Topics in Cell Phone Security

by Jethro Gideon Beekman

Research Project

Submitted to the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, in partial satisfaction of the requirements for the degree of **Master of Science, Plan II**.

Approval for the Report and Comprehensive Examination:

Committee:

John Manferdelli
Research Advisor

Date

* * * * * *

David Wagner
Research Advisor

Date

Abstract

The global cell phone network is a large and multi-faceted technology that is continuously being improved with new protocols and features. In this work we analyze the security of a few designs and implementations comprising a part of this network. First, we analyze the security of an IP Multimedia Subsystem (IMS) implementation for Android by a major US cell phone carrier, finding a man-in-the-middle attack. Secondly, we look at the 3GPP Authentication and Key Agreement (AKA) protocol, describing three new attacks on AKA in the context of Internet calling and Android. We have worked with the relevant parties to address these four attacks. And finally, we discuss the security aspects of modems in phone platforms from a systems design standpoint, highlighting threats and security objectives that can be used both in evaluating existing implementations as well as in creating new implementations.

Chapter 3 is joint work with Christopher Thompson and has been previously published as [1]. Reproduced with permission.

Chapters 4 and 6 and parts of Chapter 2 are joint work with Christopher Thompson and parts thereof have been previously published as [2]. Reproduced with permission.

Contents

Chapter 1

Introduction

Many telecommunication companies are moving ever more services to Internet-based platforms, citing flexibility, cost savings and evolvability [3]. We've seen similar transitions for (e-)mail, for television, and now telephony.

In the interim, some service providers have created Internet calling services based on IP telephony frameworks, which let users make and receive calls as they normally would—with their regular phone number—even when they do not have cellular reception, as long as they have another Internet connection. As more and more customers start using these kinds of services, analysis is required to protect the security of customers' communications.

In this work, we analyze the security of a typical mobile phone platform as well as these new IP telephony systems, especially SIP and related protocols and IP Multimedia Subsystem (IMS) framework. We identify 7 security objectives and 10 threats generally applicable to mobile phone platforms.

Our analysis uncovered a security vulnerability in the Wi-Fi Calling service provided by T-Mobile, which is based on the IMS framework and the SIP protocol. We believe that some, if not all, of our results may be applicable to other providers with Internet calling services. We have worked with T-Mobile to fix this vulnerability.

We have also identified three attacks on the 3GPP Authentication and Key Agreement (AKA) protocol, two of which are new and one which is a new application of an older attack to the context of AKA and IMS. These attacks apply generally to Android smartphones, even without an Internet calling service. We explain the possible scope of the attacks we describe, and present clear, actionable solutions that prevents each attack.

The remainder of this report is organized as follows. In Chapter 2, we present current protocols, designs, and relevant research. Chapter 3 covers our vulnerability analysis of the T-Mobile Wi-Fi Calling service. Chapter 4 covers the attack vectors that AKA and IMS create as well as our attacks on the 3GPP AKA protocol. In Chapter 5, we discuss the security evaluation of a typical phone platform. Chapter 6 gives an overview of related work. Finally, we conclude our work in Chapter 7.

Chapter 2

Background

Telephony and, by extension, Voice-over-IP is a huge, many-faceted ecosystem comprising many networks and individual systems. These systems communicate in different ways and are secured in different ways. This section presents an overview—with extra focus placed on security aspects—of the different mechanisms and protocols that are relevant to this work. Those include SIP, the standard VoIP protocol; SSL/TLS, the standard secure channel protocol; 802.11 ("Wi-Fi"); and various 3GPP (3rd Generation Partnership Project) specifications, used for cellular communication.

2.1 Cellular communication

2nd generation (2G) cellular communication has many known security flaws. For example, GSM suffers from several design and implementation flaws in its proprietary cryptographic primitives [4]. There has been significant research analyzing the security of cellular communication, however it mostly applies to 2G protocols [5]. The 3rd Generation Partnership Project (3GPP) has sought to fix many of these problems, and the specifications are publicly available [6]. 3GPP defines the Authentication and Key Agreement (AKA) Protocol [7], which is used to register mobile devices to cellular networks. The AKA protocol aims to provide mutual authentication as well as confidentiality and integrity protection, using a pre-shared key. On the mobile device, these algorithms and keys are usually implemented in an embedded smart card (SIM card). Service providers are free to use any algorithm they want with AKA, but the 3GPP provides an example set based on the AES block cipher [8].

Figure 2.1 shows a typical setup for a cellular network. The base station and home network are connected with a secure channel—how that channel is established is out of scope of the AKA protocol. The phone and the base station are connected via a wireless link. Figure 2.2 shows how the protocol works. First, the mobile device identifies itself to the base station using its mobile identity (e.g., IMSI). If the base station is out of cached authentication vectors for this device, it relays the identity to the home network which will reply with

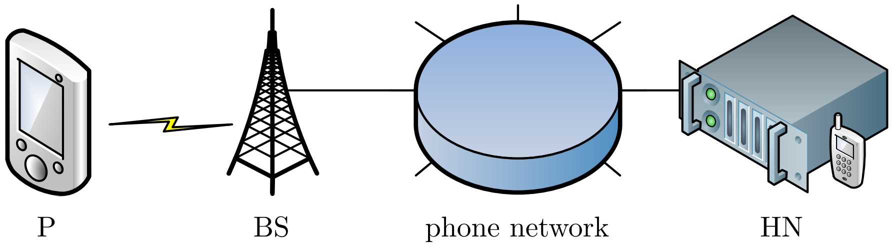
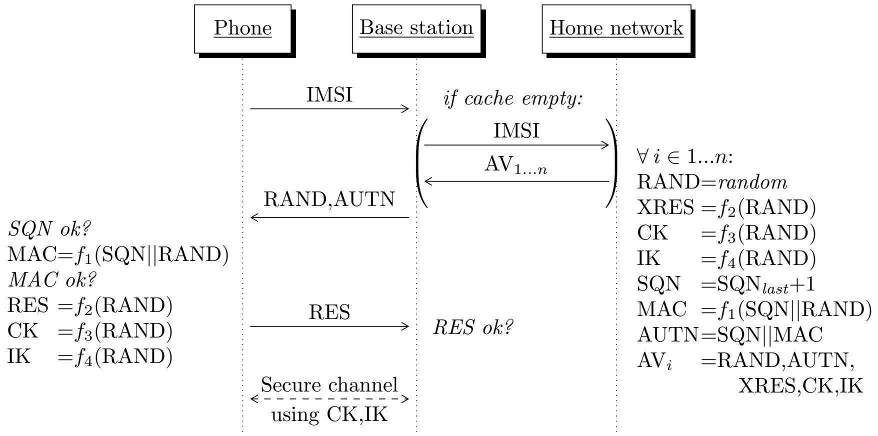


Figure 2.1: A typical cellular network setup. (P) Phone, (BS) Base station, (HN) Home network. The base station need not be operated by the home network, as long as there is some communication channel between the two.



AUTN	Authentication token	MAC	Message authentication code
AV	Authentication vector	RAND	Random number
CK	Confidentiality key	RES	Authentication response
IK	Integrity key	SQN	Sequence number
IMSI	International Mobile Subscriber Identity		

Figure 2.2: A simplified diagram of the 3GPP Authentication and Key Agreement Protocol. The home network keeps track of the last generated sequence number. The phone keeps track of the last used sequence number. In reality, the sequence number might also be encrypted with the anonymity key AK (not shown). The functions $f_1, \ldots, f_4$ are implicitly keyed with a device-specific key (not shown here) that is shared between the phone's SIM card and the home network.

a fresh set of authentication vectors. An authentication vector consists of a random challenge RAND from which all other data in the vector is based, an authentication token AUTN, an expected authentication response XRES and the confidentiality and integrity keys CK and IK.

The base station selects an unused authentication vector and sends the random number and authentication token to the device. The device checks the MAC and sequence number. If they are authentic, it generates the authentication result RES and the confidentiality and integrity keys CK and IK. It sends RES to the base station, which checks it against XRES. Now, the phone and the network have authenticated each other and a secure channel has been established using CK and IK.

2.2 Phone design

Figure 2.3 shows a typical system-level design of a cell phone. The different components are:

Mobile Application Processor (MAP) controls the modem and provides an interface to the user. For example, it is the MAP that would command the modem to initiate a call to a given phone number. The MAP might be, but doesn't need to be, on a physically different processor or chip.

On Android, there is a service called ModemManager that abstracts the modem interface to the rest of the Android ecosystem.

Modem interfaces with the network, and runs the required networking protocol stack. Also called Communications Processor (CP) or baseband. Each modem is uniquely identified to the network by the International Mobile station Equipment Identity (IMEI). The primary interface by which the modem can be controlled by the MAP are AT commands [9], which will be explained in the next section.

Subscriber Identity Module (SIM) is a smartcard provided by the phone service provider. Responsible for user (subscriber) identification and authentication. Without a SIM, the phone will not be able to register to

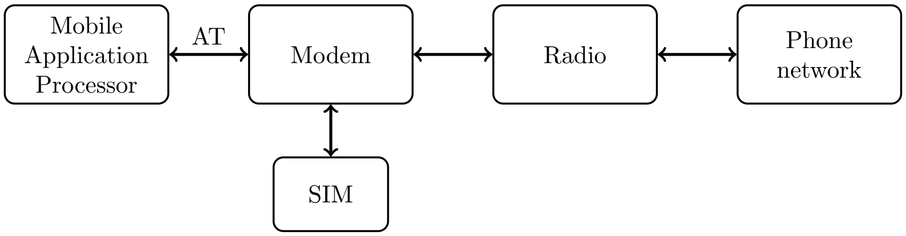


Figure 2.3: System-level overview of a typical mobile phone.

the phone network and can make only emergency calls. Phones can be configured to work with only certain SIMs, this is called a “SIM lock”.

