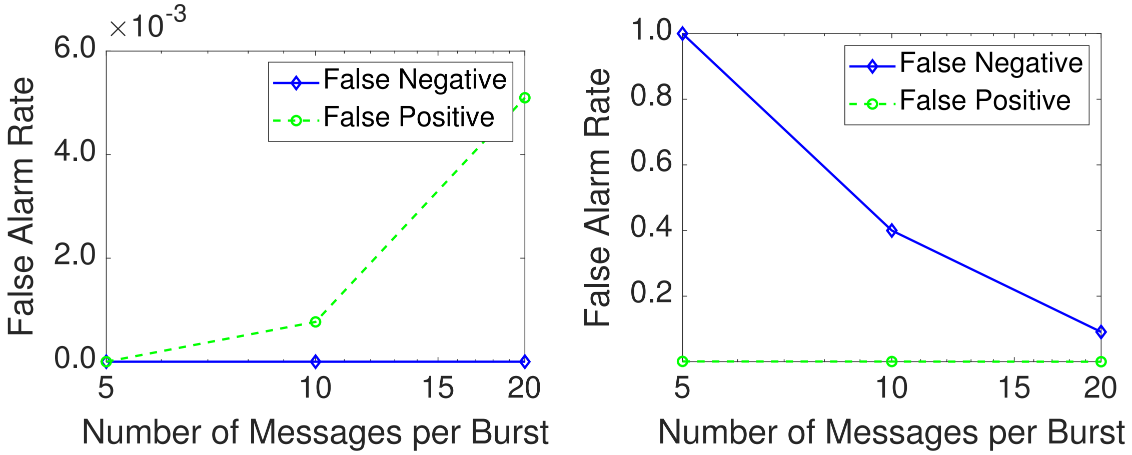


Figure 7: Identification Performance in A Local 4G/LTE Testbed

mat was chosen because CPIM messages are received but can not be displayed on current smartphones such as the iPhone 6S Plus with iOS version 11.4 (15F79). We verified the CPIM message delivery to iPhone and the confirmation from the iPhone on CPIM message delivery through the N6061A LTE protocol logging and analysis software in the testbed. In other words, the victim will receive no indication of the attacks through SMS messages in the CPIM format because the messages will not be shown on the victim's phone at all. We evaluate the performance of the identification algorithm using the following performance metrics: (1) false negative rate defined as the percentage of tests that do not generate cross-correlation values below the decision threshold within the *bound* on the delay when SMS message bursts are sent to the victim's phone and (2) false positive rate defined as the percentage of tests that generates cross-correlation values below the decision threshold when no SMS message bursts are sent to the victim's phone.

Figure 7 shows the identification performance for SMS messages sent in the CPIM format and 3GPP2 format, respectively, for varying lengths of traffic observation. In both experiments, each burst contains 20 SMS messages and the inter-burst time is



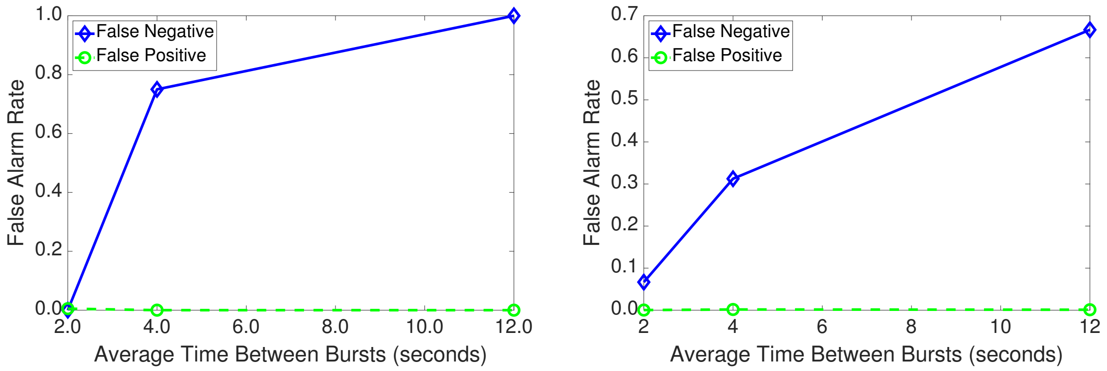
(a) CPIM

(b) 3GPP2

Figure 8: Effect on Number of Messages per Burst

on average two seconds. We can observe from Figure 7 that for messages sent in the CPIM format, both false positive rates and false negative rates are close to zero when the length of traffic is above or equal to 10 minutes. As shown in Figure 7, the false positive rates for 3GPP2 message format is close to zero and the false negative rate can reach 9% when the length of traffic is 30 minutes. From Figure 7, we can also observe the differences in identification performance for the two message formats. We conjecture that the differences are caused by the handling of SMS messages sent in the CPIM format in iOS. It seems to us that the handling of SMS messages in the CPIM format is more resource-consuming and the phone can not interpret the messages properly. So the phone can not display the messages in the CPIM format although the messages are received by the phone. This allows the adversary to increase both the level of stealth and of effectiveness of the attack.

Figure 8 shows the identification performance for different numbers of messages per burst. (The length of traffic used in these experiments is 75 minutes.) In Figure 8a, both the false negative rate and the false positive rate are close to zero for five, 10, and 20 messages per burst when the messages are sent in the CPIM format. For experiments with messages sent in the 3GPP2 format, we observe the significant



(a) CPIM

(b) 3GPP2

Figure 9: Effect on Average Time Between Bursts

decrease in false negative rate when the number of messages per burst increases. The trend indicates that more burstiness leads to better identification performance. In the same set of experiments, the false positive rate remains close to zero when the messages are sent in the 3GPP2 format.

Our next set of experiments focuses on the intervals between two successive bursts. Based on the results from the previous set of experiments, we set the number of message per burst to 20 in the remaining experiments presented in this section. From Figure 9, we observe that the false-negative rates increase significantly when the average amount of time between successive bursts increases. The results are consistent with intuition: The identification performance degrades as the number of message bursts decreases. We can also observe that the false-positive rates remain close to zero for both message formats.

We design a set of experiments to investigate the effect of Inter-Message Time (IMT) of messages within the same burst. Essentially, a smaller IMT means more burstiness, and we expect that more burstiness increases identification performance. This conjecture is verified in this set of experiments. Table 1 shows that both false-negative rate and the false-positive rate improve as IMT decreases. We set IMT to zero for all the remaining experiments presented in this section.

