Sonar: Detecting SS7 Redirection Attacks With Audio-Based Distance Bounding

Christian Peeters*, Hadi Abdullah*, Nolen Scaife*, Jasmine Bowers*,
Patrick Traynor*, Bradley Reaves†, Kevin Butler*

*University of Florida
{*cpeeters, hadi10102, scaife, jdbowers*}*@ufl.edu,*
{*traynor, butler*}*@cise.ufl.edu*

†North Carolina State University
bgreaves@ncsu.edu

***Abstract*—The global telephone network is relied upon by billions every day. Central to its operation is the Signaling System 7 (SS7) protocol, which is used for setting up calls, managing mobility, and facilitating many other network services. This protocol was originally built on the assumption that only a small number of trusted parties would be able to directly communicate with its core infrastructure. As a result, SS7 — *as a feature* — allows all parties with core access to redirect and intercept calls for any subscriber anywhere in the world. Unfortunately, increased interconnectivity with the SS7 network has led to a growing number of illicit call redirection attacks. We address such attacks with Sonar, a system that detects the presence of SS7 redirection attacks by securely measuring call audio round-trip times between telephony devices. This approach works because redirection attacks force calls to travel longer physical distances than usual, thereby creating longer end-to-end delay. We design and implement a distance bounding-inspired protocol that allows us to securely characterize the round-trip time between the two endpoints. We then use custom hardware deployed in 10 locations across the United States and a redirection testbed to characterize how distance affects round trip time in phone networks. We develop a model using this testbed and show Sonar is able to detect 70.9% of redirected calls between call endpoints of varying attacker proximity (300–7100 miles) with low false positive rates (0.3%). Finally, we ethically perform actual SS7 redirection attacks on our own devices with the help of an industry partner to demonstrate that Sonar detects 100% of such redirections in a real network (*with no false positives*). As such, we demonstrate that telephone users can reliably detect SS7 redirection attacks and protect the integrity of their calls.**

I. Introduction

Telephony systems represent the most ubiquitous and trusted communications infrastructure in the world. In both the developed and developing worlds, these networks offer reliable audio connections that allow their subscribers to chat with distant family members, perform important business transactions and even exchange highly sensitive information. Many sectors of the global economy, especially finance and infrastructure, rely on telephony systems as a critical fallback to ensure that high value transactions or significant changes to operation indeed originate from an authorized party.

The content of calls over telephone networks has been viewed as secure from most adversaries solely due to limited access to core networks. Only a small number of providers and governments have historically been able to access the underlying Signaling System 7 (SS7) network, which is used for providing translations between cellular networks. SS7 was designed on a foundation of implicit trust — that is, anyone with access is authorized to make any request. This assumption is convenient, especially in a mobile context, where providers other than a users' home network may legitimately need to quickly redirect traffic to a roaming client. Such features could also be used to maliciously redirect traffic intended for a specific user, to ensure that its delivery path included a system controlled by an adversarial party interested in intercepting such traffic. Such *SS7 redirection attacks* were long assumed to be rare; however, deregulation in the 1990s [1] and the increased diversity of access technologies to these networks have eliminated this assumption. The impact of these changes has been obvious. As recent media coverage demonstrates, SS7 redirection attacks have become increasingly common and are rumored to be a favorite means of eavesdropping by intelligence agencies [2], [3].

We develop Sonar,[1] a mechanism for detecting SS7 redirection attacks at call endpoints. Sonar relies on the key insight that SS7 redirection attacks increase the distance that call audio travels, thereby increasing audio latency. Sonar detects SS7 redirection attacks by expanding on techniques developed for line-of-sight distance bounding from the wireless community and uses the audio channel between two endpoints to transmit challenge/response messages to securely estimate the round trip time (RTT) over a multi-hop network.

This paper makes the following contributions:

- **Acoustic Distance Bounding:** We design and implement a distance bounding-inspired protocol that we refer to as Rapid Audio Exchange (RAE). This protocol relies on a series of audio tones to implement a challenge-response protocol based on the work by Hancke and Kuhn [4]. Unlike traditional wireless distance bounding protocols, RAE is designed to operate in a multihop adversarial telephony network in which the endpoints are honest. To

[1] Traditionally, Sonar is used to map locations that are inaccessible or opaque to the eye (e.g., underwater). Like real sonar systems, we rely on short bursts of audio to map connections.

DOI 10.1109/SP.2018.00006

our knowledge, this is the first technique to detect SS7 redirection attacks in an end-to-end fashion.

- **Distance vs "Mouth to Ear" RTT Time:** Intuitively, the time to traverse a network is dependent upon the speed of light/propagation and the distance between the two endpoints. This relationship has not previously been characterized in a security context for telephony networks. We examine the impact of distance on RTT by building custom hardware to measure RTT at ten locations across the United States.
- **Demonstrate Attack Detection Capabilities:** We emulate SS7 redirection attacks by implementing a testbed that intercepts our calls and redirects them via VPN to locations across the world before delivering them to their intended cellular endpoints. We show that Sonar has a 70.9% true positive rate (0.3% FP) for calls varying attacker proximity (300–7100 miles), and a 97.0% true positive rate (0.3% FP) for all US calls redirected outside of North/Central America. These tests allow us to develop a conservative model before performing real attacks.
- **Validate detection with real SS7 rerouting attacks:** We validate our testbed by legally conducting SS7 attacks with an industry partner. We find that our testbed measurements are conservative, and demonstrate the ability to identify 100% of our own redirection attacks with a 0% false positive rate. We believe these results are indicative of expected performance when deployed.

While many are calling for SS7 to be "made secure" [5], it is unlikely that this global infrastructure will be fundamentally redesigned and redeployed in the foreseeable future. As such, the most practical solution in the short and medium terms is the development of tools that allow end users to be made aware of when their calls are likely experiencing malicious redirection.

The remainder of this paper is organized as follows: Section II provides background information on telephony systems and their weaknesses; Section III formally describes our hypothesis; Section IV details our threat model; Section V details our acoustic distance bounding protocol; Section VI explains our experimental setup; Section VII presents our results that both confirm our hypothesis about additive delay caused by such attacks and that our proposed can detect them; Section IX discusses additional concerns related to Sonar; Section VIII provides detail on our execution of actual SS7 rerouting attacks and how they compare to our simulation; Section X covers related work and Section XI provides concluding remarks.

II. Background

This section provides background on how the SS7 protocol unifies the global telephone network and how network-based redirection attacks using SS7 can affect call routing.

A. Signaling System 7

The Public Switched Telephone Network (PSTN) is a diverse system connecting a variety of technologies including traditional landline, cellular, and VoIP. Each of these technologies define their own protocols to connect end devices (e.g., phones) as well as protocols for delivering calls, text messages, data, and other value added services. These interconnections are shown in Figure 1.

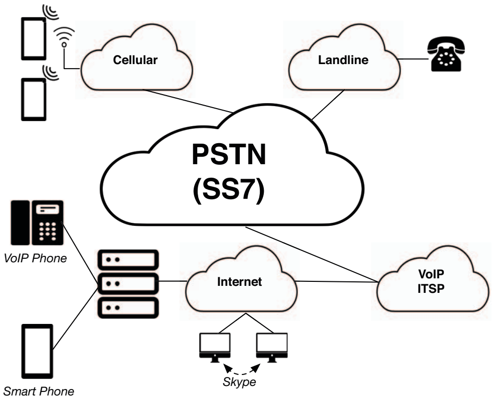

Fig. 1: SS7 connects individual telephone networks to form the global PSTN.

While access protocols vary substantially at the network edge, call signaling, mobility management, and many other network features are provided by a protocol suite known as Signaling System Number 7 (SS7). SS7 forms an all-digital communications network that is used for communications between telephone switches, important databases[2], and other core network entities. SS7 is used to set up and tear down phone calls and manage the location of mobile phones facilitate delivery of calls, messages, data, and roaming. Note that SS7 does not carry call content — only signaling information. SS7 is not only important for the many core network features that it facilitates; it is also important because it acts as a "lingua franca" between carriers, who may support different access technologies (e.g., landline and cellular).

SS7 Security Unfortunately, SS7 has many critical security vulnerabilities. The primary design flaw is that SS7 has no mechanisms for authenticating messages or determining whether a particular network endpoint is authorized to send that message. As a result, any participant in the SS7 core network can claim to send messages on behalf of another network entity and can claim to be authoritative for a given subscriber. A consequence of this is that any SS7 network node can send a query to locate any mobile subscriber – trivially enabling powerful, high-resolution tracking of a user [6]. Additionally, SS7 provides facilities to completely deny service to the endpoint. These issues are not simply bugs or oversights, but rather the result of intentional design decisions. The ability for any network to speak on behalf of any subscriber is essential to support mobility (especially roaming) and number portability. For example, a roaming subscriber may wish to forward calls to a landline; the roaming network must establish call forwarding for that subscriber. *These vulnerabilities allow any SS7 core entity to track and control the flow of calls and*

[2] These databases include caller ID names (CNAM), mappings of toll-free numbers (e.g. 1-800-XXX-XXXX) to canonical numbers, and mobile subscriber registries like the HLR in mobile networks.

text messages of any subscriber worldwide.

Some carriers attempt to block clearly malicious attack messages at the network edge using GFW firewalls, but many still do no filtering at all. For example, carriers in the United States do *not* block attack messages and provide no protection against SS7 redirection attacks [7]. While solutions to this problem exist, they require carriers to implement them, which they are either unable or unwilling to do. Our solution can be implemented by users without cooperation from carriers. The problem is not easy to solve: many "attack" messages are simply abuses of functionality. These network messages exist because a carrier intending to deliver a call to a mobile phone needs to know to which carrier to send the call. While ingress filtering of sensitive SS7 messages (like those that redirect calls) is an important first step, the design of the phone network requires faith in call routing and provides zero guarantees.

B. SS7 Redirection Attacks

The openness of the SS7 network, combined with limited controls for authentication and authorization, means that several legitimate network functions can be abused to redirect calls. These attacks are stealthy, and while it is difficult to know how often they occur, there is evidence that these attacks are increasingly common [2], [8], [9].

The idea behind these redirection attacks is to cause a call to or from the victim to be sent to the attacker instead. The attacker can then forward the call to another destination or simply answer the call. If the attacker answers the call, the attacker can also place a call to the other legitimate call party and patch the audio between the two calls, allowing interception and eavesdropping of the apparently legitimate call. There are several network functions that facilitate this.