Radio sends and receives electromagnetic RF waves, interfacing with the wireless phone network.

AT Commands

AT commands [9] are an industry standard interface for controlling a modem. Some sample commands are:

D	Dial
A	Answer call
+CGMI	get manufacturer information
+CEN	get list of emergency numbers

There are hundreds of standard commands, most of them starting with +C. Implementations can also define their own commands, these start with +X. Many of these commands give access to sensitive information. For example +CEAP and +CERP access the SIM authentication routines (see Section 5.1) for use with the Extensible Authentication Protocol (EAP). +CSIM grants blanket access to the SIM interface.

2.3 IP Multimedia Subsystem

IP Multimedia Subsystem (IMS) [10] is a framework for delivering ‘multimedia’ services over the Internet. Here, multimedia means everything related to calling and messaging that you would do with a normal phone. It is developed by the 3GPP as part of an all-IP future of telephony and is designed with interoperability with current Internet standards in mind. The standards used are Session Initiation Protocol (SIP) for call control and signaling and Real-time Transport Protocol (RTP) for real-time multimedia data. However, until IMS is completely implemented across all levels of a service provider’s infrastructure, hand-off (switching networks during a multimedia session) between IMS and older technologies is not possible. A typical IMS setup is shown in Figure 2.4.

The Session Initiation Protocol (SIP) [11], as the name suggests, allows two hosts to manage a session between them. SIP and related standards are developed by the IETF. Zourzouvillys and Rescorla [12] describe these standards and protocols briefly. SIP allows messages to route between servers on a path between the two hosts, much like SMTP, except messages never get queued at an intermediate server. What type of session will be created is described by the Session Description Protocol (SDP) [13]. The descriptor contains information such as endpoints, routers, and media protocol, encoding and encryption. The most commonly used media protocol for VoIP is Real-Time Transport Protocol (RTP) and its counterpart RTP Control Protocol (RTCP) [14].

SIP over TCP or UDP is vulnerable to man-in-the-middle attacks. The SIP messages can be encrypted using S/MIME [15] to ensure integrity and

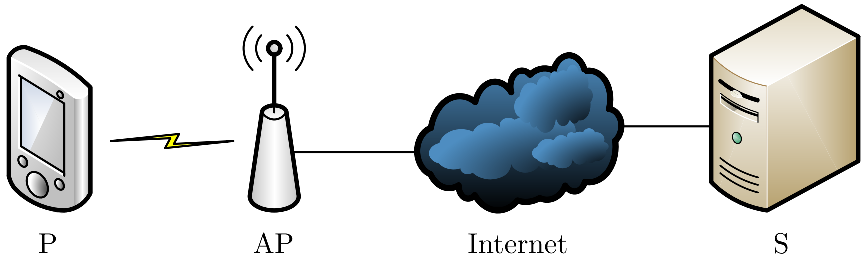


Figure 2.4: A typical IMS setup. (P) Phone, (AP) Wi-Fi access point, (S) SIP server. The SIP server is part of the phone's home network.

confidentiality. However, some SIP header information (such as To and From information) cannot be part of an encrypted S/MIME message for routing purposes. For better security, SIP connections can be encrypted by using TLS. To protect the actual voice/video data, RTP can be protected with symmetric-key encryption using SRTP [16].

SIP authentication

Authentication in SIP uses the same mechanisms as HTTP, with the WWW-Authenticate and Authorization headers. Unlike most HTTP configurations, Digest authentication [17] is often used. Digest authentication uses a challenge-response mechanism, in which a nonce is sent by the server. The client responds with a hash of the username, the password and the nonce.

AKA can also be used to perform Digest authentication. AKAv1 Digest authentication [18] specifies that RAND and AUTH are used as the nonce parameter, while RES is used as the "user password" in the response. This permits a so-called "interleaving" attack, which can occur when the same credentials are used in different contexts. AKAv2 Digest authentication [19] aims to fix this and other attacks. It explicitly has the client demonstrate that it knows the generated session keys in order to prevent man-in-the-middle attacks. The specification details four ways for implementers to prevent or reduce man-in-the-middle attacks. Despite these known attacks, AKAv1 is still in use in many widespread systems, such as IMS [20].

2.4 Transport Layer Security

Transport Layer Security (TLS, the successor to SSL) is the current standard for establishing secure channels. While cryptographically solid, some implementation issues exist. Marlinspike's Black Hat talks [21, 22] identify a few potential problems. Most of these issues have to do with authentication, specifically, certificate validation. Usually, a certificate is signed by a Certificate Authority (CA). If the verification is not done properly, man-in-the-middle attacks become trivial.

In order to execute man-in-the-middle attacks, an attacker must be on the path of the target network traffic. If the attacker is not already on the path, there are techniques for influencing the path to include the attacker. With ARP spoofing [23] the attacker tricks hosts into believing they are the router. Obviously, the attacker can only use this attack if they are not on the path but they are on a network in the path. If the attacker is not, they can use DNS cache poisoning [24], in which they trick a caching DNS resolver to cache an invalid address record for the service they want to attack. Other clients using this same resolver will now receive the malicious record when connecting to this service and will connect to the attacker instead.

Man-in-the-middle attacks would not be as threatening if not for the proliferation of wireless technologies such as Wireless LAN (802.11). In wireless networks, an attacker no longer needs physical access to invade a network. If an attacker can connect to the network they can try ARP spoofing. Or, if the attacker knows the network parameters, they can employ the 'Evil Twin' attack [25], in which they imitate a legitimate network and trick users into connecting to that network instead.

Chapter 3

Man-in-the-Middle Attack on Wi-Fi Calling

T-Mobile has a service called “Wi-Fi Calling”, which lets users make and receive calls even when without cellular service. This service is pre-installed on millions of T-Mobile Android smartphones. We analyze the security aspects of this service from a network perspective, and demonstrate a man-in-the-middle attack caused by a lack of TLS certificate validation, allowing an attacker to eavesdrop and even modify calls and text messages placed using the Wi-Fi Calling feature. We have worked with T-Mobile to fix this issue, and, as of 18 March 2013, they report that all affected customers have received an update fixing this vulnerability.

3.1 Experimental Setup

In order to analyze T-Mobile’s Wi-Fi calling system, we used the setup shown in Figure 3.1. A Wi-Fi calling enabled phone P is configured to use access point AP. AP does not have any wired connections and just acts as a wireless network switch. Another machine M connected to the same network is configured as a DHCP server and NAT router. This allows us record and control all Internet traffic to and from P.

We captured several Wi-Fi calling sessions. During our experiments, all traffic on both network interfaces was captured using libpcap. Any TLS connections were intercepted by sslsniff [26] running on M. sslsniff hijacks a TLS connection request, connects to the remote endpoint itself and generates a certificate based on what it receives from that endpoint. The certificate is signed with any certificate we specify and subsequently sent back to the client. We modified sslsniff to output the master-secret and client-random parameters of all established connections so that the TLS traffic in our packet traces could be decrypted. This allowed us to see how TLS-encrypted messages relate chronologically to other packets.

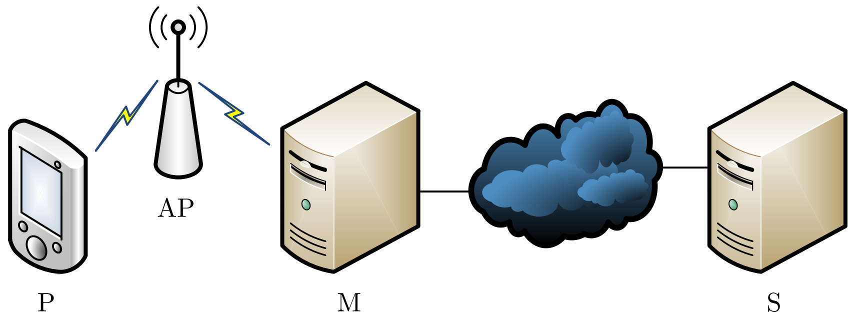


Figure 3.1: Wireless man-in-the-middle setup. (P) Phone. (AP) Access point. (M) Man-in-the-middle. (S) Service provider.

3.2 Network analysis

When first enabling Wi-Fi, the DNS conversation in Appendix A.1 takes place, requesting a chain of information about *wifi.msg.pc.t-mobile.com*. Then, a TLS connection is established to the host and port returned by DNS. These are *sba.sipgeo.t-mobile.com* and *5061*, which is the port number assigned by IANA for SIP-TLS.

The certificate chain returned by T-Mobile's server is shown in Figure 3.2. Two things stand out: first, the common name of the first certificate is simply the IP address of the server; second, the self-signed root certificate is not included in standard Certificate Authority (CA) distributions. In fact, searching the web for the exact common name of the root yielded barely any results. This can meant that the root certificate was either built-in to T-Mobile's client software, or they did not implement certificate validation correctly. In fact, the client does not seem to have any problems with sslsniff intercepting the connection, which supports the latter conclusion. Analysis of the binaries confirmed that there was no trace of the root certificate.