Inter-Message Time (ms)	**False Negative Rate**	**False Positive Rate**
0	0.0000	0.0008
50	0.1000	0.0015

(a) CPIM

Inter-Message Time (ms)	**False Negative Rate**	**False Positive Rate**
0	0.0667	0.0003
50	0.1667	0.0017

(b) 3GPP2

Table 1
Inter-Message Time of Messages within a Burst

Character Count	**False Negative Rate**	**False Positive Rate**
19	0.0000	0.0077
249	0.0000	0.0007

(a) CPIM

Character Count	**False Negative Rate**	**False Positive Rate**
19	0.7500	0.0003
249	0.0667	0.0003

(b) 3GPP2

Table 2
Number of Characters Per Message

Table 2 shows the identification performance for messages sent with different numbers of characters inside of the SMS for both the CPIM and the 3GPP2 format. We examined the collected traffic and found that packets containing messages of different lengths are padded to the same length. The identification performance for the 3GPP2 format, especially the false-negative rate, of the 19 characters per message is worse than that with 249 characters per message. However, the identification performance on the CPIM format does not change significantly with the number of characters in the message. These results indicate that messages in the CPIM format require more compute resources in general than those in the 3GPP2 format, and longer messages in both formats also require more computing resources. These differences result in differences in identification performance.

In the previous experiments, the number of messages in each burst is fixed in each experiment. We vary the number of messages sent in each burst in the next set of experiments. Figure 10 shows the identification performance with different averages

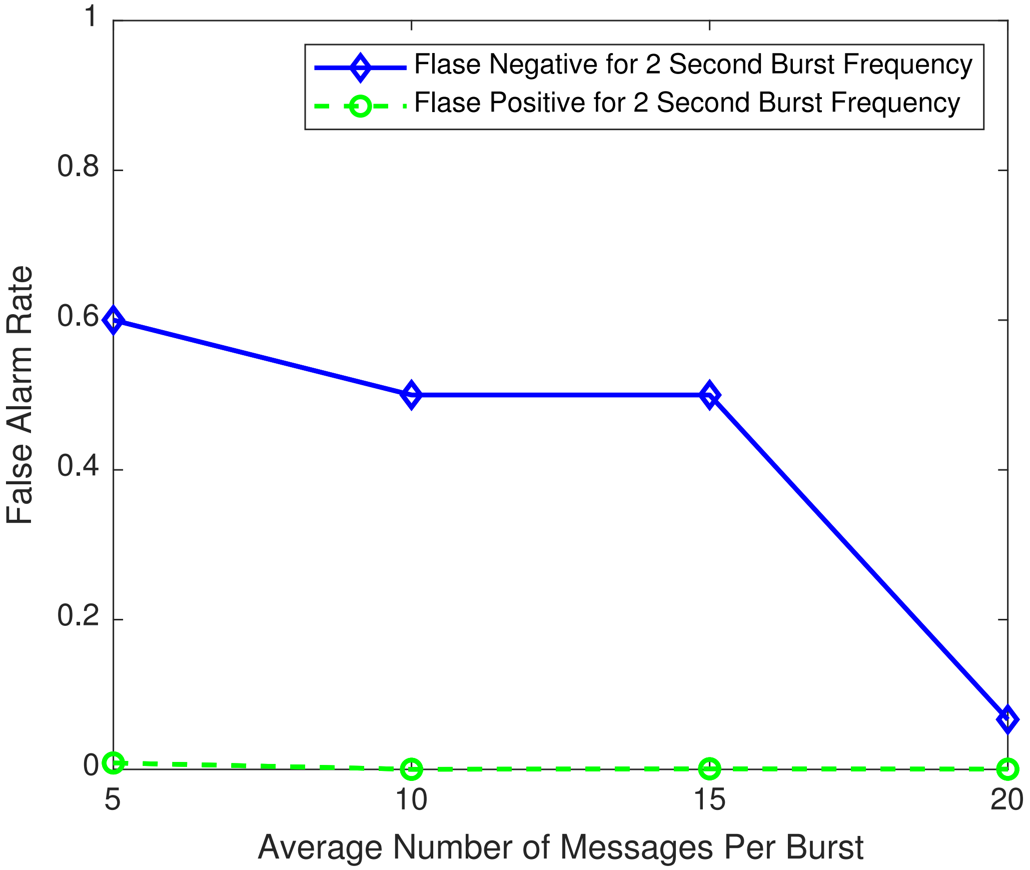


Figure 10: Identification Performance with Random Bursts (Time Between Bursts: 2 Seconds, Format: 3GPP2)

Load	**False Negative Rate**	**False Positive Rate**
10%	0.0000	0.0007
84%	0.3333	0.0006

Table 3
Effect of System Load on Identification Performance (Message Format: CPIM, Time Between Bursts: 2 Seconds, Number of Message per Burst: 20)

of the number of messages in each burst. We observe from Figure 10 that the performance improves with more average numbers of messages. The results are consistent with results on fixed number of messages per burst as shown in Figure 8. The results indicate that in general more messages per burst lead to better identification performance.

Finally, we conduct a set of experiments to investigate the identification performance with different loads on the smartphone system. In the experiments, we evaluate the load on a smartphone system as percentage of load on smartphone system RAM. The load information is obtained from SYSMonitor, an iOS system monitoring application [17]. Table 3 shows the identification performance for different levels of system load. As shown in Table 3, the identification performance is better when the load on a smartphone system is low. This is because the differences caused by the SMS message bursts are more visible to the identification when the system load is low.

Chapter V

Identifying Account Association in a Major Mobile Network

The experiments in the local testbed show the promising results as follows: (1) The SMS message bursts sent to the victim's phone can generate a pattern in the Skype traffic from the victim's phone and the pattern can be successfully detected with the proposed algorithm. (2) The SMS messages sent in the CPIM format will not be displayed on the victim's phone. In other words, the victim receives no indication of being attacked. Given the promising results, we proceed to investigate the account association identification in a major mobile network. The following experiments were conducted on one of the four major mobile networks in United States.

