Breaking Cell Phone Authentication: Vulnerabilities in AKA, IMS and Android

Jethro G. Beekman and Christopher Thompson
Electrical Engineering and Computer Science
University of California, Berkeley
jbeekman@eecs.berkeley.edu, cthompson@cs.berkeley.edu

Abstract

Next generation IP telephony such as the IP Multimedia Subsystem (IMS) framework has been used to create Internet calling services which let cellular users make and receive calls even when without cellular reception. In this paper, we look at the security aspects of Internet calling services and other systems that use the 3GPP Authentication and Key Agreement (AKA) protocol for authentication, particularly focusing on the context of cellular authentication in Android. We describe a new man-in-the-middle attack on T-Mobile's Wi-Fi Calling service, which is installed on millions of T-Mobile Android smartphones. We also describe three new attacks on AKA in the context of Internet calling and Android. We have worked with T-Mobile to fix the man-in-the-middle vulnerability, and we present clear and actionable solutions to fix the remaining vulnerabilities.

1 Introduction

Many telecommunication companies are moving more and more services to Internet-based platforms, citing flexibility, cost savings and evolvability [19]. 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 paper, we analyze the security of these IP telephony systems, especially the SIP and IMS protocols. Our analysis found a security vulnerability in the Wi-Fi Calling service provided by T-Mobile, which is based on the IP Multimedia Subsystem (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 such as Rogers Wireless, Orange Telecom, Republic Wireless, and Cincinnati Bell (among others). We also identify 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 would prevent each attack.

We have worked with T-Mobile to fix one of the vulnerabilities we present. We are working with vendors to address the remaining vulnerabilities.

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 [39]. There has been significant research analyzing the security of cellular

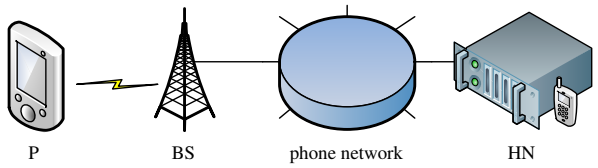

Figure 1: A typical cellular network setup. (P) Phone, (BS) Base station, (HN) Home network. The base station need not be operated by the home network, as long as there is some communication channel between the two.

communication, however it mostly applies to 2G protocols [35]. The 3rd Generation Partnership Project (3GPP) has sought to fix many of these problems, and the specifications are publicly available [10]. 3GPP defines the Authentication and Key Agreement (AKA) Protocol [13], 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 [14].

Figure 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 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 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 IP Multimedia Subsystem

IP Multimedia Subsystem (IMS) [16] is a framework for delivering 'multimedia' services over the Internet. Here, multimedia means everything related to calling and mes-