As hinted at by the DNS records and the port number, a SIP [11] dialog is initiated when the TLS connection is established. The client identifies itself using its phone number, IMEI (International Mobile station Equipment Identity)

```
1. s:/C=US/ST=WA/L=Bellevue/O=Engineering/CN=IP
   i:/C=US/O=T-Mobile USA, Inc./CN=T-Mobile USA, Inc. IP Access Nano 3G CA

2. s:/C=US/O=T-Mobile USA, Inc./CN=T-Mobile USA, Inc. IP Access Nano 3G CA
   i:/C=US/O=T-Mobile USA, Inc./CN=T-Mobile USA, Inc. Root CA

3. s:/C=US/O=T-Mobile USA, Inc./CN=T-Mobile USA, Inc. Root CA
   i:/C=US/O=T-Mobile USA, Inc./CN=T-Mobile USA, Inc. Root CA
```

Figure 3.2: Certificate chain of *sba.sipgeo.t-mobile.com:5061*.

and IMSI (International Mobile Subscriber Identity); see Appendix A.2 for the full message. Further messages follow normal SIP behavior. For example, an INVITE message including the SDP [13] body is outlined in Appendix A.3, which includes the encryption key that will be used for the SRTP [16] connection. Using the decrypted SIP dialog, an attacker is now able to record all incoming and outgoing calls and text messages (collectively "SIP traffic"). He could record, block and reroute SIP traffic. The attacker could change it by faking a sender or changing the real-time voice data or message content. He could fake incoming traffic and he can impersonate the client with forged outgoing traffic.

We verified the ability to record outgoing calls and incoming and outgoing text messages. We also verified the ability to change the destination phone number on outgoing calls by modifying sslsniff to change all occurrences of `<sip:`*dest-phone#*`@msg.pc.t-mobile.com>`, replacing a single target phone number by a different one.

3.3 Software analysis

T-Mobile has open-sourced most of its Android IMS stack [27]. By exploring the source code, we found two vulnerable components in categories that Fahl et al. [28] have previously identified. *TlsSocketFactory* creates a TLS socket and assigns the *TestTrustManager* to check the validity of certificates.

Figure 3.3 shows a shortened version of the function that verifies the certificate chain. Note that only the first element of `chain` is used, so there is no root of trust. The only check is whether the common name – extracted from the distinguished name without a proper API – is in a set of allowed IP addresses, which was previously initialized with the IP address of the remote endpoint. This means that even if proper certificate chain validation was implemented, a network attacker who controls DNS (e.g., by cache poisining, local spoofing, a man-in-the-middle attack) could redirect the client to one of it's own IP addresses for which it could easily get a valid certificate.

```
public void checkServerTrusted(X509Certificate[] chain)
    throws CertificateException
{
    String name=chain[0].getSubjectDN().getName();
    String address=name.replaceFirst("CN=", "").replaceFirst(",.*", "");
    if (!getAllowedRemoteAddresses().contains(address))
        throw new CertificateException("Wrong certificate subject");
}
```

Figure 3.3: Implementation of *TestTrustManager::checkServerTrusted*, which checks the validity of a TLS certificate.

3.4 Scope

In order to execute man-in-the-middle attacks, an attacker must be on the path of the target network traffic. If they are not already on the path, there are techniques for influencing the path to include the attacker. With ARP spoofing [23] the attacker tricks hosts into believing the attacker's network interface has the router's IP address. Obviously, the attacker can only use this if they are on an on-path network. If the attacker is not, they can use DNS cache poisoning [24], in which they trick a caching DNS resolver to cache an invalid address record for the service they want to attack. Other clients using this same resolver will now receive the malicious record when connecting to this service and will connect to the attacker instead.

Man-in-the-middle attacks would not be as threatening if not for the proliferation of wireless technologies such as Wireless LAN (802.11)—exactly the technology that Wi-Fi calling advertises in its name. In wireless networks, an attacker no longer needs physical access to invade a network. If an attacker can connect to the network they can try ARP spoofing. Or, if the attacker knows the network parameters, they can employ the 'Evil Twin' attack, in which they imitate a legitimate network and trick users into connecting to that network instead. Dai Zovi and Macaulay [25] demonstrated an extended version of this attack exploiting bugs in automatic network selection algorithms in common operating systems.

Not all versions of T-Mobile Wi-Fi calling are necessarily vulnerable. According to T-Mobile's website, the IMS stack is used on the Samsung Galaxy S II, HTC Amaze 4G, myTouch and myTouch Q. The authors have tested the attack on a Samsung Galaxy S Relay 4G and a Samsung Galaxy Note 2. It is likely that other modern T-Mobile Samsung Galaxy products are also vulnerable. Users of T-Mobile Wi-Fi calling for Business might not be vulnerable, since it uses GAN, not IMS technology [29].

3.5 Patch and Updates

In December 2012, we notified T-Mobile of this vulnerability. Over the past months, they added proper certificate validation to the T-Mobile Wi-Fi Calling feature, so that it validates the identity of the remote endpoint using their self-signed root CA. As of 18 March 2013, T-Mobile reports that they have been able to push an update with this patch to all affected customers. We have independently verified that the update pushed to T-Mobile Android phones successfully prevents this attack.

Chapter 4

Attacks on the AKA Protocol

We look at two similar ways for a cellular phone to connect to the network: using the cellular network and using IMS. Both methods use the AKA protocol to authenticate the user, and rely on the same protocols implemented in a smart card on the phone for securely interfacing with shared keys.

Figure 4.1 shows the system and the three channels where various authentication data is transferred between principals. The first channel (1) is the interface between the smart card and the phone operating system, over which the inputs and outputs of smart card functions are transferred. The second (2) is the IMS channel (SIP connection), and the third (3) is the cellular connection. These last two serve a similar function, but for different types of service. All channels convey in some form the challenge RAND and token AUTN in the direction of the phone or smart card. The smart card sends the authentication result RES and the session keys CK and IK in the direction of the network. The SIP channel transports a hash H(RES) or H(RES,CK,IK,"AKAv2") for AKA digest authentication versions 1 and 2, respecively. The 3G channel sends RES to the network and uses CK and IK, but doesn't transmit them.

Because the same protocol secrets are reused in the different channels, it is possible for an attacker to conduct man-in-the-middle attacks by connecting the channels in unexpected ways, similar to a chosen-protocol attack [31]. For example, one could intercept the authentication response RES from a 3G link and use this for the hash required in the AKAv1 digest authentication mechanism (see Section 4.1 for a more detailed look at this attack). An attacker cannot combine all channels in all possible ways. For instance, it is not possible for an attacker to use information transfered over SIP/AKAv2 to authenticate to the 3G network, because the phone does not provide the attacker the necessary authentication data. The SIP channel using AKAv2 only sends the cryptographic hash of the data, while the attacker must have the session keys to act on the 3G network. However, many combinations of these channels *do* expose security vulnerabilities. Table 4.1 enumerates the ways in which the different channels could be connected by an attacker and we analyze these ways in the following sections.

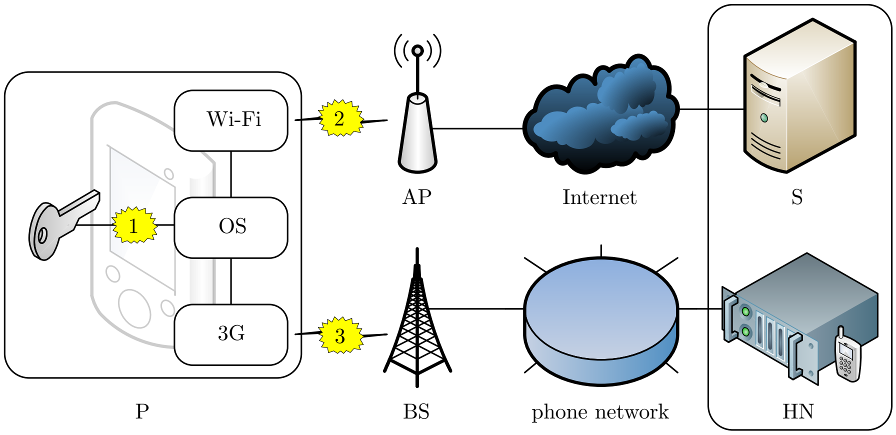


Figure 4.1: How a phone connects to the IMS and cellular systems from Figures 2.1 and 2.4. The SIP server, S, and Home Network, HN, are operated by the same entity. The authentication data channels 1, 2 and 3 indicate where various authentication data is transferred between principals.

Table 4.1: Man-in-the-Middle attacks possible

Phone side	Network side		
	SIP/AKAv1	*SIP/AKAv2*	*3G AKA*
Smart card	Y, §4.2	Y, §4.2	Y, §4.3
SIP/AKAv1	Y, §3	N[H,K]	N[H,K]
SIP/AKAv2	N[H]	Y, §3	N[H]
3G AKA	Y, §4.1	N[K]	Y, [30], §6

[H] Some or all of the authentication data we need is contained in a hash we can't invert.

[K] Session keys CK and IK are required to authenticate to the network but are not sent by the phone.

Each cell indicates whether an attack is possible when connecting the mechanism used on the phone on the left side to the mechanism used in the network on the top. If an attack is possible, we indicate where we discuss this attack. If an attack is not possible, we indicate why. It does not make sense to connect a smart card on the network side, as a SIP or 3G channel will eventually connect to the home network.