5.1 Experiment Setup

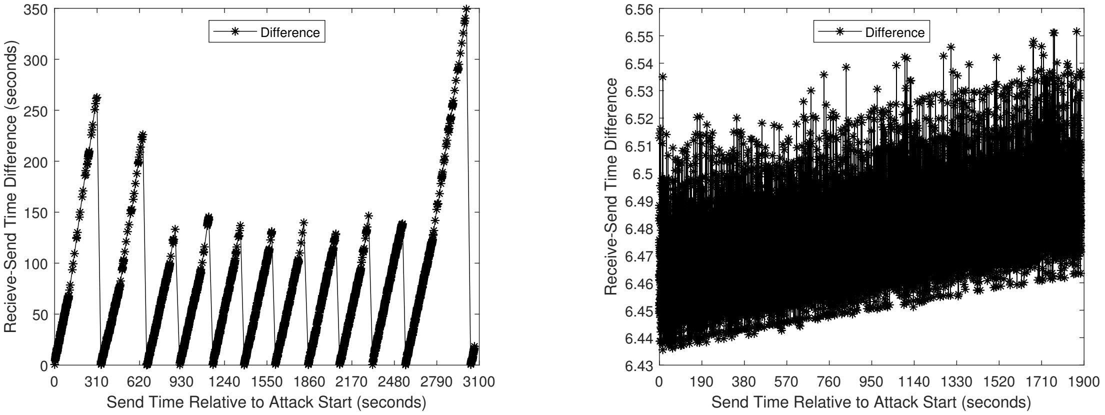
Figure 3 shows the experiment setup over a major mobile network in United States. In this setup, adversary sends SMS message bursts through a LG V20 smartphone. The victim's phone is a Motorola Moto Z Play Droid smartphone. The specifications of both phones are listed in Table 4. Both smartphones are connected to the major mobile network through 4G/LTE connections. We disabled Wi-Fi capabilities on both smartphones so that only 4G/LTE connections are used in the experiments. The version of the Skype software running on the victim's smartphone is 8.36.0.52.

Model	**CPU**	**Memory**	**VoLTE Support**	**Android Version**
Motorola Moto Z Play Droid	Qualcomm® Snapdragon™ 625	3 GB	Yes	7.1.1
LG V20 VS995	Qualcomm® Snapdragon™ 820 MSM8996	4 GB	Yes	8.0.0

Table 4
Phone Specifications for Commercial 4G/LTE Network Testing

5.1.1 Challenges

The mechanisms put in place by service operators to address the practicalities of operating messaging over commercial networks are expected to render the direct application of the attack as described in Section IV ineffective. We expect that the major challenges arise primarily from the need to protect network resources with the help of scheduling and service throttling in two locations: (1) SMS *service centers* in mobile networks are responsible for storing, forwarding, and delivering SMS messages [18]. SMS service centers are also responsible for maintaining the service operation, such as message delivery reports to message senders. Since SMS service centers are shared by many service subscribers, messages from different subscribers may get *queued*, and their processing may need to be *scheduled* by the service center. The major mobile networks also put limits on the message sending rates from their service subscribers [19, 20, 21]. If subscribers exceed these limits, they find their message delivery rates to be *throttled* by the service center. (2) The *uplink* bandwidth is shared among 4G/LTE service subscribers, and scheduling protocols are used for resource control [22]. This leads to queuing and delays. Figure 11 shows the delay between message sending time at the adversary's smartphone and corresponding message receiving time at the victim's smartphone. The delay in the commercial 4G/LTE network, as shown in Figure 11a, can vary from 0.2 seconds to about 350 seconds. Similar saw-tooth patterns



(a) Commercial 4G/LTE Network

(b) Local 4G/LTE Testbed

Figure 11: Time Difference Between Sending and Receiving a Text Message

on packet round trip time are also observed in 4G/LTE networks and the saw-tooth pattern is mainly caused by the scheduling protocols as described in [22]. Similarly, the saw-tooth patterns in message delays observed in our experiments can be largely explained by the scheduling in mobile networks. In our experiments, the periodicity is much longer because the scheduling can also happen at the SMS message level. The delay in the local testbed, as show in Figure 11b, only varies from about 6.43 seconds to about 6.55 seconds. The small variation in the SMS message delay is largely because of the small number of smartphones that are connected to the local 4G/LTE network. Figure 12 demonstrates the presence of throttling: Initially, the SMS burst messages are delivered in a burst of 20. As time progresses, the burst size is controlled by the service center, reaching a steady-state of 10 messages per burst. This throttling of the SMS message sending rate has been previously described in [19, 20, 21]. As a result of the scheduling and throttling, the identification algorithm presented in Section IV is largely ineffective in commercial 4G/LTE networks. The main reason why the pattern caused by the SMS message bursts is not detectable in large networks is that the scheduling and throttling of message delivery spread out bursts of SMS messages and make them difficult to detect.