The first mechanism is the simplest: spoofing a call forwarding request. In this attack, the attacker spoofs a call forwarding registration message from the victim to the victim's home network. This message forwards the call from the victim to an attacker controlled number. If the attacker wishes to forward the call to the victim (to completely intercept and eavesdrop or tamper with the call) once the call is delivered to the attacker, the attacker spoofs a new message to the victim's network to disable call forwarding, and immediately after places a call directly to the victim (possibly spoofing caller ID) [10].

The second mechanism designates an attacker endpoint as authoritative for all calls for a mobile phone, and it allows interception of incoming calls to any victim on a mobile network. Specifically, this is accomplished by sending a message to the victim's network that designates the attacker's SS7 endpoint as the responsible core network switch (i.e., the MSC) for the victim. Because the MSC is responsible for routing incoming calls to the victim, the attacking MSC can redirect the call to an attacker-controlled number [7].

The third mechanism abuses an SS7 feature called CAMEL [7], [10], and it allows interception of a mobile subscriber's incoming and outgoing calls. CAMEL allows home networks to provide services while a phone is roaming [11]. One feature is to intercept dialed numbers and modify them. To exploit this feature, the attacker registers itself as the victim's CAMEL server. When calls are placed, the attacker modifies the number dialed to one under the attacker's control. The attacker can then answer and forward the call audio in a manner similar to the other attacks.

E-Model User Satisfication	Mouth-to-ear-delay/ms
Users Very Satisfied	0 - 200
Users Satisfied	200 - 290
Some Users Dissatisfied	290 - 390
Many Users Dissatisfied	390 - 550
Nearly All Users Dissatisfied	>550

TABLE I: The ITU E-Model User Satisfaction for Speech Applications correlates one-way transmission times with user-perceived audio quality. Testing by the ITU demonstrates that mouth to ear delay of less than 400ms provides satisfactory audio quality to the majority of users.

C. The Role of Convergence

The growth of VoIP has led to a notion of "convergence" of the Internet and the PSTN — that is, that eventually the phone network and the Internet will become one single network. In many cases, voice communications over the Internet have already replaced PSTN-based communications. This includes the use of peer-to-peer voice clients like Signal as well as the use of VoIP to connect to Internet Telephony Service Providers (ITSPs) who provide calling service from the subscriber to the global PSTN. These ITSPs may be entities like Vonage or MagicJack, cable companies, or wholesale service providers. While peer-to-peer VoIP communications are transmitted exclusively through the Internet, calls that transit ITSPs are likely to also be facilitated by the larger PSTN — meaning that VoIP calls are not necessarily protected from attacks against SS7. VoIP infrastructures are also vulnerable to the entire arsenal of Internet-based attacks [12], including attacks on interdomain routing [13] (also discussed later in Section X). However, convergence does not just mean that some phone calls transit the Internet; it also means that much of the non-VoIP core telephony infrastructure has also replaced telephone-network specific technologies (e.g., digital trunks (T1), ISDN, and ATM networks) with IP-based networks. This *vastly lowers the barriers for core network attack* because the core network can be attacked using open source software like OpenSS7 [14] running on Internet-connected commodity hardware.

III. Hypothesis

Call audio delay is strongly correlated to the distance it travels between source and destination. An SS7 redirection attack increases the distance traveled by the call audio and can therefore be detected by measuring the RTT of the audio and comparing it against an expected range for a known distance.

A. Mouth to Ear Delay

Latency measurements in telephony systems differ from those in traditional IP networks. Specifically, latency is measured in terms of "mouth to ear delay," which represents the difference between the time that a caller begins speaking and

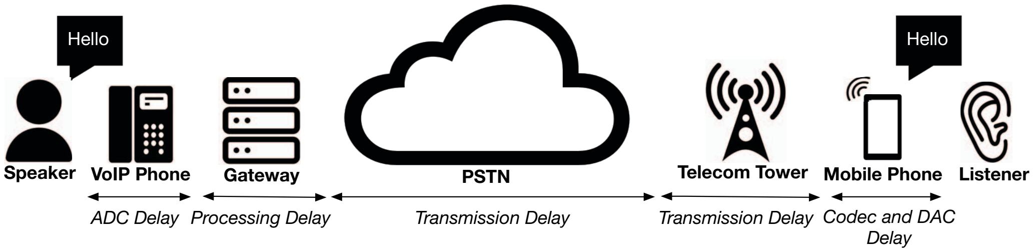

Fig. 2: Mouth-to-ear delay of audio is caused by many factors besides transmission time through the network.

the receiver hears the audio. The ITU G.114 standard provides guidelines via their "E-Model" transmission rating algorithms to delineate mouth to ear delay on perceived call quality [15]. As shown in Table I, one-way mouth to ear delay of less than approximately 400ms results in satisfactory call audio quality for the majority of phone users. This is in stark contrast to traditional IP-based networks, where latency of greater than 100ms is widely viewed as negatively impacting a connection.

Figure 2 provides some intuition regarding additional latency is added in telephony. Computational steps, including analog to digital conversion, network transcoding (i.e., transformation between audio codecs within the network), echo cancellation, and compression all delay audio delivery. These delays are on top of the traditional transmission delays in both the wired and wireless portions of telephony networks.

It is worth noting that the time introduced by the previous example is doubled when measuring RTT. Accordingly, it is not outside the normal range for audio transmitted by a sender, repeated by a receiver and returned to the sender to be *judged as having good quality even with an RTT of nearly 800ms.*

Try This At Home We invite the reader to prove the existence of long mouth to ear delays to themselves. Place a call between any two devices on telecommunication networks that are within reach of one another.[3] Make a sound into one device and listen for it on the other (which should be muted to avoid feedback). Readers will hear a noticeable delay between when the noise is made and when it is played by the other device even though the devices are located near each other.

B. Challenges to Testing the Hypothesis

While it is intuitive that increasing distance correspondingly increases delay, rigorously testing this hypothesis has significant challenges. For instance, telephony networks are largely opaque to outsiders. They do not offer the equivalent of IP network functions such as `traceroute` that would allow researchers to characterize a call path or `ping` to measure RTT. As mentioned above, they also perform significant computation (e.g., transcoding), adding significant additional latency. Routing between destinations may also vary due to changes in the networks and the varying path of the call audio. This changes on a call to call basis and causes inconsistencies in network time. Finally, networks may delay the delivery of audio based on internal priority, quality of service requirements and even segmentation into frames for the air interface. As such, we must create our own mechanisms to characterize *normal* network latency (using the ITU E-model as an independent confirmation of reasonable results), and must accept a higher degree of uncertainty than is traditionally experienced in Internet measurement studies.

IV. Security Model

Sonar detects call redirection by network-based adversaries. In this section, we detail the capabilities of these adversaries. We also identify the goals and capability claims of Sonar and outline telephony attacks that are out-of-scope for this work. We conclude with a discussion contrasting Sonar with traditional distance bounding techniques.

Adversary Capabilities We are concerned with an adversary who seeks to redirect a phone call. By "redirect" we specifically mean "change the routed path of the call". The technical means by which this can be done are extensive, and Sonar is agnostic to the means used to redirect the call to them. We focus on SS7 redirection attacks in this paper, but other redirection attacks (including, but not limited to, BGP rerouting) can also be addressed.

The adversary has a number of capabilities that can frustrate defenders. The adversary can redirect a call to an arbitrary location under their control. The adversary can also arbitrarily modify call audio. This includes producing new sounds (including speech), dropping sounds, or adding noise. Naturally, this includes dropping, modifying, or fabricating Sonar messages sent through the voice channel. We assume the adversary can redirect both incoming and outgoing calls of a target. The adversary also can know the locations of victims with high accuracy (SS7 tracking attacks make this especially practical). As a consequence, we must assume the adversary also knows what latencies are expected for the redirected call. While an adversary can have access to the Sonar system, we assume that the adversary does *not* control either call endpoint. This is actually a trivial assumption because if the adversary controlled the other endpoint, no redirection attack would be necessary. Accordingly, endpoints can trust each other to faithfully participate in the system.

In this work, the adversary is only capable of submitting SS7 messages to the network; these messages cause the call to be redirected to a network node under the attackers control. The adversary does not compromise any core SS7 component to perform this attack. Once the call is received, the adversary connects a new call to the original recipient and retransmits the audio. It is the additional physical distance and decoding

[3] Standard carrier rates may apply.

delay that creates the additional latency we use to detect these attacks. Redirections to other types of networks (e.g., VoIP) *do not reduce the chance of detection*, as gateways between these networks introduce substantially more latency than simple redirection on the PSTN.

Sonar Capabilities and Goals Sonar provides a system and protocol to securely measure audio latency in the face of the above adversary. Sonar provides call endpoints with an accurate measurement of RTT and a decision as to whether that RTT is consistent with the distance of the call. To accomplish this, Sonar includes a cryptographic challenge/response protocol. While the expected distance will vary from call to call, it will be shared with both call parties before the latency measurement begins. Drimer and Murdoch [16] identified a number of attacks against challenge/response distance bounding systems, including the adversary guessing responses, replaying previously used challenges, and using more capable hardware. Sonar is designed to protect against all of these attacks, and this is further discussed in Section V.

Scope of Sonar Sonar is designed to detect SS7 call redirection attacks. As a result, phone network attacks that do not significantly affect audio delay cannot be detected by Sonar, and must be defended against using other methods. Attacks that do not affect audio delay include denial of service attacks, attacks on SMS, and compromised end devices. Of course, if an adversary is located reasonably close to the victim, the associated redirection may not be detectable. We extensively analyze this practical limit in Section VII.

We note that there are a number of methods of phone call or other communications interception that do not rely on redirection attacks. This includes the legitimate or illegitimate uses of lawful intercept interfaces — technologies that facilitate "wiretaps" by law enforcement [17]. This also includes attacks on cellular phone networks, including so-called IMSI-catchers or Stingrays [18], [19], as well as actual compromised core network devices [20]. All of these are explicitly more difficult to achieve than SS7 attacks, and some require physical access. We note that while there is currently no effective countermeasure to protect against SS7 attacks, many of these other attacks have countermeasures that would be complementary to Sonar, including work to assure accountability in lawful intercept devices [17] and to detect eavesdropping or tampering [18], [19].