In this chapter we show three attacks on AKA, which are caused by poor cross-protocol interaction and implementation issues. These attacks are based on the poor use and handling of the AKA session keys.

4.1 False base station attack

We discovered that AKA is secure if and only if the authentication step of AKA is bound to the secure communication channel using the confidentiality and integrity keys produced by AKA. The Digest Authentication with AKA RFC [18] states in Section 5.5, *Session Protection*:

> Digest AKA is able to generate additional session keys for integrity (IK) and confidentiality (CK) protection. Even though this document does not specify the use of these additional keys, they may be used for creating additional security within HTTP authentication or some other security mechanism.

This statement provides misleading advice to implementors. If the session keys IK and CK are not used to protect the subsequent session, man-in-the-middle attacks become possible.

An attacker can impersonate a subscriber using a so-called *false base station attack* [32]. In this attack, the attacker controls the false base station F and an Internet-connected host M, as shown in Figure 4.2. Now consider the sequence of events in Figure 4.3. P is convinced to connect to base station F, and P sends its IMSI to F. M uses this to initiate a SIP connection with S. S will respond with the challenge (RAND, AUTN), which M will relay via F to P. P thinks it is authenticating to the network and will simply respond with RES. M gets RES from F and uses it to compute the Digest Authentication response. The server accepts the authorization header as if it had come from a legitimate client. M can now make and receive calls or text messages through the SIP server S using P's account.

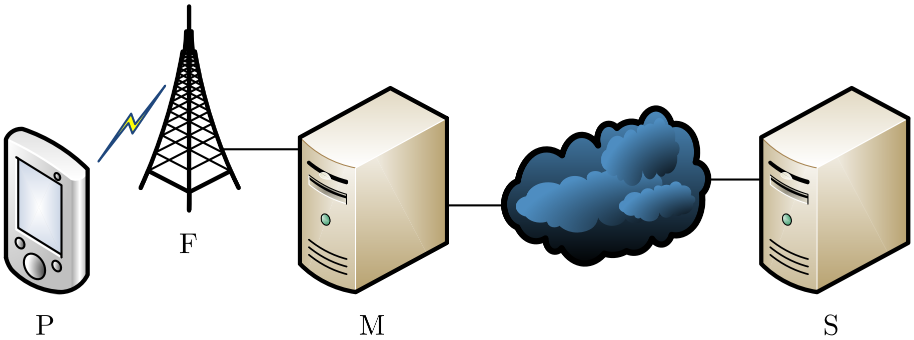


Figure 4.2: Setup for the false base station attack. (P) Phone. (F) False base station. (M) (On-path) attacker. (S) SIP server.

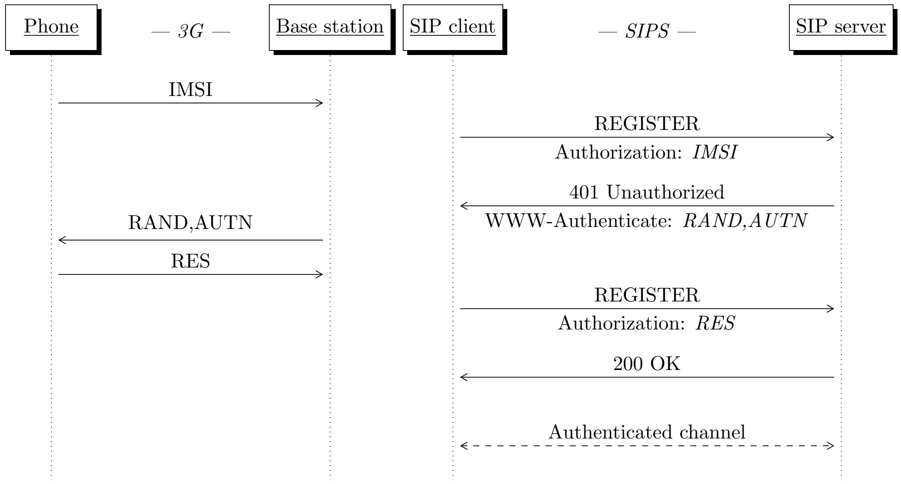


Figure 4.3: False base station attack, exploiting interactions between 3G and Digest Authentication AKA. Both the base station and the SIP client are under the attacker's control. In reality, the Authorization and WWW-Authenticate headers are much more verbose.

4.2 Malware attack

The false base station attack requires the attacker to be near the victim and to invest in a false base station.

We describe a second attack that avoids these requirements, if the attacker can get the user to install a malicious app on their Android phone. The attack takes advantage of the fact that on certain versions of Android, any app can interact with the smartcard (only the READ_PHONE_STATE permission is needed). This allows applications to call the *requestIsimAuthentication* API, which returns the authentication response RES as well as the confidentiality and integrity session keys (CK and IK).

The READ_PHONE_STATE permission is described to the user as:

> **Phone calls** *Read phone state and identity*
> Allows the application to access the phone features of the device. An application with this permission can determine the phone number and serial number of this phone, whether a call is active, the number that call is connected to and the like.

About a third of Android applications request this innocuous-looking permission [33]. In newer versions of Android, applications get this permission by default.

In this attack, as shown in Figure 4.4, the phone P is connected to an access point AP. The attacker E waits for the malicious application to contact them over the Internet. When it does, E's SIP client connects to the SIP server S and

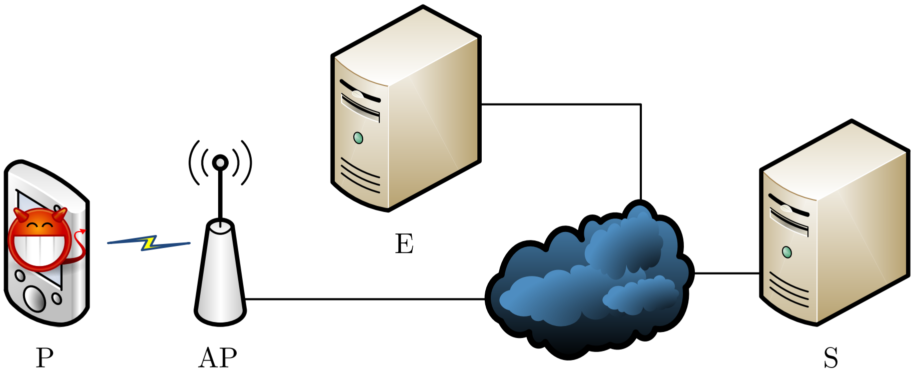


Figure 4.4: Setup for the malware attack. (P) Phone with malware installed. (AP) Access point. (E) (Off-path) attacker. (S) SIP server. In this figure, the phone is connected to the Internet via Wi-Fi, but it could also be connected via a 3G data connection instead.

receives a challenge. E then responds to the malicious application with this challenge. The application calls the *requestIsimAuthentication* function and sends back the response to E. E uses this to compute the Digest Authentication response and can now make and receive calls or text messages through the SIP server using P's account.

4.3 Imposter attack

The *requestIsimAuthentication* API also returns the confidentiality and integrity keys. Obtaining these keys enables a different attack, in which the attacker controls a mobile device I (see Figure 4.5). This attack is similar to the the malware attack, but instead of using the SIP connection, the attacker uses I to impersonate P, requesting the challenge from a legitimate base station BS. I relays the token via E to the malicious application, and the application responds with the AKA response and the confidentiality and integrity keys. I sends the response back to BS and successfully authenticates as P. In addition, I has the session keys needed for further communication.

4.4 Scope

The false base station attack on AKA is applicable to any Internet service that use AKA authentication with a mobile device.[1] The malware and imposter attacks on AKA are possible on all phones that expose these authentication APIs. Version 4.0 and above of stock Android (both stock images for the Samsung Galaxy Nexus as well as the Android Open Source Project) have this API.[2]

[1] Such as T-Mobile Wi-Fi Calling.

[2] We have tested the malware attack on a Samsung Galaxy S Relay 4G, and we believe that it extends to all phones with the same implementation of T-Mobile's Wi-Fi Calling.

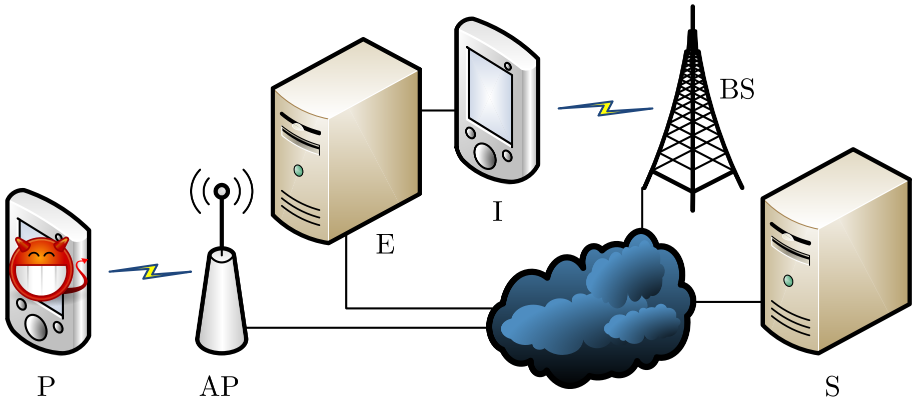


Figure 4.5: Setup for the imposter attack. (P) Phone with malware installed. (AP) Access point as in Figure 4.4. (E) (Off-path) attacker. (I) Imposter mobile device. (BS) Legitimate base station and (S) Service provider as in Figure 2.1.