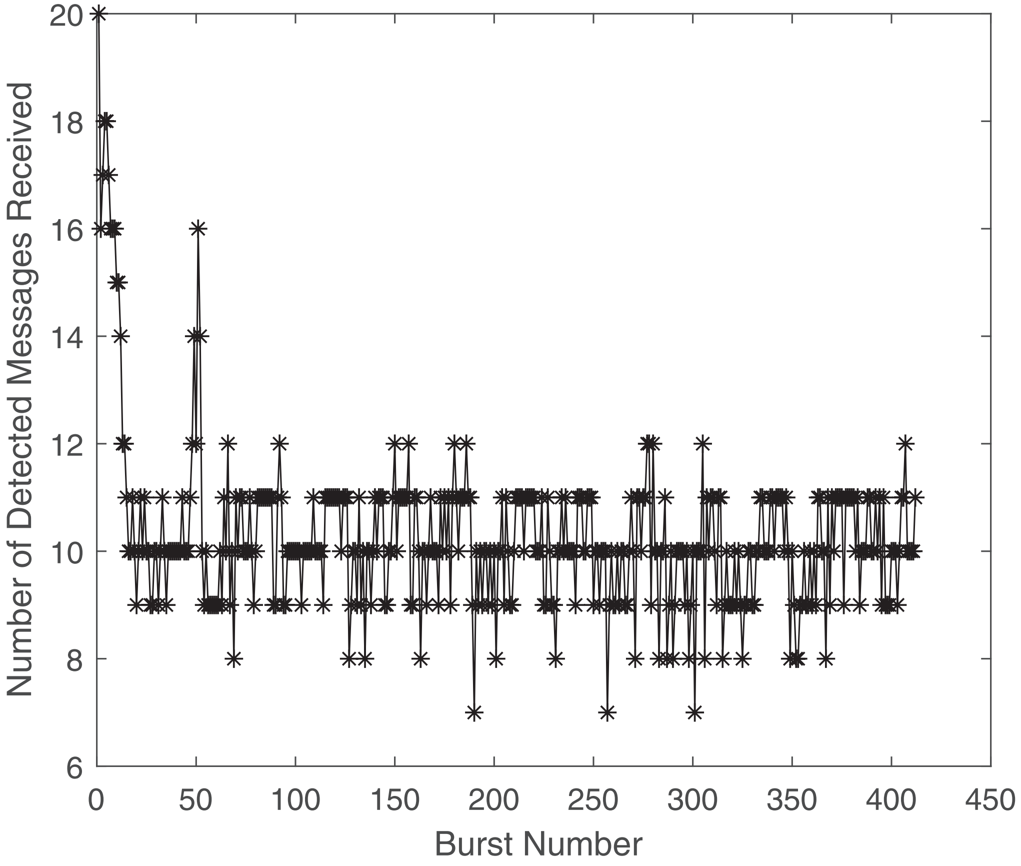


Figure 12: Throttling in A Commercial 4G/LTE Network

5.2 Identification on a Major Mobile Network

Although the scheduling makes the identification algorithm described in Algorithm 1 largely ineffective, the adversary can still take advantage of the scheduling in commercial 4G/LTE networks for the identification on account association. Figure 12 shows that the numbers of the messages actually received by the victim's smartphone and the number of messages sent from the adversary's smartphone is actually 20 per burst. In Figure 12, the decrease over time in the number of messages received shows throttling on the SMS message sending rate as described in [19, 20, 21]

But we can also observe from both Figure 12 and Figure 11a that the scheduling effect and the throttling effect are not happening at the very beginning of SMS message bursts. Instead, the spreading caused by the scheduling effect, as shown in Figure 11a, and the throttling effect will gradually take effect. These effects make sense from the network operator's point of view because network operators need some time to detect the message bursts and then take actions on the SMS messages. Based on these observations, we re-design our identification algorithm to detect the *increase* of burst spreading, rather than the burst interference directly, such as done in Algorithm 1. The resulting algorithm is described in Algorithm 3[1]. This algorithm leverages information about the scheduling in commercial 4G/LTE network and takes advantage of the spreading and throttling caused by the scheduling for identification.

The major difference between the algorithm for the local testbed (Algorithm 1) and the algorithm for the commercial 4G/LTE network (Algorithm 3) is the new variable $eff_{duration}$ in Algorithm 3. The variable $eff_{duration}$ is the minimal length of all the SMS message bursts in time. It is essentially the duration during which SMS messages can possibly generate patterns that can be used for identification. SMS messages sent in the same burst but beyond the duration $eff_{duration}$ are too spread out and cannot significantly contribute to the pattern generation.

The array b_{len} in Algorithm 3 records the number of SMS messages sent during the effective duration $eff_{duration}$. Essentially, b_{len} indicates the degree of burstiness of each SMS message burst. More burstiness can lead to more interference with Skype traffic, and this in turn leads to more obvious patterns that can be used for the identification. Similarly as for Algorithm 1, if the pattern is generated by the SMS message bursts, the array b_{len} and the array $traf$ will be highly negatively correlated. So the Decision function as presented in Algorithm 2 can detect the pattern in the same way it does in Algorithm 1. We note that, in contrast to Algorithm 1, Algorithm 3 does not need to identify intervals without burst interference in order to compute the cross correlation. Instead, it relies on the later burst intervals, which have been spread

[1] Both the CrossCorrelation function and the Decision function used in Algorithm 3 are the same as the corresponding functions used in Algorithm 1.

```
input : $len_{burst}$ - the length of the SMS message burst traffic available for
        detection, $len_{traf}$ - the length of traffic collected at the data collec-
        tion point, array $b_{begin}$ with $b_{begin}[i]$ indicating the beginning time
        of the $i$th SMS message burst, array $b_{end}$ with $b_{end}[i]$ indicating the
        ending time of the $i$th SMS message burst, $bound$ - the bound on
        the delay between the sending time of a burst of SMS messages and
        the arrival time of the corresponding pattern observed at the data
        collection point , $inc$ - step increase;
output: $dec$ - Detection decision, 1 means detected, 0 means not;
for $i \leftarrow 1$ to the length of array $b_{begin}$ do
    $b_{duration}[i] \leftarrow b_{end}[i] - b_{begin}[i]$
end
$eff_{duration} \leftarrow$ minimum of the array $b_{duration}$ ;
$t \leftarrow b_{end}(1)$ ;
$i \leftarrow 1$;
while $t < len_{burst}$ do
    $b_{len}[i] \leftarrow$ the number of SMS messages sent during interval
      $[b_{begin}(i), b_{begin}(i) + eff_{duration}]$;
    $i \leftarrow i + 1$;
    $t \leftarrow b_{end}[i]$;
end
// counting number of effective messages sent in each burst
$shift \leftarrow 0$;
$t \leftarrow b_{end}[1]$;
$j \leftarrow 1$;
while $shift + b_{end}[n] < len_{traf}$ do
    for $i \leftarrow 1$ to $n$ do
        $traf[i] \leftarrow$ the number of packet arrivals at the data collection point
          during interval $[b_{begin}[i] + shift, b_{begin}[i] + shift + eff_{duration}]$;
        // For simplification, we assume that the traffic collected
           at the data collection point is long enough so that the
           array $traf$ and $b_{len}$ can be of the same length.
    end
    $corrval[j] \leftarrow$ CrossCorrelation($b_{len}[1..n]$, $traf[1..n]$);
    $j \leftarrow j + 1$;
    $shift \leftarrow shift + inc$;
end
$mean_{corr} \leftarrow$ mean of array corrval;
$std_{corr} \leftarrow$ stand deviation of array corrval;
$min_{corr} \leftarrow$ the minimum of $corrval[1..\lceil \frac{bound}{inc} \rceil]$ ;
// $\lceil \rceil$ denotes the ceiling function
$dec \leftarrow$ Decision($min_{corr}, mean_{corr}, std_{corr}$);
```

Algorithm 3: Identification Algorithm for a Commercial 4G/LTE Network (The Decision function is the same as shown in Figure 2.)