V. Sonar Protocol

We now describe the operation of Sonar. We begin by highlighting the ways in which Sonar differs from prior work in distance bounding. We then proceed to describe our protocol, which is based on the RFID Distance Bounding Protocol by Hancke and Kuhn [4], but adapted to our setting. After describing the protocol, we discuss how this protocol provides secure measurement of RTT. We continue by discussing how we can transmit data through the audio of the phone call, and we conclude by describing how to establish an end-to-end shared secret for use in the protocol.

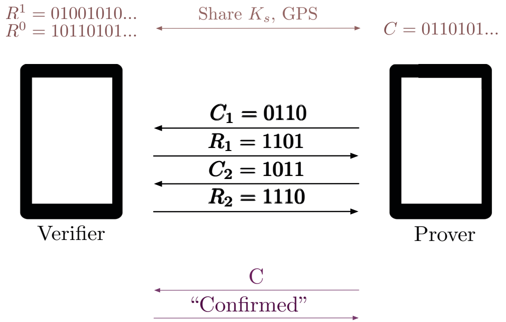

Fig. 3: Sonar uses a distance-bounding inspired mechanism to measure call RTT.

A. Sonar vs Traditional Distance Bounding

Distance bounding is a well studied technique for limiting attack distance. It is most prominently featured in situations where distance between two parties can be known a priori; two frequent examples are line-of-sight wireless communications [4] and smartcard-based payment terminals [16]. Analogous to traditional distance bounding, Sonar requires both parties on the call to participate in the protocol. However, Sonar bounds distances in a unique context against a *fundamentally different* adversary than "traditional" distance bounding techniques. These differences are illustrated in Figure 4. In traditional distance bounding, the distances to be measured are typically limited by physical constraints, for example, the communications delay between a smart card and reader, or the distance to propagate a short-range wireless transmission. As a result, distance bounding typically provides centimeter-level resolution of distance. By contrast, Sonar needs to detect attacks given a known but *highly variable* physical distance. Most distance bounding also assumes a direct connection in legitimate cases. The distance the call travels during routing may also vary significantly; as a result, calls that vary in physical distance by hundreds of miles may experience comparable audio latency. Also, as shown in Figure 4, in traditional distance bounding the prover may actually be the adversary, an attack known as "mafia fraud" [21]. In Sonar, the adversary is a compromised *network* and both parties are trusted. Finally, in Sonar, both parties need confidence that the call has not been intercepted, while in traditional distance bounding, only the verifier needs to know the distance.

B. Protocol Definition

Sonar, like the Hancke-Kuhn protocol, has three phases: "initialization," "rapid audio exchange(RAE)," and "reconciliation." This protocol is visualized in Figure 3.

We follow the common practice in distance bounding of referring to the participants as prover P and verifier V. In traditional distance bounding, the onus is on the prover to participate honestly to prove their distance to the verifier. Note that in Sonar the prover and verifier work collaboratively to accurately measure the call RTT. It is important to have a convention of which party is prover or verifier to prevent an ambiguity that could be exploited by the adversary, so we arbitrarily assign the role of "prover" to the caller.

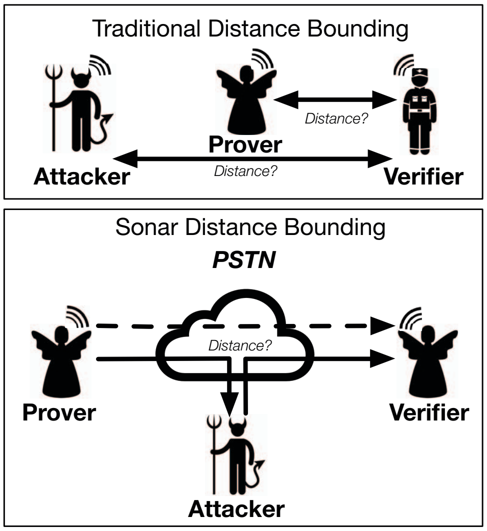

Fig. 4: Sonar vs traditional distance bounding. While in traditional bounding the verifier does not trust the prover, both endpoints are trusted in Sonar.

The initialization phase exchanges information needed for other parts of the protocol, especially for the RAE. The RAE is a challenge-response step that exchanges data between the prover and verifier. Finally, in the reconciliation phase both parties confirm that they have measured a reasonable RTT and commence to complete the call.

Initialization Phase In the initialization phase, P and V first establish a session key, k_s, through in band signaling. This key can be established in a multitude of ways, including Authloop [22] and Authenticall [23]. Second, V generates a random challenge bitstream, C, and both V and P generate two pseudorandom bitstreams, R^0 and R^1, based on the session key. Both P and V have to ensure that the generated keys and data are not used across multiple sessions. Finally, unique to Sonar, P and V exchange GPS location information [4]. This allows both parties to verify that the measured RTT is consistent with the call distance, which is depicted in Figure 4.

Rapid Audio Exchange This phase is marked by P and V securely exchanging data in order to measure the RTTs. In Hancke-Kuhn distance bounding, V sends P a series of individual bits from C sequentially. For each challenge bit, P responds as fast as possible with the next bit from R^0 if the bit is 0, and R^1 otherwise. V accepts each response if it is correct and arrives within some specified time. Because each of $P's$ responses depend on the challenge sent by V, P cannot send responses earlier to create an impression of a lower RTT. This design also allows V to validate responses as they are received. This is termed as "rapid bit exchange."

In contrast to the Hanke-Kuhn protocol, during execution of the Sonar protocol, V sends a batch of l challenge bits as audio from C instead of a single bit. P responds as soon as possible with l response bits, where each bit is drawn from R^0 or R^1 corresponding to the respective challenge bit. Because we send many challenge bits at once, we term this a "rapid audio exchange" (RAE) in recognition of the fact that we are sending bursts of information in audio – not single bits. Total protocol execution time is primarily caused by relatively long RTT, so by batching bits we can maintain high security while limiting execution time.

Reconciliation Phase After the RAE completes, the reconciliation phase establishes whether the data was exchanged without being tampered and decides whether the RTT is consistent with the distance. This can be determined without prior knowledge of a ground truth value, however, possessing this information could improve detection rates. As described in later sections, we do use ground truth RTT values in our experiments. All messages in this phase are sent over a secure channel. V and P first evaluate their received responses to determine that they were correct and unmodified by the adversary. V confirms that the sequence of response bits are correct. If correct, V then sends the transmitted challenge bits C to P so that they can verify that they received the correct challenge and not challenges fabricated by an adversary. V also sends the measured RTT to P, along with the verifier's decision as to whether the RTT is consistent with their measurement. After V confirms C, they respond with an acknowledgment that C was correct and they confirm a desire to continue with the call. If at any point in the reconciliation a check fails, the party with the failing check messages the other and disconnects the call.

Attacks Prevented Drimer and Murdoch [16] identified attacks against challenge-response distance bounding systems, including the adversary guessing responses, re-playing previously used challenges, and using more capable hardware than the prover or verifier. Because all RAE messages are pseudorandomly generated based on the output of a key unknown to the adversary, the adversary can successfully guess challenge or response bits with probability 2^{-n}, where n is the total number of bits.[5] This prevents an adversary from preemptively sending bits to provide a smaller RTT than would otherwise be measured. Similarly, because session keys are guaranteed to be unique for every call, replay attacks are not possible. An adversary's ability to create an advantage with faster hardware can be limited by ensuring that the prover and verifier processing time is much less than the RTT. Finally, because an attacker cannot predict a challenge or response and must be located on the redirected call, an adversary cannot cause a verifier to receive a response to a challenge faster than the RTT between the prover and verifier.

C. Additional Considerations

Data Transmission Sonar can transmit audio using any one of several techniques. The simplest technique involves sending DTMF tones (commonly known as "touch tones" used to dial

[4] Though we use GPS location information in our proposed protocol, it is not a requirement for Sonar. Other alternative methods of providing location information may be used, which is necessary for devices that do not possess a GPS receiver.

[5] This attack success probability is lower than in Hancke-Kuhn because the adversary cannot spoof challenge bits without being detected.

digits) to represent 1 of 16 possible values.[6] These tones have the advantage that they are simple to implement and work with all phone systems, but have the disadvantage that only a few tones per second (tens of bits per second) can be sent. Voice modem technologies (as used for dial-up internet access) could also be used, though standard voice modems cannot transmit data over cellular or VoIP calls due to voice-specific data compression techniques. Sonar could instead leverage new techniques for secure data transmission over phone calls [22] to transmit information at roughly 500 bits per second. Note that we assume a low probability of bit error in this protocol; this can be assured by using redundant data coding along with cryptographic integrity checks of messages after transmission. Finally, for non-RAE portions of the protocol we note that an out of band communications channel (such as a secure data connection) can also be used.

Secret Management The security of rapid bit exchange relies on a shared secret being known to the distance bounding participants but not the adversary. We assume that our protocol participants have already established a long-term symmetric secret, or have a means of establishing a secret in an authenticated way in the presence of an adversary. Recent work in call authentication [22], [23] has provided methods that Sonar can use to establish these secrets. Both techniques rely on public keys assigned to phone users to conduct secure communications. Both techniques also derive a shared secret that can be used by Sonar.

Changes in Carrier Due to variations in cellular providers, telephony devices may experience variation in call RTT due to differences in infrastructure provided by each carrier. For our experiments, we use a single cellular provider to eliminate any additional delay. However, we note that the time added by switching between cellular providers is small. Even in this situation, call quality still needs to meet the ITU E-Model standard [15]. In the event of implementation, variations in carrier can be accounted for in the protocol, which would allow the detection rate to remain the same regardless of time introduced by variations in cellular provider.

VI. Experimental Design

Having designed a protocol that uses RTT to verify the legitimacy of a call, we now design a series of experiments to validate our initial hypothesis. First, we need to measure RTTs to verify that call latency is directly correlated with the physical distance between the communicating parties. Then, we will need to establish if the audio in a rerouted call exhibits higher RTTs in comparison to a legitimate call. This can be accomplished by emulating an actual rerouting attack. We then develop an emulated SS7 attack to launch redirection on our own devices. This will allow us to redirect our own calls to an arbitrary physical location. Finally, we execute real SS7 attacks in order to validate the accuracy of our emulation. The real attacks are discussed in a later section.