4.5 Analysis and solutions

A problem with the false base station attack is that F cannot decrypt any other traffic from P, because it does not have the confidentiality and integrity keys. Therefore, M cannot relay any legitimate traffic. This might make P suspicious. The malware attack does not have this issue. To prevent the false base station attack, the SIP channel must also use the confidentiality and integrity keys. For example, upon receiving a valid Digest Authentication response from the client, the server could force TLS renegotiation with one of the pre-shared key cipher suites [34] using CK and IK as the pre-shared keys. M does not have these keys and therefore cannot proceed with the renegotiation. The RFC should be updated to stress the importance of the confidentiality and integrity keys, for example by replacing the phrase "they may be used" with "they MUST be used".[3] Another option is to use AKAv2, which was designed in part to defend against attacks like this.

To prevent the malware and imposter attacks, the *requestIsimAuthentication* API needs to be secured.[4] The authors cannot think of a valid reason why any third party application should need to access this function. AKA is part of the security core of cellular communication and should only be accessible to system software.

These implementations use the different but similar *calculateAkaResponse* API.

T-Mobile has an option for users to turn Wi-Fi Calling on and off. Presumably, turning it off disables SIP service—and thus the false base station and malware attacks—for that account. However, the exposed API can still be used for the imposter attack.

[3] The RFC authors have been contacted, although no concrete action has been taken as of this writing.

[4] Android security bug #1305406714.

Chapter 5

Phone Platform Threat Analysis

With smartphones becoming targeted more and more by malware, it is necessary to take a good look at the security of a phone as a whole. This chapter describes an evaluation of the security of a typical phone platform, focusing specifically on the modem.

Modems are used in phone and tablet platforms to connect to 4G/3G/2G "phone" networks, for voice and data connectivity. Modems used to be only discrete solutions, but newer products increasingly integrate modems into phone platforms (e.g., on the same chip). This eliminates duplicate functionality and reduces the cost.

In this chapter, we look at the continued integration of modems in these mobile platforms from a security perspective. The general structure of and the threats to mobile platforms are analyzed.

Threat model

This analysis is done from the perspective of a phone manufacturer that wants to create a platform that is secure for their users, while maintaining compliance with industry and regulatory standards. As such, we define the following adversaries:

Regular end user – A user, normally the owner of the phone, who uses it to make calls, text, surf the web, etc. The user might want to make free calls or do other things for their personal gain, but wants to be able to keep using the phone.

Malicious end user – A malicious person that has physical access to the phone, either once or multiple times. Examples include an "evil maid" or a phone thief.

Malware – Malicious software that is (surreptitiously) installed on the phone, and which is used to interfere with its operation, or obtain sensitive information or access to other systems.

It is assumed that all the adversaries could have rooted/jailbroken access to the phone.

Scope

Physical attacks other than those performed by a simple adversary are considered out of scope. A simple adversary is one with limited skills and financial means. For example, acquiring installing a modchip would be just within such an adversary's capabilities.

Exploits that are not physical attacks but where an adversary needed to do a physical attack to develop the exploit (BORE: Break Once Run Everywhere) are in scope. An example of such an attack would be the extraction of a global secret key from a single device.

Also out of scope are phone network attacks, both by and to the network. Availability is also out of scope; there are too many factors that limit the continuity of cell phone service, e.g., battery power, physical damage, network coverage.

Methodology

This material is assembled through review of industry standards, as well as through review of design documents from a large phone parts manufacturer. We focus our review on features and the like that are the result of a physical need or an external requirement (e.g., phone numbers) and not those that result from implementation details (e.g., firmware, cryptographic keys).

In Section 2.2 and Figure 2.3 on page 5 a typical phone design is described. From the components in this design, we identify the interfaces, security features and assets related to the modem listed in Table 5.1. In each of the following sections, we give an overview of the unit and we identify security objectives, threats and Trusted Code Base (TCB) relating to the unit. (For Radio calibration data, we only give an overview and list the threats.)

The security of the SIM itself is not evaluated; the phone service providers are responsible for its security.

Table 5.1: Security requirements for different phone units

Unit	Access control	Confidentiality	Integrity
SIM access	✓		
SIM lock			✓
Emergency numbers			✓
IMEI			✓
SIM PIN		✓	✓
Radio calibration data			✓

5.1 SIM access

The Subscriber Identity Module (SIM) is a smartcard provided by a service provider to a customer and is used to authenticate and potentially secure phone network communication. It also has auxiliary uses, such as the storage of contacts and text messages, and it can even run Java applets [35]. Full and direct access to the SIM means access to the network.

The subscriber identity – including International Mobile Subscriber Identity (IMSI) and Temporary Mobile Subscriber Identity (TMSI) – is stored on the SIM and is used to identify the user to the network [7, §6.1-2]. The MAP (Mobile Application Processor) does not need to know this information, although a quick internet search on "Android IMSI" reveals that most phones do expose it in some way, presumably through the use of the AT+CIMI command.

The SIM is a party in the Authentication and Key Agreement (AKA) protocol [7, §6.3], which is used during network registration to authenticate the SIM (and thus the user) and to setup session keys for confidentiality and integrity protection. In a standard phone network setup, the MAP doesn't need to access the AKA interface. However, newer uses may require MAP access to the authentication routine – uses such as IP Multimedia Subsystem (IMS, "Phone network over IP") and the EAP-SIM and EAP-AKA authentication methods [36, 37].

The SIM feature to store contacts and text messages is hardly used anymore. These are usually stored by the MAP in some other way. Modern phones do usually have a feature to import this stored data, so the MAP needs to be able to access it.

There might be other use cases for SIM access not covered here.

Security Objectives

- Only entities which have a specific need to communicate with the SIM may access it, and that access should be restricted to just the functions they need.

  For example, the MAP might need access to the identity, phone book and stored messages, while the modem will need access to the identity and authentication routines. But authentication may not always be limited to the modem. Technologies such as IMS may require the MAP to access the SIM authentication routine.

Threats

The end user is not considered an adversary for SIM access control. A user is generally both allowed and able to remove a SIM from their phone. The user can then use it with a different device or connect it to a computer to have raw access.

- Malware that has access to the SIM authentication routines can assume the end user's network identity, see Sections 4.2 and 4.3.
- A user's privacy can be compromised by malware that has access to the stored SIM data.

  It should be noted that the stored SIM data is probably of much lower value than the other information that could be more readily available to malware. However, depending on the implementation, SIM card access might be a different permission than access to MAP contacts/messages, and hence access control is still required. The user might not be aware that such data is stored on the SIM while evaluating the privacy risk of an application that requests generic SIM access.

TCB

The TCB for SIM access control is the modem firmware, as well as the MAP ModemManager and services that have privileged access to ModemManager. This means that root-level malware has automatic access to the SIM with the same privilege level as the MAP.

Security Recommendations

To reduce the access that the MAP has to the SIM, generic access AT commands should be blocked to improve security. However, since AT commands are a standardized interface, it is necessary to implement all commands for standards compliance, even if they make the platform insecure. It might be possible to create some sort of configurable firewall on the modem to protect the platform. For example, a white list of secure commands could be programmed in the modem firmware and all non-whitelisted commands would be ignored.

In order to support network authentication for applications such as IMS, the modem should implement the entire IMS stack directly instead of exposing the complete authentication interface to the MAP. This way, the modem is the encryption endpoint and the session keys never have to leave the modem.

5.2 SIM lock

SIM Subsidy lock (SIM lock in short, also called personalization) is the ability of a phone device to be use-limited to a certain subset of SIMs. The main usage, as the name implies, is to prevent use of a subsidized device with a SIM that is not from the subsidizer. The mechanism is implemented entirely on the device. The SIM just provides identity information: IMSI, GID1 (Group IDentifier level 1) and GID2. This identity is hierarchical: the IMSI contains a country code (MCC, digits 1-3) and a network code (MNC, digits 4-5), the GID1 specifies the service provider, etc. A SIM lock describes which part of the identity must match the data specified in the SIM lock. Different SIM lock subsets are defined in 3GPP TS 22.022 [38]. Table 5.2 specifies what part of the SIM identity must match for different lock types, each type specifying an increasingly smaller subset of SIMs.

The most specific lock type matches the entire IMSI, which should be used as a lock-on-first-use, where the IMSI of the first SIM card inserted is recorded and locked. Afterwards the phone will only work with this specific SIM. This is meant as a theft-prevention mechanism: the SIM will be blacklisted after a theft and the phone won't work with a different SIM due to the binding. It is also possible to implement custom (device specific) lock types that might also require custom data stored on the SIM.

The SIM lock system has two persistent states:

Personalized – The device has a lock specified for a subset of SIMs.

Depersonalized – The device can be used with any SIM.

And two run-time states:

Locked – The device is personalized and an invalid SIM is inserted.

Unlocked – The device is depersonalized, or it is personalized and a valid SIM is inserted.

During the phone boot process, the modem will verify the SIM against stored personalization information, if any, and set the run-time state accordingly. In the locked state, the modem will provide a means for depersonalization. This is generally implemented using a so-called "unlock code" which can be provided to the user at the subsidizer's discretion. If the unlock code is correct, the personalization information will be invalidated or removed, so that the depersonalized state persists. SIM lock implementations typically include countermeasures against unlock code brute-forcing, such as a maximum number of tries, or a retry timeout.