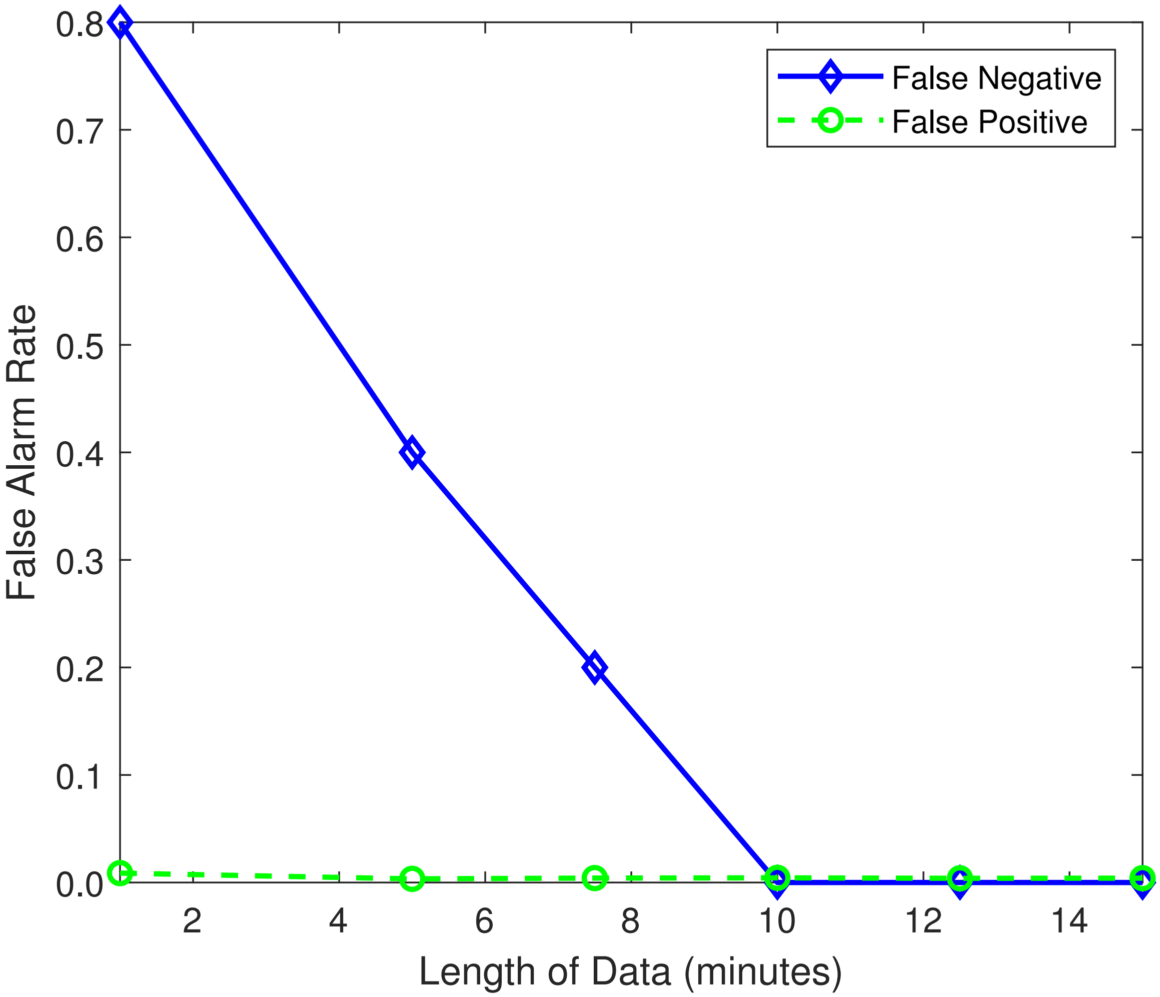


Figure 13: Identification Performance in A Commercial 4G/LTE Network

enough by the network to lead to the negative correlation values needed to identify burst interference.

5.3 Identification Performance

We evaluated the identification algorithm in Algorithm 3 with the same set of performance metrics described in Section IV. Figure 13 shows the identification performance when the number of messages per burst is 20 and the average time between message bursts is two seconds. We can observe from Figure 13 that the false positive rate is close to zero. The false negative rate can reach around 20% when the traffic length

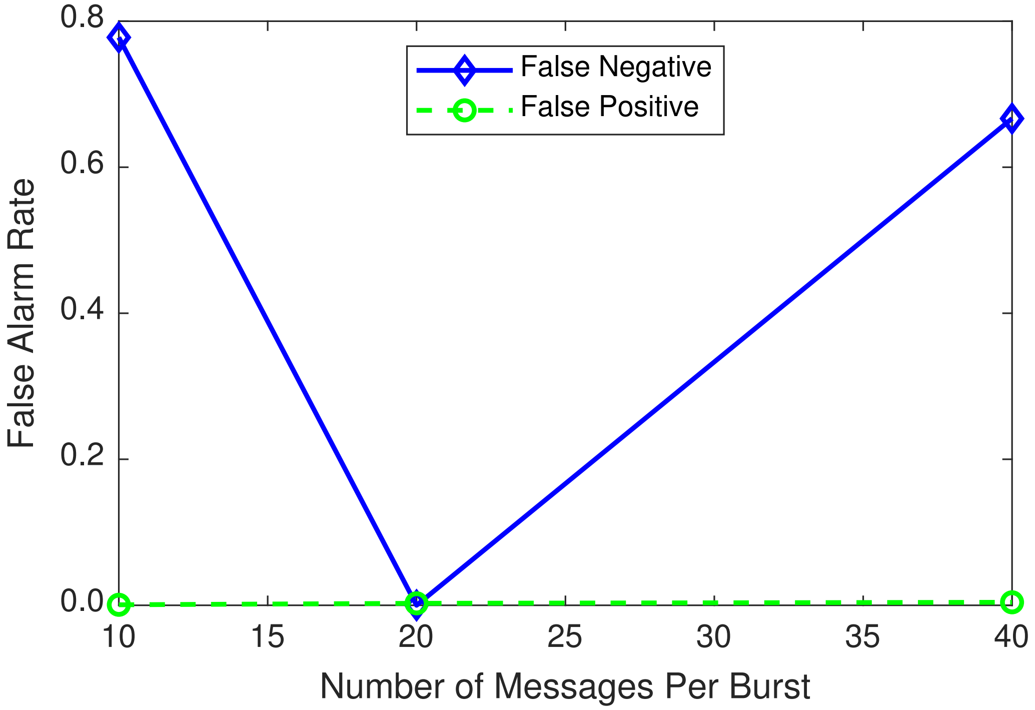


Figure 14: Effect on Number of Messages per Burst

is about 7.5 minutes long and the false negative rate can reach around 0% when the traffic length is 10 minutes or longer.

Figure 14 shows false alarm rates for different number of messages per burst. Both the false positive rate and the false negative rate can approach zero when the number of messages per burst is 20. When the the number of messages per burst is 10, the false negative rate is high because interference caused by the message bursts is not strong enough. The false negative rate is also high when the number of message per burst is 40. The high false negative rate is because of a large amount of message losses due to the limit on the message rate. Figure 15 shows identification performance on the average time between bursts. The length of traffic used in this set of experiments is 60 minutes. Figure 15 shows that both the false positive rate and the false negative rate will be close to zero when the average time between bursts is less than or equal to 4 seconds. When the average time between bursts is 10 seconds, the false negative rate is high mainly due to less amount of bursts to interfere with the Skype traffic.