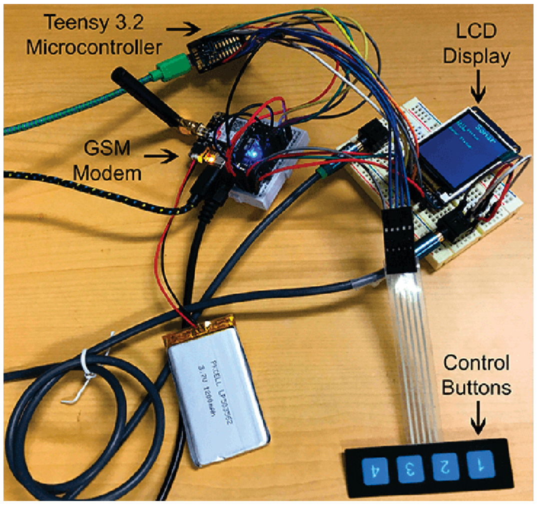

Fig. 5: Pictured are the components that make up the cellular calling device. At the very top of the image is the Teensy 3.2 microcontroller. Directly below that, easily distinguished by its antenna, is the Adafruit FONA GSM module. To the right of that is an LCD display used for call information. The control buttons at the bottom of the image are used to initiate and end calls as well as initiate playing DTMF tones.

A. Network Measurement

We focus first on the problem of measuring the RTT and routing path for an arbitrary call. Since we cannot query or influence the route a call takes through the PSTN, we construct a system that will allow us to select and know the call route in order to collect any meaningful measurement of RTT or distance. We do this for 3 different telecommunication setups: Cellular-to-Cellular, VoIP-to-VoIP, and VoIP-to-Cellular.

Cellular-to-Cellular. First, we need to characterize the RTT for calls on a cellular network. Measuring RTT for cellular devices is challenging, especially with modern smartphones that restrict access to call audio and network features. To overcome this barrier, we designed hardware that grants direct access to the call audio stream. We chose to use the GSM network for the reasoning that GSM modems are easily accessible and well documented.We note that regardless of cellular network or technology we chose to incorporate into our experiments, the results will be analogous due to our system operating in the voice channel. We built two devices: one device places calls and transmits DTMF tones while the other that accepts the call and echoes back any received audio. To collect RTT data with these devices, we keep the calling device at our lab in Gainesville, FL while the other is moved to multiple locations. We then initiate calls between the two devices and measure the RTT. As shown in Figure 5, we use a Teensy 3.2 [24] microcontroller and an Adafruit Fona [25] GSM module.

To obtain precise measurements of the audio stream, we attached a logic analyzer [26] to the audio output of the calling device. The calling device notifies the logic analyzer to start recording immediately after it sends the DTMF tone to the echoing device. We then measured the difference between the

[6] Most touch tone phones have 12 digits, but an additional 4 tones are defined for special functionality.

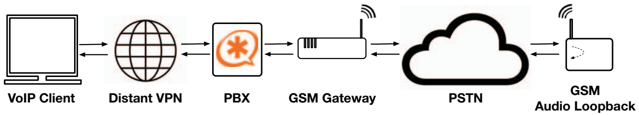

Fig. 6: Our VoIP to Cellular testbed. A VoIP client registers with the PBX server that routes the call to the GoIP. The call then connects to the cellular network and to the echoing device.

start of the recording to the beginning of the echoed DTMF tone to obtain the RTT.

VoIP-to-VoIP. While we cannot control the path a PSTN call takes through the network, we have some control over the path VoIP traffic takes over an IP network. We therefore seek to also understand how latency and physical distance relate to VoIP calls, and if this model is similar to PSTN calls, which we describe in greater detail below. Acquiring RTT data for VoIP-to-VoIP calls was conducted in a manner similar to the Cellular-to-Cellular calls. We used two PCs running the VoIP software PJSUA [27], where one is the caller and the other is an audio loopback receiver. After the call is established, the caller starts recording the call while simultaneously playing an audio sequence into the stream. The RTT for the call audio is obtained by measuring the time between the echoed audio entering the audio stream and being recorded at the receiver. This is representative of the ”worst case scenario” for time introduced by rerouting attacks.

VoIP-to-Cellular. Finally, because of the difficulty of executing actual SS7 rerouting attacks legally and ethically for a variety of locations, and because no emulator exists, we seek to emulate SS7 redirection attacks. In order to emulate a redirection, we use a VPN to force the IP network traffic for a VoIP call to traverse an additional hop in the network. After traversing the VPN, the call returns to a local Asterisk PBX, connected to a Hybertone GoIP-1 GSM gateway[7], which places a cellular call to the cellular echoing device described above. This flow is shown in Figure 6. If RTT and distance are truly correlated, this experiment will influence RTT by augmenting the normal VoIP traffic with an additional network hop, faithfully emulating an SS7 redirection attack.

For this experiment, the VoIP client is identical to the VoIP-to-VoIP experiment. In order for our client to place calls, we set up a private branch exchange (PBX) server that allows the two to communicate. The calls are tunneled through the PBX server to the gateway where they are transfered onto the PSTN. These three devices, outlined in Figure 7, are at a fixed location, giving us the ability to control the call route. Once this infrastructure is in place, we can reroute and measure calls to arbitrary locations.

It may initially seem that VoIP-to-PSTN providers (e.g., Vonage and MagicJack) could provide similar functionality with less complexity. However these systems suffer from a similar problem to the PSTN: they are opaque and call routing is not visible. These systems must route a call to a switch that physically connects to the PSTN, adding additional unknown distance and routing to the call. By fixing the location of our PBX and gateway, we know the exact route the calls must take. In essence, we are constructing our own VoIP-to-PSTN provider. Accordingly, we can make more precise measurements for both distance and RTT.

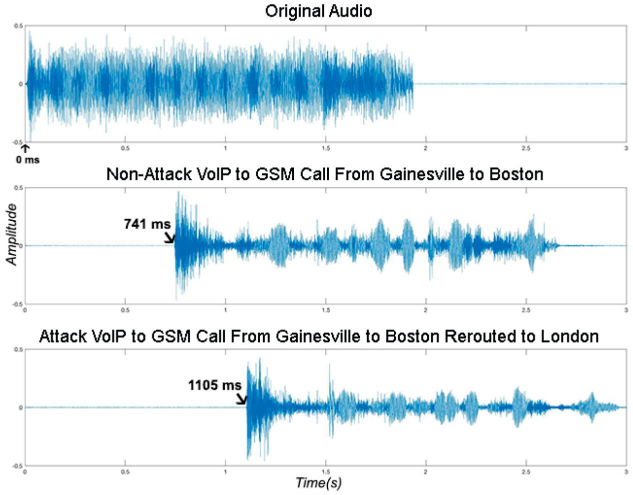

Fig. 7: The top waveform is the time domain representation of the audio used to obtain RTT values for the VoIP to cellular experiments. Beneath that is a recording for a VoIP to cellular call from our lab in Gainesville, FL to Boston. The top audio sample is played as soon as the call is connected and returns after 741ms. The bottom waveform is call audio for a VoIP to cellular call from our lab to Boston where call audio is rerouted to London. It can be seen that the audio returns at 1105ms, much later than the call without rerouting.

B. Data Collection

With collection methods designed we can now describe our extensive network measurements. In order to collect data for a geographically diverse set of locations, we physically shipped or personally transported the cellular echoing device (prover) to multiple locations throughout the continental United States, including most major regions: Southeast (Atlanta, GA; Miami, FL), Northeast (Boston, MA; Stroudsburg, PA), West (Eugene, OR; San Diego, CA; San Jose, CA), South (Houston, TX), and the Midwest (Chicago, IL). Since our device required a human volunteer to physically power on the device and ship it to its next location, we were constrained by the availability of willing volunteers and the quality of cellular phone service at their locations. We also selected locations where our VPN provider maintained endpoints to allow us to perform VoIP-to-VoIP experiments, allowing us to verify that VPN latency corresponds to expected latency during a real SS7 attack.

At each location, we executed two types of calls: legitimate calls and rerouted calls. For legitimate calls, we measured RTT alongside great circle distance[8] while performing VoIP-to-Cellular and Cellular-to-Cellular calls. This allowed us to

[7] A VoIP GSM gateway connects VoIP calls to the PSTN through the cellular network.

[8] Great circle distance is the shortest distance between two points on a sphere. It is called “great circle” because the path forms a long arc when drawn on a flat map.

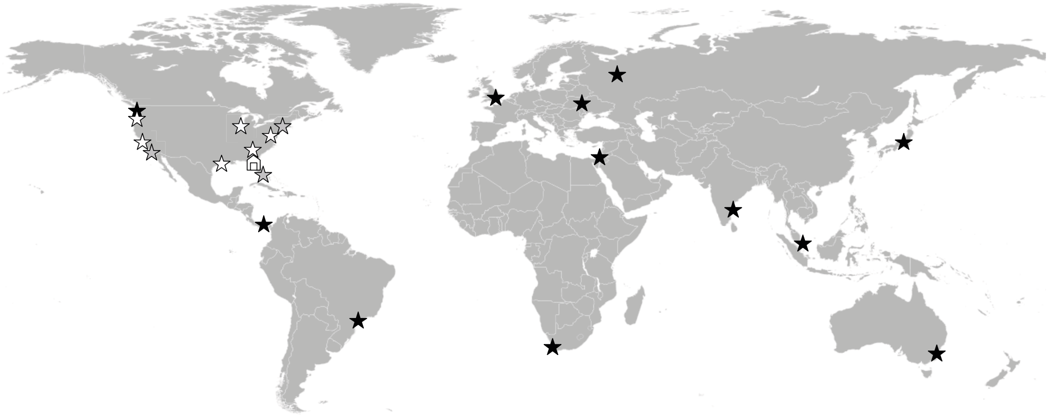

Fig. 8: Black stars indicate locations where a VPN server was used for rerouting call data. White stars indicate locations where the echoing device traveled. Grey stars indicate locations where the echoing device was sent and where a VPN server was used to reroute calls.The house icon indicates the location of our lab in Northern Florida.

both baseline the expected values and test for false positives. Finally, we emulated an attack call by performing a VoIP-to-Cellular call while rerouting the VoIP call through a VPN endpoint, forcing the traffic to traverse an additional route.