Table 5.2: Different types of SIM Subsidy locks

Identity Lock type	Network IMSI 1-5	Network Subset IMSI 6-7	Service Provider (GID1)	Corporate (GID2)	SIM/ USIM IMSI 8-15
Network	✓				
Network subset	✓	✓			
SP	✓		✓		
Corporate	✓		✓	✓	
SIM/USIM	✓	✓			✓

Security Objectives

- A personalized device shall only work with a specific subset of SIMs.
- Only a user with a valid unlock code shall be able to depersonalize the device.

Threats

- An end user wanting to use a subsidized device with a different network.
- An end user (phone thief) wanting to use an IMSI-locked phone with a different SIM.

TCB

The TCB is modem firmware and lock information storage. Since the modem is responsible for enforcing the lock, subverting the firmware will result in an unlocked or depersonalized device. If the lock information can be modified, different (less restrictive) locking rules could be configured.

5.3 Emergency numbers

The modem has a “limited service mode” (also: “emergency calls only mode”, “restricted mode”, “restricted/limited access mode”, etc.), which means that only emergency calls are allowed. Limited service mode will be entered in the following cases:

- No SIM is inserted.
- The device is SIM locked (see Section 5.2).
- Regular service is prohibited on all networks in range.
- There was an error during the secure boot verification of the modem (implementation-defined).
- Potentially other cases not covered here.

Emergency numbers are defined in 3GPP TS 22.101 [39, §10]. Certain numbers should be hard-coded in the phone, while others are stored on the SIM and yet other numbers may be downloaded from the phone network.

Security Objectives

- A device in limited service mode shall be able to make only emergency calls.

- Only phone numbers specified by the SIM, preprogrammed in the device, or downloaded from the phone network shall be available for emergency calls.

Availability is out of scope for this analysis. The MAP is responsible for directing the modem to do an Emergency Call, and the user needs to initiate through some sort of UI. This UI can be manipulated by malware. 3GPP TS 22.101 [**?** , §10.1]iscusses an emergency call button, but it is not a requirement. If and when Secure UI is added to phones, availability of emergency calls should be reconsidered as a security objective.

Threats

- An end user wanting to subvert subsidy lock/roaming restrictions by changing the set of emergency numbers. Note that the network the user is on might not accept emergency calls to non-emergency numbers.

- Malware redirecting emergency calls, possibly in a manner invisible to the end user.

TCB

The TCB for these objectives is the modem firmware and emergency number storage. Since the modem is responsible for initiating calls, subverting the firmware will result in being able to initiate emergency calls to any number. If the stored numbers can be modified, different numbers could be programmed.

If availability were to be added as an objective, the TCB would need to be re-evaluated.

5.4 IMEI

The International Mobile station Equipment Identity (IMEI) identifies the device, as opposed to the IMSI, which identifies the user (subscriber). Some the design goals for the IMEI are described in a 3GPP terminal security working group note [40]:

> - Deter using stolen terminals
> - Blacklisting type non-approved terminals
> - Identify emergency call terminal[1]
> - SIM lock[2]

[1] One can make emergency calls without a SIM card, in which case the IMEI will still provide some form of identification.

[2] The use case here is presumably meant as a way to tie SIM lock data to a specific device.

3GPP TS 22.016 “defines the principal purpose and use of [IMEI]” [41, §2] and states in particular: “The IMEI shall be unique and shall not be changed after the ME’s final production process. It shall resist tampering, i.e. manipulation and change, by any means (e.g., physical, electrical and software).”

The IMEI is communicated to the network during phone registration. The IMEI is also available to the MAP through the +CGSN AT command, where it can be used as a unique identifier for the phone.

Security Objectives

- The IMEI of a device shall not be changed after manufacturing.

Threats

- An end user wanting to use an unapproved device, i.e., a device not approved by a service provider for use on its network.
- An end user (phone thief) wanting to use a blacklisted device.

TCB

The TCB is modem firmware and IMEI storage. Since the modem is responsible for communicating the IMEI to the network, subverting the firmware will result in being able to change the IMEI. If the stored IMEI can be modified, a different IMEI could be programmed.

5.5 SIM PIN

Users can configure a PIN on their SIM, which will need to be entered every time the SIM is (re)powered. Normally, a user would enter the PIN through a UI on the MAP on boot. The MAP will then send this to the modem, which will pass it to the SIM.

Without the PIN, the SIM will not operate. As such, it is a way for a user to protect his network identity (phone number, receiving calls, etc.). After entering an invalid PIN 3 times, the PIN will be disabled and a PUK (PIN Unlock Key) needs to be entered. The PUK cannot be changed by the user and is generally longer and stronger than the PIN.

The MAP and the modem both might want to remember the PIN so that the user need not re-enter it in case of a soft reset, for example due to disabling airplane mode, a crash or watchdog timeout. This opens up an attack vector where an adversary might be able to read the PIN from RAM.

Threats

- An end user (SIM thief) wanting to assume the SIM owner’s network identity. The SIM might have been stolen independently, or the adversary might have had access to the phone as well, for example swapping the SIM.

Security Objectives

- PIN protection by the MAP/modem should be such that performing a physical attack on the SIM is at least as cost-effective as reading the PIN from MAP/modem RAM.

5.6 Radio calibration data

During phone manufacturing, the radio is calibrated. This device-specific calibration data needs to be stored on the phone and can't be recreated in the field. As such, deletion of this data is a persistent DoS attack.

An open question remains whether there are any other adversaries and threats, besides the PDoS attack. For example, could the end user manipulate the calibration data and boost the phone signals beyond regulatory permissible values?

Threats

- Malware blocking network access even after it has been removed. Note that, as described in the section on Emergency Calls, malware has many ways to deny service, but usually not in a persistent way.

5.7 Discussion

We identify 7 security objectives for and 10 threats to the phone platform. The objectives and threats identified can be used in future work to review current implementations. Additionally, new implementations can use this work as a guideline for secure design.

More work is needed to evaluate AT command security and devise solutions limiting the AT command attack surface. In particular, a large effort is required within the industry to properly secure the SIM authentication routines. Without proper protection, malware will continue to be able to exploit that functionality and use it for illicit purposes such as unwanted phone calls. Additional work could also concentrate on exploring the functionality and security SIM PIN and Radio calibration data, as well as regulations on phone power levels and such.

The authors are not aware of any existing scholarly articles in this space that perform a technical review of modem security.

Chapter 6

Related work

Zhang and Fang [30] describe a redirection attack against AKA. Since the mutual authentication phase in AKA only authenticates the user to the service provider and vice-versa, any route between the phone and the service provider is valid. This allows a man-in-the-middle attacker to relay the encrypted traffic to another network, which might cause the user to appear to be roaming (perhaps incurring roaming charges) while he is in fact in range of his home network. The authors propose a revised protocol that includes the identity of the base station in the authentication phase.

Meyer and Wetzel [42] describe an attack on AKA that works in a mixed 2G/3G environment. In this environment, the mutual authentication succeeds, but there is no integrity protection of the subsequent channel. This allows a man-in-the-middle attacker to request an insecure cipher mode with all its consequences.

Three reviews by Keromytis [43–45] survey the VoIP security research space, with a focus on SIP. In [43] six types of VoIP threats defined by the VoIP Security Alliance (VoIPSA) are described. The threats are shown in Table 6.1, including the categorization of over 50 works by Keromytis. Many papers address multiple categories. Social threats include misrepresentation of identity, social engineering, and Spam Over Internet Telephony (SPIT). Interruption of

Table 6.1: Works in VoIP security research, as categorized in [43].

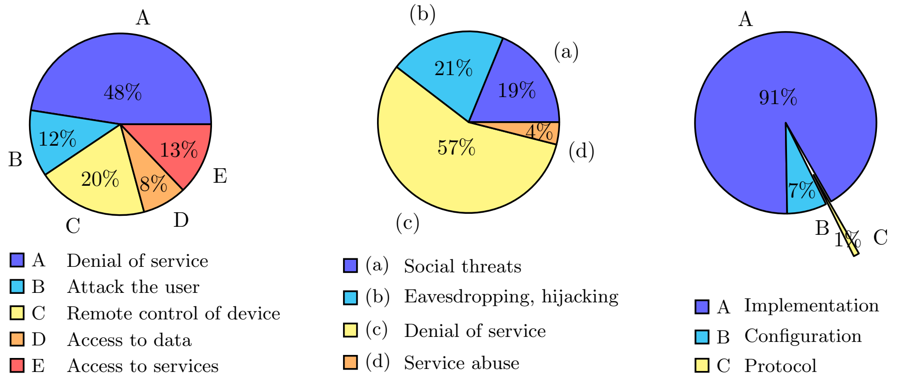
Threat category		Number of papers
Overview		36
(a)	Social threats	49
(b)	Eavesdropping, interception, and modification	34
(c)	Denial of service	19
(d)	Service abuse	8
(e)	Physical access	*out of scope*
(f)	Interruption of services	*out of scope*
Cross-category		51

services is different from denial of service in that it describes non-intentional interruptions, such as changing environmental conditions and over-subscription. Threats (e) and (f) are generally considered outside the scope of computer security research. It is not clear in which category traffic analysis (inference) would be.