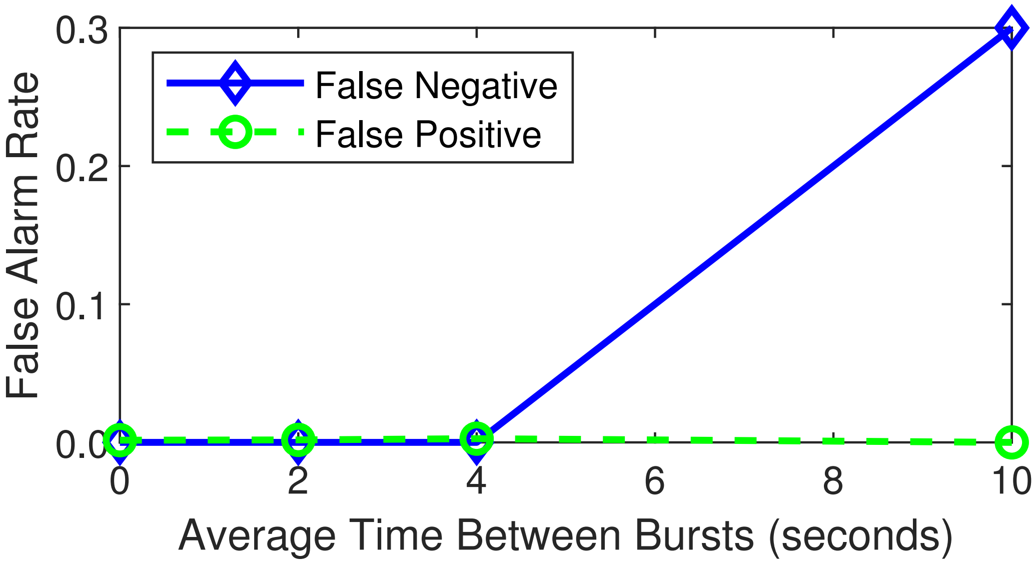


Figure 15: Effect on Average Time Between Successive Bursts

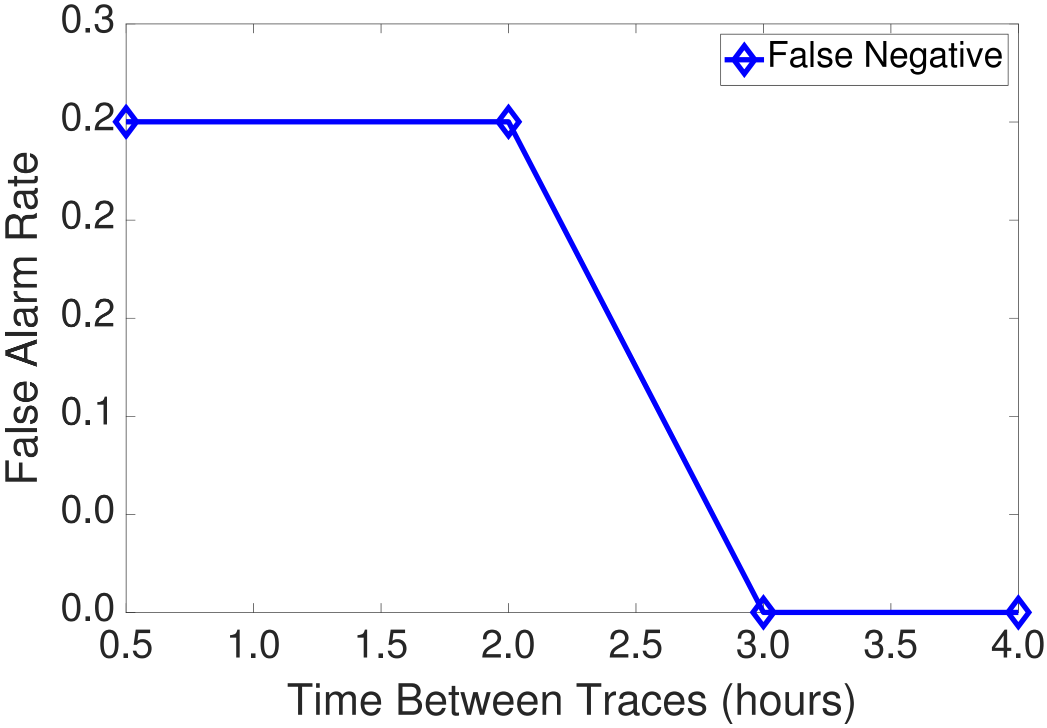


Figure 16: Time Between Experiment Traces

One interesting, and somewhat unexpected, finding from the experiments on the commercial mobile network is related to the *time between experiments*: In this set of experiments, we keep the length of the traces constant (at 75 minutes), as we do the number of messages per burst (at 20) and the average time between two successive bursts (4 seconds). We do, however, vary the time between experiments, that is, the time after which we repeat the experiment. We notice that the performance of subsequent experiments is lower than that of the first experiment. This can be explained because the service center has detected the sources of high-rate SMS message bursts and is now throttling them.

Figure 16 shows that, as the time between two experiments increases, the identification performance actually improves. (We only display false-negative rates, since false-positive rates are constantly close to zero, as shown in the other experiments.) When the time between two successive experiments is longer than three hours, the identification performance largely recovers. This indicates that the commercial mobile network keeps track of their 4G/LTE service subscribers who send SMS messages at too high a rate. Moreover, the results indicate that this information gets either largely discarded or flushed after about three hours.

Chapter VI

Discussion

6.1 Countermeasures

One of the major reasons behind the success of the proposed attacks is the assistance from the mobile network operator: The throttling on the SMS message sending rate actually helps the adversary in generating patterns for confirming the identification. But it is challenging for a commercial 4G/LTE network to defeat the proposed attacks for two reasons. First, the network needs to throttle SMS message sending bursts based on a history and some throttling algorithm actually makes throttling decisions based on weighted average of a sending history [23]. So the network needs some time to take actions on message bursts from a phone. Second, SMS is a store and forward service. A commercial mobile network usually has millions of subscribers. For scalability, it is desired for the network to delivery SMS messages as soon as possible. Because of the two reasons, it is better to implement countermeasures at the phone side. The fundamental reason for the success of the proposed attacks is at the phone side: (1) The phone prioritizes VoLTE packets, more specifically, SMS packets in the proposed attacks over normal IP packets. The prioritization is necessary for Quality of Service of VoLTE services. But the prioritization will also enable the proposed attacks. (2) The SIP protocol used for transporting SMS packets is a relatively resource-consuming protocol [24, 25]. (3) Smartphones are generally resource-constrained devices mainly because of its small form factor. In our future work, we plan to investigate approaches to mitigate the attacks and in the mean time without significant degradation on QoS of VoLTE packets.