We chose four VPN endpoints within the United States, distributed at each corner of the country, to emulate an SS7 attack being executed within the US: Seattle, WA; San Diego, CA; Miami, FL; and Boston, MA. These RTTs are expected to be substantially lower than calls routed internationally due to the shorter distance. Accordingly, these calls should be more difficult to detect. To emulate international SS7 attacks, we used 11 international VPN endpoints across multiple regions: South America (Panama City, Panama), Western Europe (London, UK), Eastern Europe (Moscow, Russia), Middle East (Tel Aviv, Israel), and Asia Pacific (Tokyo, Japan). After training on these cities, we use VPNs in extra cities to validate our model. Those cities consist of Sydney, Singapore, São Paulo, Kiev, Chennai, and Cape Town.

The audio loopback device was carried to 9 other locations, shown in Figure 8. We also performed calls locally at our lab in Gainesville, Florida (located in the northern portion of the peninsula). At each location we executed and measured approximately 240 calls, corresponding to 4 hours of experiments per location. In total, the experiments took nearly 50 hours to run over a period from May 2, 2017 through August 25, 2017. Note that this period does not include early test measurements and device calibrations that began as far back as January 2017.

C. Detection

We next describe the design of an anomaly detection system that allows us to characterize the effectiveness of using RTT to detect calls. Building this detection system poses a number of challenges. First, a lack of an exhaustive dataset of call data for all locations means that our detection model needs the ability to interpolate estimates of reasonable RTT for locations not in the training data. Second, note that *routing* distance can vary significantly from great-circle geographic distance. While RTT intuitively increases with distance, due to variances in physical routing of the call there is a high variance in the actual RTT from call-to-call. This means that attempting to estimate the actual distance from an RTT is quite difficult, and we found that even the most flexible regression models (which we do not use) can misestimate distances by thousands of miles. As a result, two locations that are the same geographic distance can have very different audio latencies. The model must take this into account. Third, our test locations were carefully chosen to provide insight into how varying distances affect RTT, not the most likely or probable attack locations (which will of course vary from victim to victim). While it would be possible to train a binary classifier to distinguish between our collected attack data and our legitimate call data and get good accuracy, this test would be heavily biased by our choice of attack locations. Finally, to characterize the effect of attackers located close to a victim some of our redirection experiments move a call only a short distance. Because we have several close attacks for every legitimate call, some classifier models would be more likely to consider legitimate calls as false positives.

Because we are not trying to detect specific redirection attacks (e.g., Atlanta to Boston via London) but rather *any* redirection, we found the most appropriate model to be an anomaly detector. We developed a classifier using the commonly used One-Class Support Vector Machine classifier (OCSVM) [28]. This classifier is similar to a standard support vector machine, except that rather than identifying a decision boundary between two classes it identifies a boundary that includes all training data but minimizes the area not including training data.

Like traditional SVM classifiers, OCSVM can also use a kernel method to learn non-linear boundaries (among other properties). We use a radial-basis function (RBF) as our kernel because it allows for learning a generalized organically-shaped region with no assumptions about the underlying data distribution. Like most machine learning techniques, this technique requires the selection of hyperparameters that affect the model learned from data. We set these factors extremely conservatively to minimize the possibility of false positives on legitimate calls with extreme latency values relative to

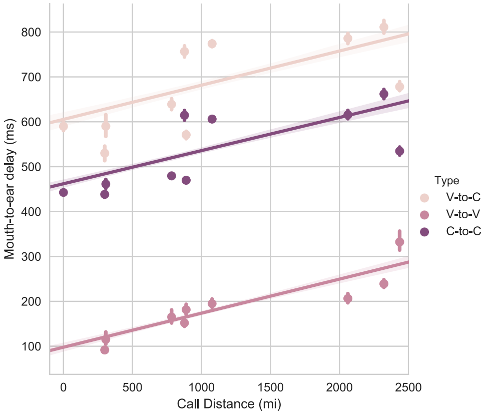

Fig. 9: Measured RTT vs distance for all calls. Regardless of call technology, RTT is strongly correlated with call distance.

their location. OCSVM uses a hyperparameter ν to effectively regularize the boundary learned. ν can be interpreted as both a) the maximum percentage of "outliers" to be ignored in the training data and b) the maximum classification error of in-region data. Because we consider all of our call data points to be legitimate, we set $\nu = 0.01$. This conservatively limits false positives and expands the learned region, making our classifier more likely to accept legitimate calls outside the training data set. RBF kernels use a hyperparameter γ that can be interpreted as the effect that any individual data point has on the learned model. We set $\gamma = 0.05$ so that every point has a significant and far reaching effect on the model. This also gives protection against misclassifying extreme legitimate values.

We note that this model assumes no prior underlying distribution of the data. This is critical because while RTT tends to increase with distance, the rate at which it does so can vary on technology used, network conditions, routing topologies, and variable effects like congestion. Note also that our model is currently only trained and evaluated on VoIP-Cellular calls. This is because VoIP-to-Cellular calls are the only types of calls, not including the actual SS7 attack calls we made, for which we were able to collect both legitimate and redirected data for a variety of locations. We do not include the data we collected for real SS7 rerouting calls in our model which is further explained in a later section.

VII. Experimental Results

We now present the results of our experiments, discuss our analysis methodology, and demonstrate how call redirection can be accurately detected. These results are then used in a later section in determining the accuracy of our simulation with data from real SS7 rerouting attacks.

A. Legitimate Calls

Figure 9 displays the means (with error bars) of non-attack calls for each location broken down by technology (Cellular-Cellular, VoIP-VoIP, and VoIP-Cellular). The figure shows trend lines for each call type and shows visually the correlation between call distance and RTT.

We begin by verifying our hypothesis that RTT is correlated with distance. We calculated the Spearman correlation ρ between distance and measured RTT for each of the three types of calls. Spearman correlations indicate monotonicity of the data being compared and assume no underlying distribution of the data. In our case, higher ρ values (approaching 1) indicate a better fit of the points to the regression, meaning a higher likelihood that RTT and distance are correlated. For VoIP-to-Cellular calls, we calculated ρ at 0.68; Cellular-to-Cellular at 0.75; and VoIP-to-VoIP at 0.78 (all of which fall in the accepted range for strong correlation). All three calculations had a p-value of < 0.001, indicating that these results are also statistically significant. Therefore, we can reject the null hypothesis that these two variables are uncorrelated.

Several trends are evident from this plot. First, different call types experience different levels of RTT – VoIP-Cellular calls experience approximately 150ms more latency than Cellular-Cellular calls. Second, while there is strong correlation between distance and RTT, the relationship is not perfectly linear. This reflects the fact that RTT measures actual call distance through the network, which may diverge from the ideal great circle distance. Finally, we find all RTTs for Cellular-Cellular calls are consistent with "Users Very Satisfied" or "Users Satisfied" mouth-to-ear delay guidelines from the ITU E-Model. Because carriers use the E-Model to provision network services with a high quality of service, our measured values are highly likely to be consistent with the true RTT.

B. Effects of Rerouting on RTT

With the understanding that distance is correlated to RTT, we now seek to emulate an SS7 attack and measure the effects. As we stated in Section VI, conducting SS7 attacks legally and ethically is difficult. Additionally, no emulator exists to allow us to test this part of our hypothesis on the telephony network. We therefore use the VoIP-to-VoIP measurements to baseline our expectations. As Figure 9 shows, the VoIP-to-VoIP legitimate call RTTs increase similarly with distance to Cellular-to-Cellular calls. However, VoIP-to-VoIP calls have substantially lower RTT than any call we placed over the cellular network. While this is expected given that the telephone network must alter call audio, it provides an additional insight to our methodology: using a VPN to redirect calls will cause a *smaller* increase in RTT than we would expect from a redirected call in the phone network, *making our emulated calls more difficult to detect than a real SS7 attack*. We confirm this with real SS7 attacks in section VIII.

Our emulated SS7 attacks consisted of a VoIP-to-Cellular call routed over a geographically-diverse set of VPN endpoints. The results are shown in Figure 10, which shows the RTT vs. the great-circle distance of the call including the redirection. Again we see an increase in RTT as call distance increases. The Spearman ρ value was 0.79 ($p < 0.001$), indicating a strong correlation between RTT and call distance. This confirms that the hypothesis still holds with our redirected calls.

C. Analysis of Detection

While we have confirmed that RTT and distance are correlated and that redirection of VoIP calls results in an expected

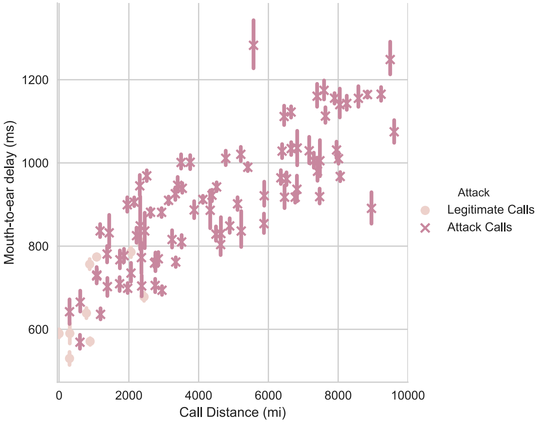

Fig. 10: RTT for VoIP to Cellular calls. Most notably, delay increases with the increase of distance.

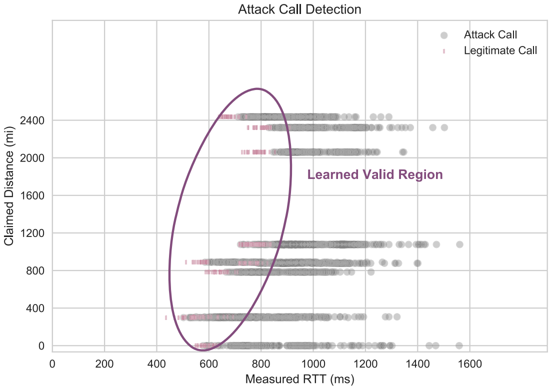

Fig. 11: The classifier generated detection graph. Calls that fall within the area inside the oval are categorized as legitimate, while those outside are deemed as attack. In our testbed the classifier detected attack calls with a false positive rate of 0.3%.

increase in RTT, we now focus on evaluating our detector.

Figure 11 shows the raw plot of RTT versus the *claimed* great-circle distance (which does not include any additional distance gained by redirection) for legitimate and attack calls. Note that in Sonar the verifier knows the distance the call *should* take (the claimed distance) and measures the actual RTT. The oval region indicated on the graph is the decision boundary learned by the classifier; points outside that region are classified as attack calls. For all calls this resulted in a true positive rate of 70.9% and false positive rate of 0.3%. Because we know the claimed destination of the call, however, we can break down these rates by expected destination.