In [44], over 200 known and/or disclosed security vulnerabilities are surveyed, almost all are mentioned in the Common Vulnerabilities and Exposures (CVE) database. Figure 6.1 shows the classification of these vulnerabilities by threat (as defined in [43]), effect, and cause. Of all the vulnerabilities, 3 (1%) are due to protocol issues. Those attacks are possible because the SIP specification does not explicitly require the URI part of the Digest Authentication to be the same as the the actual URI used in the request, which enables the relaying of credentials between SIP sessions.

In [45], the authors tried to identify all VoIP security research papers. This led to 245 publications forming a closed cross-citation set being surveyed. The same subset of four of the six VoIPSA threats is used (n=111), plus an additional eight categories to cover overviews and other miscellaneous work (n=134). The authors find that more research is necessary in the areas of denial of service, service abuse, cross-protocol and cross-implementation problems, configuration management, and implementation errors.

Generic Access Network (GAN) [46], also called Unlicensed Mobile Access (UMA), extends cellular communication into the Internet. It replaces the transport and lower layers of the regular 2G/3G system with their Internet Protocol suite equivalents. Using this instead of a hybrid 3G/IMS solution has the advantage of switching service between cellular networks and the Internet uninterrupted as well the benefit of a single security architecture. Grech and Eronen [47] give a high-level overview of the protocol and describe possible



(a) (b) (c)

Figure 6.1: VoIP vulnerabilities categorized [44] by (a) effect, (b) threat, and (c) cause.

attacks and solutions. One issue is that while the use of IPsec is required, use of the NULL encryption option is allowed in some cases.

Golde et al. [48] analyze the security of femtocells (low-power base stations for home use) that use the GAN protocol to connect to their home network. These femtocells maintain two separate connections. One is with the phone and the other with the home network. These connections use unrelated keying material, which means that the cell decrypts and then re-encrypts all data that it is forwarding between the two connections. This allows an attacker with 'root' access to the femtocell to perform a man-in-the-middle attack, similar to the attack described in Chapter 3.

Georgiev et al. [49] show that while browsers are currently quite good at dealing with TLS certificate validation, other software (e.g., Amazon's EC2 Java library, PayPal SDKs) that uses TLS often has implementation errors or simply does not do any certificate checks. Similarly, Falh et al. [28] show that many Android applications (8% of 13,500 tested) do not properly validate TLS certificates. The authors identify several commonly vulnerable *TrustManager* and *SocketFactory* components. They also found other issues such as a lack of visual feedback to the user.

Schrittwieser et al. [50] analyze the security of nine smartphone messaging and VoIP applications. They found that six of those applications use insecure authentication protocols that allow an attacker to impersonate users, enumerate subscribers, or spoof sender-IDs.

Chapter 7

Conclusion

In this work we describe several attacks related to VoIP. Three of these attacks can be categorized as implementation errors, while one is a cross-protocol issue. Both of these fields have been identified as needing more work to avoid security holes [45]. The man-in-the-middle attack described in Chapter 3 is rather straightforward, and ideally should have been caught during development. The false base station attack on AKA/SIP in Section 4.1 is more subtle. The malware and imposter attacks in Sections 4.2 and 4.3 are possible because of improperly secured authentication functions in the Android API, both in a vendor-customized version and in core Android.

For each attack we provide an implementation solution that eliminates the vulnerability. We have worked with T-Mobile to fix the errors in their TLS validation, and their security team has pushed an update which we have verified stops the attack. We are in contact with vendors to address our attacks against AKA on Android.

We must, however, reiterate that a stronger solution to the false base station attack on AKA/SIP can only come from protocol updates enforcing secure operation. A step in the right direction would be to abandon AKAv1 [18] and use the AKAv2 protocol [19].

Additionally, we present a general security evaluation of a typical mobile phone platform, identifying 7 security objectives for and 10 threats to such platforms. We make recommendations for security improvements regarding SIM authentication and AT commands.

Appendix A

Wi-Fi calling conversations

A.1 Initial DNS conversation from Wi-Fi calling client

```
> NAPTR wifi.msg.pc.t-mobile.com
< NAPTR 100 10 S SIPS+D2T _sips._tcp.sba.sip.t-mobile.com

> SRV _sips._tcp.sba.sip.t-mobile.com
< SRV 10 10 5061 sba.sipgeo.t-mobile.com

> A sba.sipgeo.t-mobile.com
< A <IP>

> PTR <reverse(IP)>.in-addr.arpa
< PTR m<hex(IP)>.tmodns.net
```

A.2 Initial REGISTER message from Wi-Fi calling client

```
REGISTER sip:msg.pc.t-mobile.com SIP/2.0
To: <sip:src-phone#@msg.pc.t-mobile.com>
Max-Forwards: 70
Supported: 100rel, eventlist
User-Agent: T-mobile TAS
P-Last-Access-Network-Info: 3GPP-UTRAN-TDD; utran-cell-id-3gpp=ID#1
P-Access-Network-Info: IEEE-802.11; i-wlan-node-id=ID#2
P-Preferred-Identity: sip:src-phone#@msg.pc.t-mobile.com
Call-ID: UUID#1@src-IP
From: <sip:src-phone#@msg.pc.t-mobile.com>;tag=UUID#2
CSeq: 1 REGISTER
Content-Length: 0
Via: SIP/2.0/TLS src-IP:src-port;branch=UUID#3
Contact: <sip:src-phone#@src-IP:src-port;transport=TLS>;expires=3600;
```

```
    reg-id=1;+sip.instance="<urn:gsma:imei:IMEI;svn=00>";
    +g.3gpp.icsi-ref="urn%3Aurn-7%3A3gpp-service.ims.icsi.mmtel";
    +g.3gpp.smsip;
    +g.3gpp.icsi-ref="urn%3Aurn-7%3A3gpp-service.ims.icsi.e-location"
Authorization: Digest username="IMSI@msg.pc.t-mobile.com",
    realm="msg.pc.t-mobile.com", nonce="",
    uri="sip:msg.pc.t-mobile.com", response="", algorithm=AKAv1-MD5
Privacy: none
```

A.3 INVITE message for call originating from Wi-Fi calling client

```
INVITE sip:dest-phone#@msg.pc.t-mobile.com;user=phone SIP/2.0
To: <sip:dest-phone#@msg.pc.t-mobile.com>
Max-Forwards: 70
Supported: 100rel
User-Agent: T-mobile TAS
P-Early-Media: supported
P-Last-Access-Network-Info: 3GPP-UTRAN-TDD; utran-cell-id-3gpp=ID#1
P-Access-Network-Info: IEEE-802.11; i-wlan-node-id=ID#2
Call-ID: UUID#1@src-IP
From: <sip:src-phone#@msg.pc.t-mobile.com>;tag=UUID#2
CSeq: 1 INVITE
Content-Length: size
Via: SIP/2.0/TLS src-IP:src-port;branch=UUID#3
Contact: <sip:src-phone#@src-IP:src-port;transport=TLS>;
    +g.3gpp.icsi-ref="urn%3Aurn-7%3A3gpp-service.ims.icsi.mmtel";
    +sip.instance="<urn:gsma:imei:IMEI;svn=00>"
Allow: INVITE, CANCEL, ACK, OPTIONS, BYE, PRACK, UPDATE, NOTIFY,
    REFER, MESSAGE
Accept-Contact: *;
    +g.3gpp.icsi_ref="urn%3Aurn-7%3A3gpp-service.ims.icsi.mmtel";
    +g.3gpp.smsip
Content-Type: application/sdp

v=0
o=- session-ID 1 IN IP4 src-IP
s=-
t=0 0
m=audio src-port RTP/SAVP 0
i=StreamMediaAudio
c=IN IP4 src-IP
a=crypto:1 AES_CM_128_HMAC_SHA1_32 inline:b64-enc'd key (16B), salt (14B)
a=rtpmap:0 PCMU/8000
a=sendrecv
```

Bibliography

[1] Jethro G. Beekman and Christopher Thompson. Man-in-the-middle attack on T-Mobile Wi-Fi Calling. Technical Report UCB/EECS-2013-18, EECS Department, University of California, Berkeley, Mar 2013. URL http://www.eecs.berkeley.edu/Pubs/TechRpts/2013/EECS-2013-18.html.

[2] Jethro G. Beekman and Christopher Thompson. Breaking cell phone authentication: Vulnerabilities in AKA, IMS, and Android. In *7th USENIX Workshop on Offensive Technologies*, 2013. URL https://www.usenix.org/conference/woot13/workshop-program/presentation/Beekman.

[3] Daniel Berninger. Proposal: HD Internetworking Committee. http://vcxc.org/documents/HDICRev2.3.pdf, 2011.

[4] M. Toorani and A. Beheshti. Solutions to the GSM security weaknesses. In *Proceedings of the 2nd International Conference on Next Generation Mobile Applications, Services and Technologies*, pages 576–581, September 2008. doi: 10.1109/NGMAST.2008.88.

[5] SANS Institute. The GSM standard (an overview of its security). https://www.sans.org/reading_room/whitepapers/telephone/gsm-standard-an-overview-security_317, 2001.

[6] 3rd Generation Partnership Project. 3GPP specifications. URL http://www.3gpp.org/specifications.

[7] 3rd Generation Partnership Project. 3G security; security architecture. 3GPP TS 33.102 version 11.4.0, September 2012. URL http://www.3gpp.org/ftp/Specs/html-info/33102.htm.