Chapter VII

Related Work

7.1 Vulnerabilities of SMS and 4G/LTE

Traynor *et al.* [26] and Enck *et al.*[27] proposed Denial of Service (DoS) attacks using SMS on a GSM (2G) network. The proposed attacks can be launched by sending several SMS messages back-to-back in 2G networks. Since 2G and 3G networks were never originally designed to handle SMS due to their architecture, such an attack causes a complete denial of service to either the entire network or to the victim phone if too many SMS messages are sent. It was additionally observed in [26] that SMS messages could easily suppress voice communication services in 2G and 3G phones because of how SMS was built into the 2G and 3G architectures. Traynor *et al.* [26] and Enck *et al.*[27] provide recommended solutions to mitigate these DoS attacks, which include the separation of IMS and control services. Current 4G/LTE networks separate SMS and voice from the control plane to the data plane and bearers are used for Quality of Service (QoS) of SMS messages and voice calls [12, 13, 14, 15].

Even without rooting a phone, malicious 4G/LTE users can still preform simple attacks to a victim through SMS alone over 4G/LTE as described in [21]. Tu *et al.* [21] found that despite the shift to 4G/LTE, SMS could be exploited through malicious applications, send spoofed SMS messages, send unnecessary spam messages to the IMS service, hijack accounts linked via SMS, make donations through one's bill by a rogue SMS-capable application signing up for an unauthorized donation, and signing up via a rogue SMS-capable application sending an SMS to initialize an unauthorized service subscription.

Shaik *et al.* [28] studied how a rogue Evolved Node Base Station (eNodeB) can be used to attack unsuspecting victims. With the right equipment and information, an adversary can set up a fake eNodeB and collect information about the user, such as the user's location. The adversary would also have the option of denying certain users certain network services. In particular, Shaik *et al.* [28] focused on how connecting to a rogue eNodeB could easily leak user information, as well as provide information about the area of coverage of other surrounding eNodeBs. Due to how IMS is set up in an 4G/LTE network, our experiment could also be applied to a rogue eNodeB and in essence be even more effective.

Kim *et al.* [25] focused on the exploits and vulnerabilities of IMS with a focus on VoLTE. In particular, it was found that early VoLTE adaptations had several security flaws. Some of these flaws included identification spoofing and potential free data channels, which can lead to denial of service attacks and over-billing, especially if a legitimate user has access to control the phone's application processor.

IP Multimedia Subsystem (IMS) uses the physical downlink and uplink shared channels to transmit its services. These channels are responsible for sharing downlink and uplink resources in an 4G/LTE connection that are not control related, nor Internet related. Lichtman *et al.* [29] researched on the different channels in 4G/LTE, including the ones previously mentioned, and the the vulnerabilities of each channel. They found by targeting certain signals and channels, an adversary could essentially jam a network connection.

7.2 Remote Traffic Analysis

The attacks proposed in this paper are related to previous research on remote traffic analysis, which aims at disclosing sensitive information through remote probing. Gong *et al.* [30] showed that traffic patterns leaked through side channels can be used to recover important semantic information. In particular, Gong *et al.* found that they could gather such information remotely by sending probes to relay critical network timing information without requiring direct access to the connection itself.

Murdoch and Danezis [31] investigated how traffic analysis can also be used for deanonymizing the connections between two clients. Murdoch and Danezis used traffic analysis to reveal the nodes that make up a Tor connection.

Kadloor *et al.* [32] studied how packet-based networks are vulnerable to remote traffic analysis attacks, again by using timing probes. Such attacks would be mounted in any scenario where a shared routing resource exists among users. They found that a real-world attack successfully compromised the privacy of a user without requiring significant resources in terms of access, memory, or computational power.

Chapter VIII

Conclusion

In this paper we direct the attention to an emerging class of attacks that is enabled by the increase deployment of platforms that run a variety of different services in an integrated fashion. Such platforms enable attacks to leverage information on one service to attack, or at least infer information about, users on another service. We call these attacks "cross-service attacks". We describe and evaluate an example of such an attack, the account association attack in 4G/LTE networks. The goal of this attack is to associate a target mobile phone number with a user account of an IP-based service. In this example, the adversary launches the account association attack by sending SMS messages to the target phone number and by analyzing patterns in traffic related to the IP account. We evaluate the proposed attacks in both a local 4G/LTE testbed and a major commercial 4G/LTE network. Our extensive experiments show that the proposed attacks can successfully identify account association with both false negative and false positive rates close to zero. Our experiments also indicate that proposed attacks can be launched in a way that the victim receives no indication of being under such an attack.

REFERENCES

[1] Sophia Antipolis. Evolution to lte report: 4g market and technology update. Specification 24.341, GSA, Jan 2017.

[2] Lte subscriptions to 4q 2018: Gsa-ovum update. Presentation, GSA, Jan 2019.

[3] Graham Klyne and Derek Atkins. Common Presence and Instant Messaging (CPIM): Message Format. Technical Report 3862, August 2004. URL `https://rfc-editor.org/rfc/rfc3862.txt`.

[4] Ed Elkin. The secret value of volte, 2014. URL `http://blog.tmcnet.com/next-generation-communications/2014/04/the-secret-value-of-volte.html`.

[5] J. Rosenberg. *RFC3261: SIP: Session Initiation Protocol*, Jun 2002.

[6] J. E. Vargas Bautista, S. Sawhney, M. Shukair, I. Singh, V. K. Govindaraju, and S. Sarkar. Performance of cs fallback from lte to umts. *IEEE Communications Magazine*, 51(9):136–143, Sep. 2013. ISSN 0163-6804. doi: 10.1109/MCOM.2013.6588662.

[7] Android developers: Smsmanager. URL `https://developer.android.com/reference/android/telephony/SmsManager`.

[8] Tom Berson. Skype security evaluation. Technical Report ALR-2005-031, Anagram Laboratories, October 2005.

[9] C. J. Fourie and W. J. Perold. On using genetic algorithms to optimize high frequency superconducting digital circuits for optimal yield. In *IEEE AFRICON. 6th Africon Conference in Africa,*, volume 2, pages 505–510 vol.2, Oct 2002. doi: 10.1109/AFRCON.2002.1159959.