Intuitively, calls redirected to a nearby attacker destination result in lower detection rates and higher false positives. This is due to the increase in call distance being relativity low and therefore causing the RTT to only increase slightly or in some extreme causes stay relatively the same. For example, when calling San Diego, CA from our location, the presence of an attacker in Boston, MA will be easier to detect (81% detection accuracy) than in Miami, FL (48% detection accuracy). This is because the distance between our lab in Gainesville, FL and Boston, MA is greater than that between us and Miami, FL. Furthermore, international calls have the highest detection rates in our data set which is, in part, due to distance. We were able to detect 100% of attacks from Tokyo and Moscow.

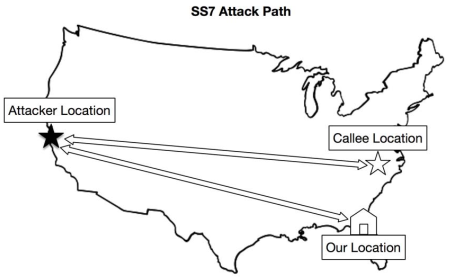

Fig. 12: Our actual SS7 rerouting attack redirects a call from our lab in Gainesville, FL to San Francisco before it reaches its intended destination of Chapel Hill, NC.

Finally, we collected extra data to further validate our classifier against calls attacked by foreign adversaries. Our goal was to validate our model for locations outside of those used in our training set. In addition to the previous attack locations, these new attack points (Sydney, Singapore, Sao Paulo, Kiev, Chennai, and Cape Town) were taken when the audio loopback device was in Houston and Miami. The classifier that was trained with the original larger data set was used with these additional RTTs to see whether the classifier would correctly predict all of these calls as attacked. The classifier was able to predict the attack calls with 100% accuracy (0% false positive).

Overall, our results confirm our hypothesis that redirected calls can be detected using round trip time measurements.

VIII. Real Attack Measurements

To observe how real SS7 attacks impact the RTT of call audio, we partnered with Vaulto, a telecommunications company that has the capability to execute actual SS7 call rerouting. To the best of our knowledge, this is the first work to collect measurements on real SS7 rerouting attacks. We only targeted our own research devices in these experiments. When called with the echoing device we used in previous experiments the call is rerouted to San Francisco via SS7 before reaching its intended destination. Note that due to its peering agreement with another provider, our partner was *only* allowed to redirect calls to their office in the Bay Area. Accordingly, we were not allowed to recreate all of our emulation experiments. The route the call travels is presented in Figure 12.

We collected RTT data for 50 calls, 25 of which experienced rerouting. To determine if the two groups are statistically different, we performed a power analysis on the data. This analysis tells us whether or not there exists a statistically significant difference between regular and attack calls. With a $p < 0.001$, we calculated a power of 1.0 and a sample size of 10 per class. Accordingly, we were easily able to reject the

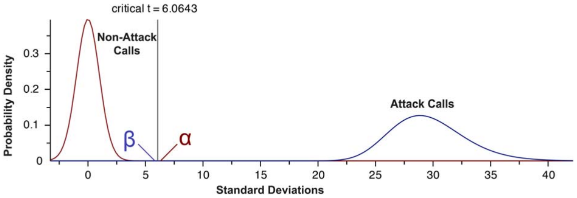

Fig. 13: Distribution curves for normal and real SS7 redirection attack impacted calls. The curve is a plot of probability density versus standard deviations. The critical-t value is the point at which the null hypothesis can be rejected. Our statistical tests not only demonstrate that our sample is more than large enough, but also that normal and attack traffic exhibit RTTs that differ in a statistically significant way.

null hypothesis. Figure 13 shows the clear distinction between normal and attack traffic.

The results of real SS7 rerouting attacks differ from our attack emulations. On average, an actual SS7 rerouting attack to San Francisco added an additional second to the average legitimate call RTT. This is higher than the average time introduced by rerouting to Moscow, Tokyo, or Tel Aviv in our testbed. The reason that the time introduced by actual SS7 rerouting is significantly higher than our testbed is due to additional in-network processing that calls must go through. Actual SS7 rerouting attacks require the call audio to traverse additional hardware such as service switching points, signal transfer points, and service control points. These add latency to the call before it researches its true destination. Geographic and technological concerns prevent us from determining the exact time introduced by additional SS7 hardware and audio stacks at the network level.

This is the first research paper to observe the effects of SS7 rerouting attacks and propose a possible end-to-end solution. *Our emulation is a conservative estimate of actual SS7 rerouting attacks in terms of RTT and because of such, evaluates our system on a more challenging criteria than what would be seen with actual SS7 rerouting attacks.*

IX. Discussion

A. Limitations

Under certain conditions, detection may have increased difficulty. If an adversary hijacks an SS7 node that is close to the path a legitimate call would take, Sonar may have difficulty detecting the attack. The RTT increase introduced by the additional distance the call audio travels will be minimal in this scenario, however, the time added by the additional audio stacks and processing at the SS7 node will still remain the same. In this case, we believe at the worst, Sonar would perform similarly to the way it did in our emulations. Additionally, as mentioned in section V, Sonar requires both parties to actively participate in the protocol for rerouting detection to occur. This is analogous to traditional distance bounding [4].

B. Varying Network Conditions

Variations in RTTs occur regularly in networks. These differences are generally the result of network traffic conditions and variations in routing paths. Extensive work has been done in the network measurement communities to characterize such variations in both traditional and cellular data networks. However, to our knowledge, a similar public longitudinal study across telephony voice networks has not been conducted.

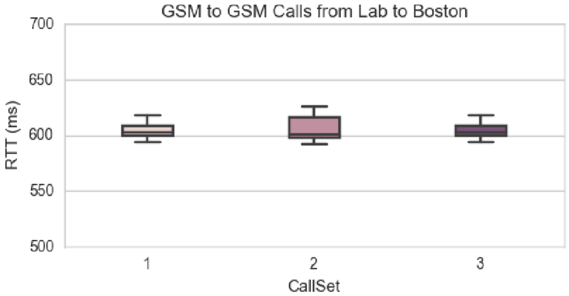

Fig. 14: Variation in recorded audio RTTs between our lab and Boston, MA evenly spaced across a two day period. These experiments, taken from a Sunday night to a Monday evening, show only little variation between tests at different times.

We conducted a very small scale study to determine whether measurements in this space are subject to wide variation. We are not arguing that this is a comprehensive study, but rather that our results are a reasonable approximation of normal network conditions in systems with circuit switched or circuit-switched-like behavior. Figure 14 is a box plot representing three sets of audio latency tests collected over a two day span. These tests were conducted between Cellular devices in our lab and Boston, MA, and represent collection during a Sunday evening, Monday morning and Monday evening. This small study confirms that very little variation was seen during our testing. We intend to perform such a study in future work.

C. Localizing Callers

Our results showed that RTT was sufficient to determine that a call is likely rerouted, but that RTT was insufficient to actually determine the location of the other party. While our current threat model assumes mutually trusted parties, phone calls are often used among parties that may not initially trust each other [29], and having accurate location measurement would allow us to relax Sonar's trust assumptions.[9]

Determining location is hindered by a number of factors, one of which is that RTT is a scalar value that can only resolve distance, not position. As a result, accurate and reliable RTT measurements from multiple positions on the globe would be necessary to triangulate caller position. Such triangulation is not possible with a single call. One possibility would be for a group of mutually trusting collaborators to sequentially call a proving party to provide triangulation measurements. However, because an adversary with the ability to redirect calls can simply choose to not delay some of the triangulation calls, the adversary has a significant amount of power to influence the

[9] Although we note that Sonar assumes both call parties trust each other, the Sonar protocol is actually designed such that a prover cannot reduce RTT because responses depend on a challenge only known to a prover. Note also that an adversarial prover could still add delay, just not reduce it.

measurement. A second issue is that distance bounding only guarantees a *lower* bound, not an upper bound; a dishonest prover could add delay to change their triangulation.

An appealing option might be to leverage conference calls so that all triangulating parties are on the same call concurrently. Unfortunately, in most telephone systems conference calls are implemented using a hub-and-spoke model, where either all calls are placed to a central bridge or a single call party acts as the meeting point for all other callers. This means that a group could conceivably bound the distance to the conference bridge, but only if that bridge responded honestly to all verifiers.

It may be possible to use other features besides RTT to localize a caller. For example, it may be possible to use systems that learn noise characteristics of phone calls in various locations [30] to locate a caller with reasonable accuracy. However, such techniques would require exhaustive global characterization, making them impractical at scale. While delay is unlikely to be useful for localization, future work may lead to other techniques for call localization.

X. Related Work

SS7 has been known to be vulnerable to compromise for some time [31], [32]. To date, attack research has detailed how mobile users may be tracked [6], service may be denied [33], and how attackers can redirect calls and text messages [33], [7]. Current defenses against SS7 attacks focus mainly on network layer filtering [7], [34], [35] and several research groups have measured the extent of this filtering from a network perspective [36], [34]. However, this filtering has a number of problems [34]: it does not address every attack, not all network equipment supports filtering, when filtering is implemented it increases load on an already strained network, and customers of networks that filter are not protected when they roam on non-filtering networks. The SnoopSnitch app [37] offers a mechanism to detect messages sent to a mobile device indicative of SS7 tracking. However, it relies on capabilities provided by debug interfaces only available to a select set of mobile devices, and this detection mechanism is simply not available for most devices. It is important to note that the anticipated successor for SS7, Diameter, is also vulnerable to redirection attacks [35]. Sonar is the first system to protect any type of phone against redirection attacks in any telephone network – especially those that do not create messages during interception. Sonar is also the first system to be tested against real SS7 rerouting attack data.

SS7 is not the only mechanism that can redirect calls — the Border Gateway Protocol can be exploited to redirect Internet traffic (including VoIP calls). Similar to SS7, BGP was developed and deployed without mechanisms for authentication, authorization, or accountability [13]. BGP is also vulnerable to malicious entities fabricating or modifying messages. These include announcing fraudulent ownership of or short routes to targeted IP address blocks. These attacks redirect flows from the victim IP addresses to the network controlled by the attacker. It is important to note that while end-to-end encryption is possible with VoIP calls [38], certain configurations are still vulnerable to content recovery [39]. Because Sonar is agnostic to the underlying network topology, it can also detect redirected VoIP calls.