[8] 3rd Generation Partnership Project. 3G security; specification of the MILENAGE algorithm set: An example algorithm set for the 3GPP authentication and key generation functions f1, f1*, f2, f3, f4, f5 and f5*; document 1: General. 3GPP TS 35.205 version 11.0.0, September 2012. URL http://www.3gpp.org/ftp/Specs/html-info/35205.htm.

[9] 3rd Generation Partnership Project. AT command set for user equipment (UE). 3GPP TS 27.007 version 12.4.0, March 2014. URL http://www.3gpp.org/ftp/Specs/html-info/27007.htm.

[10] 3rd Generation Partnership Project. IP multimedia subsystem (IMS); stage 2. 3GPP TS 23.228 version 11.6.0, September 2012. URL `http://www.3gpp.org/ftp/Specs/html-info/23228.htm`.

[11] SIP: Session Initiation Protocol. IETF RFC 3261, June 2002. URL `https://tools.ietf.org/html/rfc3261`.

[12] T. Zourzouvillys and E. Rescorla. An introduction to standards-based VoIP: SIP, RTP, and friends. *IEEE Internet Computing*, 14(2):69 –73, March/April 2010. ISSN 1089-7801. doi: 10.1109/MIC.2010.31.

[13] SDP: Session Description Protocol. IETF RFC 4566, July 2006. URL `https://tools.ietf.org/html/rfc4566`.

[14] RTP: A transport protocol for real-time applications. IETF RFC 3550, July 2003. URL `https://tools.ietf.org/html/rfc3550`.

[15] Secure/Multipurpose Internet Mail Extensions (S/MIME) version 3.2 message specification. IETF RFC 5751, January 2010. URL `https://tools.ietf.org/html/rfc5751`.

[16] The Secure Real-time Transport Protocol (SRTP). IETF RFC 3711, March 2004. URL `https://tools.ietf.org/html/rfc3711`.

[17] HTTP authentication: Basic and digest access authentication. IETF RFC 2617, June 1999. URL `https://tools.ietf.org/html/rfc2617`.

[18] Hypertext Transfer Protocol (HTTP) digest authentication using Authentication and Key Agreement (AKA). IETF RFC 3310, September 2002. URL `https://tools.ietf.org/html/rfc3310`.

[19] Hypertext Transfer Protocol (HTTP) digest authentication using Authentication and Key Agreement (AKA) version-2. IETF RFC 4169, November 2005. URL `https://tools.ietf.org/html/rfc4169`.

[20] 3rd Generation Partnership Project. 3G security; access security for ip-based services. 3GPP TS 33.203 version 12.1.0, September 2012. URL `http://www.3gpp.org/ftp/Specs/html-info/35205.htm`.

[21] M. Marlinspike. New tricks for defeating SSL in practice. *BlackHat DC*, February 2009.

[22] M. Marlinspike. More tricks for defeating SSL in practice. *Blackhat USA*, July 2009.

[23] Raúl Siles. Real World ARP Spoofing. SANS, `http://pen-testing.sans.org/resources/papers/gcih/real-world-arp-spoofing-105411`, 2003.

[24] Steven M. Bellovin. Using the domain name system for system break-ins. In *Proceedings of the 5th conference on USENIX UNIX Security Symposium*, 1995.

[25] D. A. Dai Zovi and S. A. Macaulay. Attacking automatic wireless network selection. In *Proceedings from the 6th Annual IEEE SMC Information Assurance Workshop*, pages 365–372, June 2005. doi: 10.1109/IAW.2005.1495975.

[26] M. Marlinspike. sslsniff. `http://www.thoughtcrime.org/software/sslsniff/`.

[27] T-Mobile USA, Inc. The IMS project for Android. `https://code.google.com/p/the-ims-open-source-project-for-android/`.

[28] Sascha Fahl, Marian Harbach, Thomas Muders, Lars Baumgärtner, Bernd Freisleben, and Matthew Smith. Why Eve and Mallory love Android: an analysis of Android SSL (in)security. In *Proceedings of the ACM conference on Computer and communications security*, pages 50–61, 2012. ISBN 978-1-4503-1651-4. doi: 10.1145/2382196.2382205.

[29] Cisco Systems, Inc. T-Mobile Wi-Fi Calling for Business with Cisco Unified Wireless Network. Cisco Partner Solution Profile, 2010. URL `http://www.cisco.com/en/US/solutions/collateral/ns340/ns394/ns348/ns828/c22-638810-00_tMob_sBrief.pdf`.

[30] Muxiang Zhang and Yuguang Fang. Security analysis and enhancements of 3GPP authentication and key agreement protocol. *IEEE Transactions on Wireless Communications*, 4(2):734–742, March 2005. ISSN 1536-1276. doi: 10.1109/TWC.2004.842941.

[31] John Kelsey, Bruce Schneier, and David Wagner. Protocol interactions and the chosen protocol attack. In *Proceedings of the 5th International Workshop on Security Protocols*, Apr 1997.

[32] 3rd Generation Partnership Project. 3G security; security threats and requirements. 3GPP TS 21.133 version 4.1.0, January 2002. URL `http://www.3gpp.org/ftp/Specs/html-info/21133.htm`.

[33] Adrienne Porter Felt, Kate Greenwood, and David Wagner. The effectiveness of application permissions. In *Proceedings of the 2nd USENIX conference on Web application development*, pages 75–86, 2011.

[34] Pre-shared key ciphersuites for transport layer security (TLS). IETF RFC 4279, December 2005. URL `https://tools.ietf.org/html/rfc4279`.

[35] Karsten Nohl. Rooting SIM cards. In *Black Hat USA 2013*, July 2013. URL `https://www.blackhat.com/us-13/archives.html#Nohl`.

[36] Extensible Authentication Protocol Method for Global System for Mobile Communications (GSM) Subscriber Identity Modules (EAP-SIM). IETF RFC 4186, January 2006. URL `https://tools.ietf.org/html/rfc4186`.

[37] Extensible Authentication Protocol Method for 3rd Generation Authentication and Key Agreement (EAP-AKA). IETF RFC 4187, January 2006. URL `https://tools.ietf.org/html/rfc4187`.

[38] 3rd Generation Partnership Project. Personalisation of mobile equipment (ME); mobile functionality specification. 3GPP TS 22.022 version 11.0.0, September 2012. URL `http://www.3gpp.org/ftp/Specs/html-info/22022.htm`.

[39] 3rd Generation Partnership Project. Service aspects; service principles. 3GPP TS 22.101 version 13.2.0, March 2014. URL `http://www.3gpp.org/ftp/Specs/html-info/22101.htm`.

[40] 3rd Generation Partnership Project. 3GPP terminal identity security: levels, requirements and mechanisms. 3GPP TSG SA WG3 Meeting #9 S3-99510, December 1999. URL `http://www.3gpp.org/ftp/tsg_sa/WG3_Security/TSGS3_09/docs/s3-99510-3GPP%20-IMEI%20Security.doc`.

[41] 3rd Generation Partnership Project. International mobile station equipment identities (IMEI). 3GPP TS 22.016 version 11.0.0, September 2012. URL `http://www.3gpp.org/ftp/Specs/html-info/22016.htm`.

[42] Ulrike Meyer and Susanne Wetzel. A man-in-the-middle attack on UMTS. In *Proceedings of the 3rd ACM workshop on Wireless security*, pages 90–97, 2004. ISBN 1-58113-925-X. doi: 10.1145/1023646.1023662.

[43] Angelos D. Keromytis. A survey of Voice over IP security research. In *Proceedings of the 5th International Conference on Information Systems Security*, pages 1–17, 2009. ISBN 978-3-642-10771-9. doi: 10.1007/978-3-642-10772-6_1.

[44] Angelos D. Keromytis. Voice over IP: Risks, threats and vulnerabilities. In *Proceedings of the Cyber Infrastructure Protection (CIP) Conference*, June 2009.

[45] Angelos D. Keromytis. A comprehensive survey of Voice over IP security research. *IEEE Communications Surveys and Tutorials*, 14(2):514–537, 2nd quarter 2012. ISSN 1553-877X. doi: 10.1109/SURV.2011.031611.00112.

[46] 3rd Generation Partnership Project. Generic Access Network (GAN); stage 2. 3GPP TS 43.318 version 11.0.0, September 2012. URL `http://www.3gpp.org/ftp/Specs/html-info/43318.htm`.

[47] S. Grech and P. Eronen. Implications of Unlicensed Mobile Access (UMA) for GSM security. In *Proceedings of the 1st International Conference on Security and Privacy for Emerging Areas in Communications Networks*, pages 3–12, September 2005. doi: 10.1109/SECURECOMM.2005.23.

[48] Nico Golde, Kévin Redon, and Ravishankar Borgaonkar. Weaponizing femtocells: The effect of rogue devices on mobile telecommunications. In *Annual Network & Distributed System Security Symposium*, Feb 2012.

[49] Martin Georgiev, Subodh Iyengar, Suman Jana, Rishita Anubhai, Dan Boneh, and Vitaly Shmatikov. The most dangerous code in the world: validating SSL certificates in non-browser software. In *Proceedings of the ACM conference on Computer and communications security*, pages 38–49, 2012. ISBN 978-1-4503-1651-4. doi: 10.1145/2382196.2382204.

[50] S. Schrittwieser, P. Frühwirt, P. Kieseberg, M. Leithner, M. Mulazzani, M. Huber, and E. Weippl. Guess who's texting you? evaluating the security of smartphone messaging applications. In *Proceedings of the 19th Annual Symposium on Network and Distributed System Security*, February 2012.