[10] Salman A. Baset and Henning G. Schulzrinne. An analysis of the skype peer-to-peer internet telephony protocol. Technical report, 2004.

[11] A. Awan and R. Venkatesan. Design and implementation of enhanced crossbar cioq switch architecture. In *Canadian Conference on Electrical and Computer Engineering 2004 (IEEE Cat. No.04CH37513)*, volume 2, pages 1045–1048 Vol.2, May 2004. doi: 10.1109/CCECE.2004.1345297.

[12] C. Gessner and O. Gerlach. Voice and sms in lte. White Paper 1e, Rohde and Schwarz, May 2011.

[13] Sophia Antipolis. 3gpp: Technical specification group services and system aspects: Support of sms over ip networks; stage 3 (release 7). Specification 24.341, 3GPP, Jun 2018.

[14] Sophia Antipolis. 3gpp: Technical specification group services and system aspects: Ip multimedia subsystem (ims): Stage 2 (release 15). Specification 23.228, 3GPP, Sep 2018.

[15] Dispelling lte myths. URL `http://www.3gpp.org/news-events/3gpp-news/1268-Dispelling-LTE-Myths`.

[16] H. Garudadri. *RFC4393: MIME Type Registrations for 3GPP2 Multimedia Files*, Mar 2006.

[17] Zehui Wang. SYSMonitor - System Status Wgt, Nov 2014. URL `https://apps.apple.com/us/app/sysmonitor-system-status-wgt/id937687645`.

[18] Voice over lte: The new mobile voice. Technical Report m2012042721, Alcatel-Lucent, April 2012. URL `https://www.tmcnet.com/tmc/whitepapers/documents/whitepapers/2013/6685-voice-over-lte-new-mobile-voice-inspire-new.pdf`.

[19] At&t messaging toolkit: Your everything mobile communications. Technical Report AB-2241-01, AT&T, May 2017. URL https://www.business.att.com/content/dam/attbusiness/briefs/messaging-toolkit-everything-mobile-communications.pdf.

[20] Verizon: Network api sla limits, 2018. URL \url{http://lte.vzw.com/content/vdc/en/verizon-tools-apis/verizon_apis/network-api-direct-development/napi_technical_resources/napi_sla.html}.

[21] Guan-Hua Tu, Chi-Yu Li, Chunyi Peng, Yuanjie Li, and Songwu Lu. New security threats caused by ims-based sms service in 4g lte networks. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS '16, pages 1118–1130, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4139-4. doi: 10.1145/2976749.2978393. URL http://doi.acm.org/10.1145/2976749.2978393.

[22] Ilija Hadžić, Yoshihisa Abe, and Hans C. Woithe. Edge computing in the epc: A reality check. In *Proceedings of the Second ACM/IEEE Symposium on Edge Computing*, SEC '17, pages 13:1–13:10, New York, NY, USA, 2017. ACM. ISBN 978-1-4503-5087-7. doi: 10.1145/3132211.3134449. URL http://doi.acm.org/10.1145/3132211.3134449.

[23] Cyril Goutte. Adaptive spam message detector, 06 2006. URL https://patents.google.com/patent/US20060123083A1/en.

[24] M. Cortes, J. R. Ensor, and J. O. Esteban. On sip performance. *Bell Labs Technical Journal*, 9(3):155–172, 2004. ISSN 1538-7305. doi: 10.1002/bltj.20048.

[25] Hongil Kim, Dongkwan Kim, Minhee Kwon, HyungSeok Han, Yeongjin Jang, Dongsu Han, Taesoo Kim, and Yongdae Kim. Breaking and fixing volte:

Exploiting hidden data channels and mis-implementations. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, CO, USA, October 12-16, 2015*, pages 328–339, 2015. doi: 10.1145/2810103.2813718. URL `https://doi.org/10.1145/2810103.2813718`.

[26] Patrick Traynor, William Enck, Patrick Mcdaniel, and Thomas La Porta. Mitigating attacks on open functionality in sms-capable cellular networks. In *ACM MobiCom '06*, pages 182–193, 2006.

[27] William Enck, Patrick Traynor, Patrick Mcdaniel, and Thomas La Porta. Exploiting open functionality in sms-capable cellular networks. In *Proceedings of the ACM Conference on Computer and Communication Security (CCS*, pages 393–404. ACM Press, 2005.

[28] Altaf Shaik, Ravishankar Borgaonkar, N. Asokan, Valtteri Niemi, and Jean-Pierre Seifert. Practical attacks against privacy and availability in 4g/lte mobile communication systems. *CoRR*, abs/1510.07563, 2015. URL `http://arxiv.org/abs/1510.07563`.

[29] Marc Lichtman, Jeffrey H. Reed, T. Charles Clancy, and Mark Norton. Vulnerability of LTE to hostile interference. *CoRR*, abs/1312.3681, 2013. URL `http://arxiv.org/abs/1312.3681`.

[30] Xun Gong, Nikita Borisov, Negar Kiyavash, and Nabil Schear. Website detection using remote traffic analysis. In Simone Fischer-Hübner and Matthew Wright, editors, *Privacy Enhancing Technologies*, pages 58–78, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. ISBN 978-3-642-31680-7.

[31] S. J. Murdoch and G. Danezis. Low-cost traffic analysis of tor. In *2005 IEEE Symposium on Security and Privacy (S P'05)*, pages 183–195, May 2005. doi: 10.1109/SP.2005.12.

[32] S. Kadloor, X. Gong, N. Kiyavash, T. Tezcan, and N. Borisov. Low-cost side channel remote traffic analysis attack in packet networks. In *2010 IEEE International Conference on Communications*, pages 1–5, May 2010. doi: 10.1109/ICC.2010.5501972.

APPENDIX

ACRONYMS

CSU	Cleveland State University
LTE	Long Term Evolution
4G	Fourth Generation
SMS	Short Message Service
QoS	Quality-of-Service
SIP	Session Initiation Protocol
IMS	IP Multimedia Subsystem
UE	User Equipment
E-UTRAN	Evolved Universal Terrestrial Radio Access Network
eNodeB	Evolved Node Base Station
EPC	Evolved Packet Core
VOIP	Voice over IP
VoLTE	Voice over LTE
MME	Mobile Management Entity
HSS	Home Subscriber System
SGW	Serving Gateway
PGW	Packet Data Gateway
PCRF	Policy and Charging Rule Function
CPIM	Common Presence and Instant Messaging
3GPP	3rd Generation Partnership Project
GSA	Global Mobile Supplier Association
5G	Fifth Generation
3GPP2	3rd Generation Partnership Project Message Format 2
DoS	Denial of Service