Sonar builds on prior work in distance bounding protocols. Though there are many such protocols[40], [41], [21], [42], [43], [44], none were directly appropriate for our application. Specifically, such protocols have been used primarily in short ranged applications[41], [45], [46], [47], especially with ultra-wide band (UWB) frequencies[41]. Because of the large distance range of our application and the variation in signal path in each phone call, our distance bounding method needed to accommodate these constraints. Our work is also reminiscent of research on Internet tomography [48], [49], [50]; however, the point-to-point nature of phone calls makes such techniques inappropriate for addressing this particular attack.

Finally, telephony security has received much attention over the past few years. This includes taxonomizing [51], [52] and detecting network-based fraud such as Interconnect Bypass [53] and OTT bypass [54]. Other work has focused on problems affecting the end-user, including robocalls [55], abusive calls [56], [57], and how such calls can support other types of crime [58], [59]. Efforts to address these problems has focused either on both detection [60], [61], [62], [30] and mechanisms to authenticate caller identity [38], [63], [22], [64]. Additionally, significant efforts have identified flaws particular to cellular networks [65], [66], [67], [19], [18] including problems in GSM [68], [69], UMTS [70], [71], [72], [73], [74], and LTE [75], [76], [77], [78], [79], [80] standards and the encryption algorithms they use [81], [82].

XI. Conclusion

SS7 redirection attacks threaten the confidentiality of all calls passing through the PSTN. While the ability to redirect calls is a legitimate feature (e.g., call forwarding), the number of parties with the ability to perform such an action without explicit authorization is now large. We design and implement Sonar as a means of detecting such attacks, which characteristically increase the latency of call audio what they are executed. As such, Sonar develops the Rapid Audio Exchange protocol and a model for correlating distance and RTT. Using calls in real networks and our own redirection testbed, we demonstrate the ability to detect such redirections with high accuracy - as much as 97.0% (0.03% FP) when they leave North America. We also executed real SS7 attacks and were able to detect them all. We believe that Sonar can easily be included in future handset hardware, and represents the best means of reliably detecting such attacks in the short and medium terms.

Acknowledgments

This work was supported in part by the US National Science Foundation under grant numbers CNS-1617474, CNS-1464088 and CNS-1540217. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. We would also like to thank Assaf Klinger and Vaulto (www.vaulto.co), Amir Houmansadr, and our anonymous reviewers for their assistance.

References

[1] The 104th Congress of the United States, "The Telecommunications Act of 1996," http://www.fcc.gov/Reports/tcom1996.txt, 1996, Pub. LA. No. 104-104.

[2] S. Alfonsi, "Hacking your phone," *CBS News: 60 Minutes*, April 2016.

[3] D. Goodin, "Thieves Drain 2FA-Protected Bank Accounts by Abusing SS7 Routing Protocol," Ars Technica - https://arstechnica.com/security/2017/05/thieves-drain-2fa-protected-bank-accounts-by-abusing-ss7-routing-protocol/, 2017.

[4] G. P. Hancke and M. G. Kuhn, "An RFID Distance Bounding Protocol," in *First International Conference on Security and Privacy for Emerging Areas in Communications Networks (SECURECOMM'05)*, Sep. 2005, pp. 67–73.

[5] S. Gallagher, "Congressman to FCC: Fix phone network flaw that allows eavesdropping," Ars Technica - https://arstechnica.com/security/2016/08/congressman-to-fcc-fix-phone-network-flaw-that-allows-eavesdropping/, 2016.

[6] Tobias Engel, "Tracking Mobile Phones," http://berlin.ccc.de/~tobias/25c3-locating-mobile-phones.pdf, Berlin, 2008.

[7] Karsten Nohl, "SS7 Attack Update and Phone Phreaking," https://www.youtube.com/watch?v=BbPLscWQ1Bw, 2016.

[8] J. Cox, "Senator demands answers from telecom giants on phone spying," https://www.thedailybeast.com/senator-demands-answers-from-telecom-giants-on-phone-spying, sep 2017.

[9] T. Fox-Brewster, "All that's needed to hack gmail and rob bitcoin: A name and a phone number," https://www.forbes.com/sites/thomasbrewster/2017/09/18/ss7-google-coinbase-bitcoin-hack/#484e841941a4, sep 2017.

[10] C. Brendan, "8 SS7 vulnerabilities you need to know about," http://www.cellusys.com/2015/10/20/8-ss7-vulnerabilities-you-need-to-know-about/, Oct. 2015.

[11] "3GPP Specification TS 29.078: Customized Applications for Mobile network Enhanced Logic (CAMEL) Phase X; CAMEL Application Part (CAP) specification," 2015.

[12] A. D. Keromytis, "A Comprehensive Survey of Voice over IP Security Research," *IEEE Communications Surveys Tutorials*, vol. 14, no. 2, pp. 514–537, 2012.

[13] Butler, K., Farley, T.R., McDaniel, P., and Rexford, J., "A Survey of BGP Security Issues and Solutions," vol. 98, no. 1, Jan. 2010.

[14] "OpenSS7," http://www.openss7.org/, 2017.

[15] Telecommunication Standardization Sector of ITU, "One-way transmission time," International Telecommunications Union (ITU), Tech. Rep. G.114, May 2003.

[16] S. Drimer and S. J. Murdoch, "Keep Your Enemies Close: Distance Bounding Against Smartcard Relay Attacks," in *USENIX Security*, vol. 2007, 2007, pp. 87–102.

[17] A. Bates, K. R. Butler, M. Sherr, C. Shields, P. Traynor, and D. Wallach, "Accountable wiretapping -or- i know they can hear you now," *Journal of Computer Security*, vol. 23, no. 2, pp. 167–195, Jun. 2015.

[18] Dabrowski, Adrian, Pianta, Nicola, Klepp, Thomas, Mulazzani, Martin, and Weippl, Edgar, "IMSI-catch Me if You Can: IMSI-catcher-catchers," in *Proceedings of the 30th Annual Computer Security Applications Conference*, ser. ACSAC '14, 2014, pp. 246–255.

[19] Dabrowski, Adrian, Petzl, Georg, and Weippl, Edgar R., "The Messenger Shoots Back: Network Operator Based IMSI Catcher Detection," in *Research in Attacks, Intrusions, and Defenses*, Sep. 2016, pp. 279–302.

[20] Vassilis Prevelakis and Diomidis Spinellis, "The Athens Affair," *IEEE Spectrum*, Jun. 2007.

[21] C. Cremers, K.B. Rasmussen, B. Schmidt , and S. Capkun, "Distance hijacking attacks on distance bounding protocols," Proceedings of the IEEE Symposium on Research in Security and Privacy, 2012.

[22] Bradley Reaves, Logan Blue, and Patrick Traynor, "AuthLoop: Practical End-to-End Cryptographic Authentication for Telephony over Voice Channels," in *Proceedings of 25th USENIX Security Symposium*, Austin TX, Aug. 2016.

[23] Bradley Reaves, Logan Blue, Hadi Abdullah, Luis Vargas, Patrick Traynor, and Tom Shrimpton, "AuthentiCall: Efficient Identity and Content Authentication for Phone Calls," in *Proceedings of the 26th USENIX Security Symposium*, 2017.

[24] "Pjrc: Electronic projects," https://www.pjrc.com/teensy/, 2017.

[25] "Adafruit," https://www.adafruit.com/, 2017.

[26] "Saleae," https://www.saleae.com/, 2017.

[27] "Pjsip," http://www.pjsip.org, 2017.

[28] B. Schölkopf, J. C. Platt, J. C. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the Support of a High-Dimensional Distribution," *Neural Computation*, vol. 13, no. 7, pp. 1443–1471, Jul. 2001.

[29] Susan E. McGregor, Polina Charters, Tobin , and Franziska Roesner, "Investigating the computer security practices and needs of journalists," in *Proceedings of the 24th USENIX Conference on Security Symposium*. Berkeley, CA, USA: USENIX Association, 2015, pp. 399–414.

[30] V. A. Balasubramaniyan, A. Poonawalla, M. Ahamad, M. T. Hunter, and P. Traynor, "PinDr0p: using single-ended audio features to determine call provenance," in *Proceedings of the 17th ACM conference on Computer and communications security*, ser. CCS '10. New York, NY, USA: ACM, 2010, pp. 109–120.

[31] T. Moore, T.Kosloff, J. Keller, G. Manes, and S. Shenoi, "Signaling system 7 (SS7) network security," 45th Midwest Symposium on Circuits and Systems, 2002.

[32] "SMS SS7 Fraud," http://www.gsma.com/newsroom/wp-content/uploads/2012/12/IR7031.pdf, GSMA, Tech. Rep. IR.70, 2005.

[33] Luca Melette, "Effective SS7 protection," 2016.

[34] Dominique Lazanski, "Interconnect Security," Jun. 2016.

[35] "Legacy Systems Risk Reductions Final Report," https://regmedia.co.uk/2017/03/29/ss7.pdf, The Communications Security, Reliability and Interoperability Council V, Tech. Rep., Mar. 2017.

[36] Dmitry Kurbatov, "Statistics of Vulnerabilities in SS7 Networks," Jun. 2016.

[37] Security Research Labs, "SnoopSnitch," https://play.google.com/store/apps/details?id=de.srlabs.snoopsnitch, Dec. 2015.

[38] J. Callas, A. Johnston, and P. Zimmermann, "RFC 6189 - ZRTP: Media Path Key Agreement for Unicast Secure RTP," https://tools.ietf.org/html/rfc6189, Apr. 2011.

[39] Andrew M. White, Austin R. Matthews, Kevin Z. Snow, and Fabian Monrose, "Phonotactic Reconstruction of Encrypted VoIP Conversations: Hookt on Fon-iks," in *Proceedings of the 2011 IEEE Symposium on Security and Privacy*, ser. S&P '11. Washington, DC, USA: IEEE Computer Society, 2011, pp. 3–18.

[40] C. Meadows, P. Syverson, and L. Chang, "Towards more efficient distance bounding protocols for use in sensor networks," Proceedings of the Conference on Security and Privacy for Emerging Areas in Communication Networks, 2013.

[41] N.O. Tippenhauer, H. Luecken, M. Kuhn, and S. Capkun, "UWB rapid-bit-exchange system for distance bounding," Proceedings of the 8th ACM Conference on Security & Privacy in Wireless and Mobile Networks, 2015.

[42] N.O. Tippenhauer, C. Pöpper, K.B. Rasmussen, and S. Capkun, "On the Requirements for Successful GPS Spoofing Attacks," Proceedings of the ACM Conference on Computer and Communication Security (CCS), 2011.

[43] J. Clulow, G.P. Hancke, M.G. Kuhn, and T. Moore, "So near and yet so far: Distance-bounding attacks in wireless networks," Proceedings of European Conference on Security and Priacy in ad-hoc and sensor networks (ESAS), 2006.

[44] S. Vaudenay, "Privacy failure in the public-key distance-bounding protocols," *IET: Information Security*, 2016.

[45] D. Singelee and B. Preneel, "Location verification using secure distance bounding protocols," Proceedings of IEEE Conference on Mobile Adhoc and Sensor Systems Confrence (MASS), 2005.

[46] M. Poturalski, "Secure neighbor discovery and ranging in wireless networks," Ph.D. dissertation, EPFL, Lausanne, 2011.

[47] A. Compagno, M. Conti, A.A. D'Amico, G. Dini, P. Perazzo, and L. Taponecco, "Modeling enlargement attacks against UWB distance bounding protocols," vol. 11, no. 7, July 2016.

[48] M. Coates, A. O. H. III, R. Nowak, and B. Yu, "Internet Tomography," *IEEE Signal Processing Magazine*, vol. 19, no. 3, pp. 47 – 65, May 2002.

[49] M. G. Rabbat, M. J. Coates, and R. D. Nowak, "Multiple-Source Internet Tomography," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 12, pp. 2221 – 2234, 2006.

[50] P. Gill, Y. Ganjali, B. Wong, and D. Lie, "Dude, Where's that IP? Circumventing Measurement-based IP Geolocation," in *Proceedings of the USENIX Security Symposium (SECURITY)*, 2010.

[51] Merve Sahin, Aurélien Francillon, Payas Gupta, and Mustaque Ahamad, "SoK: Fraud in Telephony Networks," in *Proceedings of the IEEE European Symposium on Security and Privacy*, Apr. 2017.

[52] G. Macia-Fernandez, P. Garcia-Teodoro, and J. Diaz-Verdejo, "Fraud in Roaming Scenarios: An Overview," *IEEE Wireless Communications*, vol. 16, no. 6, pp. 88–94, 2009.

[53] Bradley Reaves, Ethan Shernan, Adam Bates, Henry Carter, and Patrick Traynor, "Boxed Out: Blocking Cellular Interconnect Bypass Fraud at the Network Edge," in *Proceedings of 24th USENIX Security Symposium*, Aug. 2015.

[54] Sahin, Merve and Francillon, Aurélien, "Over-The-Top Bypass: Study of a Recent Telephony Fraud," in *CCS 2016, 23rd ACM conference on Computer and communications security, October 24-28, 2016, Vienna, Austria*. ACM Press, 2016, pp. 1106–1117.

[55] H. Tu, A. Doupé, Z. Zhao, and G. J. Ahn, "SoK: Everyone Hates Robocalls: A Survey of Techniques Against Telephone Spam," in *2016 IEEE Symposium on Security and Privacy (SP)*, May 2016.

[56] P. Gupta, B. Srinivasan, V. Balasubramaniyan, and M. Ahamad, "Phoneypot: Data-driven understanding of telephony threats," in *22nd Annual Network and Distributed System Security Symposium, NDSS 2015, San Diego, California, USA, February 8-11, 2015*, 2015.

[57] M. Balduzzi, P. Gupta, L. Gu, D. Gao, and M. Ahamad, "MobiPot: Understanding Mobile Telephony Threats with Honeycards," in *Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security*, New York, NY, USA, 2016, pp. 723–734.

[58] A. Costin, J. Isacenkova, M. Balduzzi, A. Francillon, and D. Balzarotti, "The Role of Phone Numbers in Understanding Cyber-crime Schemes," in *2013 Eleventh Annual Conference on Privacy, Security and Trust*, Jul. 2013, pp. 213–220.

[59] N. Miramirkhani, O. Starov, and N. Nikiforakis, "Dial One for Scam: A Large-Scale Analysis of Technical Support Scams," in *Proceedings of the 24th Network and Distributed System Security Symposium (NDSS)*, 2017.

[60] H. K. Bokharaei, A. Sahraei, Y. Ganjali, R. Keralapura, and A. Nucci, "You can SPIT, but you can't hide: Spammer identification in telephony networks," in *2011 Proceedings IEEE INFOCOM*, Apr. 2011, pp. 41–45.

[61] N. Jiang, Y. Jin, A. Skudlark, W.-L. Hsu, G. Jacobson, S. Prakasam, and Z.-L. Zhang, "Isolating and analyzing fraud activities in a large cellular network via voice call graph analysis," in *Proceedings of the 10th international conference on Mobile Systems, Applications, and Services*. ACM, 2012, pp. 253–266.

[62] S. Rosset, U. Murad, E. Neumann, Y. Idan, and G. Pinkas, "Discovery of fraud rules for telecommunications-challenges and solutions," in *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM Press, 1999, pp. 409–413.

[63] H. Mustafa, W. Xu, A. Sadeghi, and S. Schulz, "You Can Call but You Can't Hide: Detecting Caller ID Spoofing Attacks," in *2014 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, Jun. 2014, pp. 168–179.

[64] H. Tu, A. Doupé, Z. Zhao, and G.-J. Ahn, "Toward authenticated caller ID transmission: The need for a standardized authentication scheme in Q. 731.3 calling line identification presentation," in *ITU Kaleidoscope: ICTs for a Sustainable World (ITU WT), 2016*. IEEE, 2016, pp. 1–8.

[65] P. Traynor, P. McDaniel, and T. La Porta, "On Attack Causality in Internet-connected Cellular Networks," in *Proceedings of 16th USENIX Security Symposium on USENIX Security Symposium*. Berkeley, CA, USA: USENIX Association, 2007.

[66] P. Traynor, W. Enck, P. McDaniel, and T. L. Porta, "Exploiting Open Functionality in SMS-Capable Cellular Networks," *Journal of Computer Security (JCS)*, vol. 16, no. 6, pp. 713–742, 2008.

[67] Traynor, Patrick, Enck, William, McDaniel, Patrick, and Porta, Thomas La, "Mitigating attacks on open functionality in SMS-capable cellular networks," vol. 17, no. 1, 2009, pp. 40–53.

[68] Singh, J. and R. and Lindskog, D., "GSM OTA SIM Cloning Attack and Cloning Resistance in EAP-SIM and USIM," in *2013 International Conference on Social Computing (SocialCom)*, Sep. 2013, pp. 1005–1010.

[69] Denis Foo Kune, John Koelndorfer, Nicholas Hopper, and Yongdae Kim, "Location leaks over the GSM air interface," in *Proceedings of the 2012 Network and Distributed Systems Symposium*, Feb. 2012.

[70] U. Meyer and S. Wetzel, "A Man-in-the-middle Attack on UMTS," in *Proceedings of the 3rd ACM Workshop on Wireless Security*, ser. WiSec '04. New York, NY, USA: ACM, 2004, pp. 90–97.

[71] Zahra Ahmadian, Somayeh Salimi, and Ahmad Salahi, "New attacks on UMTS network access," in *Proceedings of the 2009 Conference on Wireless Telecommunications Symposium*, ser. WTS'09, Piscataway, NJ, USA, 2009, pp. 291–296.

[72] S. G. Georgios Kambourakis, Constantinos Kolias and J. H. Park, "DoS Attacks Exploiting Signaling in UMTS and IMS," *Computer Communications*, vol. 34, no. 3, Mar. 2011.

[73] Myrto Arapinis, Loretta Mancini, Eike Ritter, Mark Ryan, Nico Golde, Kevin Redon, and Ravishankar Borgaonkar, "New privacy issues in mobile telephony: fix and verification," in *Proceedings of the 2012 ACM Conference on Computer and Communications Security*, ser. CCS '12, New York, NY, USA, 2012, pp. 205–216.

[74] Beekman, Jethro G. and Thompson, Christopher, "Breaking Cell Phone Authentication: Vulnerabilities in AKA, IMS and Android," in *Proceedings of the 7th USENIX Conference on Offensive Technologies*, ser. WOOT'13. Berkeley, CA, USA: USENIX Association, 2013.

[75] H. Kim, D. Kim, M. Kwon, H. Han, Y. Jang, D. Han, T. Kim, and Y. 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*, ser. CCS '15. New York, NY, USA: ACM, 2015, pp. 328–339.

[76] G. Tu, C. Li, C. Peng, and S. Lu, "How voice call technology poses security threats in 4G LTE networks," in *2015 IEEE Conference on Communications and Network Security (CNS)*, Sep. 2015, pp. 442–450.

[77] C.-Y. Li, G.-H. Tu, C. Peng, Z. Yuan, Y. Li, S. Lu, and X. Wang, "Insecurity of Voice Solution VoLTE in LTE Mobile Networks," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2015, pp. 316–327.

[78] R. P. Jover, "LTE security, protocol exploits and location tracking experimentation with low-cost software radio," *ArXiv Technical Report*, Jul. 2016.

[79] A. Shaik, R. Borgaonkar, N. Asokan, V. Niemi, and J.-P. Seifert, "Practical attacks against privacy and availability in 4G/LTE mobile communication systems," in *Proceedings of the 2016 Network and Distributed Systems Security Symposium (NDSS)*, 2016.

[80] G.-H. Tu, C.-Y. Li, C. Peng, Y. Li, and S. 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*, ser. CCS '16. New York, NY, USA: ACM, 2016, pp. 1118–1130.

[81] E. Barkan, E. Biham, and N. Keller, "Instant ciphertext-only cryptanalysis of GSM encrypted communication," *Journal of Cryptology*, vol. 21, no. 3, pp. 392–429, 2008.

[82] O. Dunkelman, N. Keller, and A. Shamir, "A Practical-Time Attack on the A5 / 3 Cryptosystem Used in Third Generation GSM Telephony," *International Association for Cryptologic Research (IACR)*, vol. 8, no. December 2009, pp. 393–410, 2010.