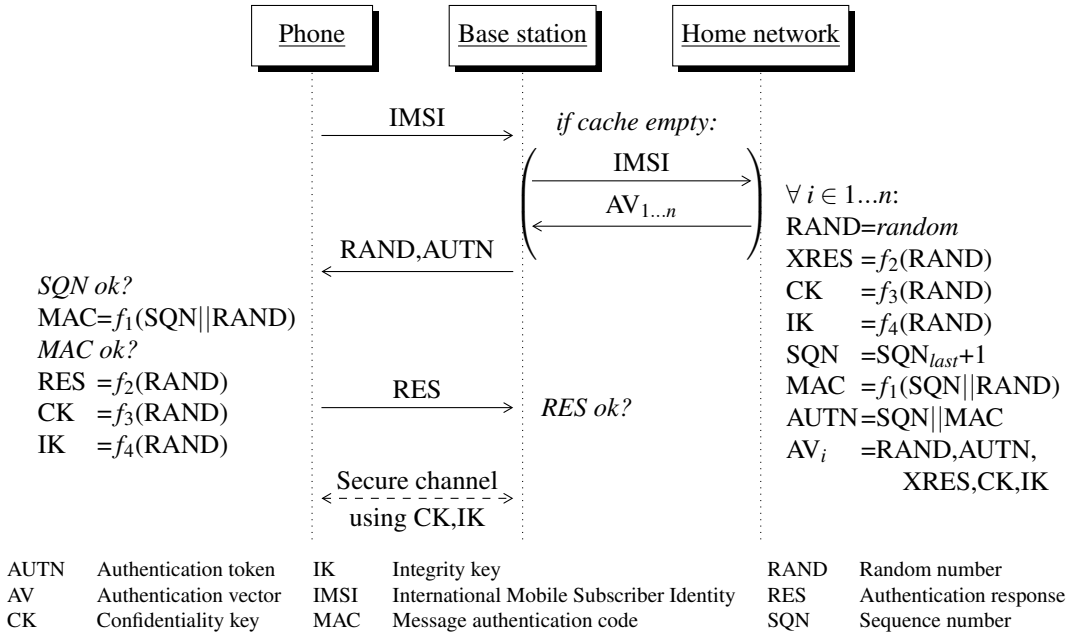

Figure 2: A simplified diagram of the 3GPP Authentication and Key Agreement Protocol. The home network keeps track of the last generated sequence number. The phone keeps track of the last used sequence number. In reality, the sequence number might also be encrypted with the anonymity key AK (not shown). 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.

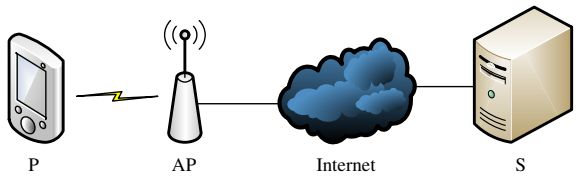

Figure 3: A typical IMS setup. (P) Phone, (AP) Wi-Fi access point, (S) SIP server. The SIP server is part of the phone's home network.

saging 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 3.

The Session Initiation Protocol (SIP) [3], 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 [41] 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) [8]. 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) [4].

SIP over TCP or UDP is vulnerable to man-in-the-middle attacks. The SIP messages can be encrypted using S/MIME [9] to ensure integrity and 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 [5].

2.2.1 SIP authentication

Authentication in SIP uses the same mechanisms as HTTP, with the WWW-Authenticate and Authorization headers. Unlike most HTTP configurations, Digest authentication [1] 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 [2] 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 [6] 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 [12].

2.3 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 [32,33] identify a few potential problems. Most of these issues have to do with authentication, or more technically, 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 [37] 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 [18], 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 [21], in which they imitate a legitimate network and trick users into connecting to that network instead.

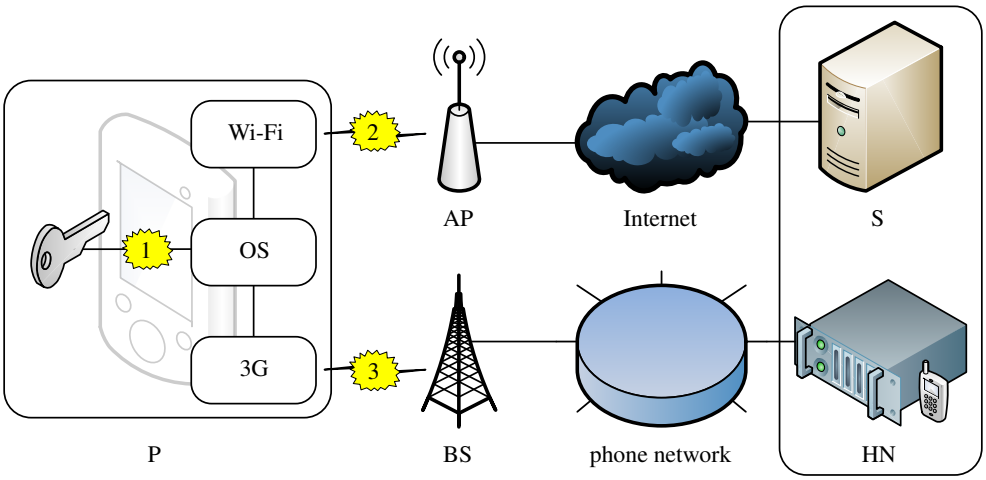

Figure 4: How a phone connects to the IMS and cellular systems from Figures 1 and 3. The SIP server, S, and Home Network, HN, are operated by the same entity. The authentication data channels 1, 2 and 3 indicate where various authentication data is transferred between principals.

3 Attack vectors

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 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 $H(\text{RES})$ or $H(\text{RES,CK,IK,“AKAv2”})$ 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 [27]. 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 5.1 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 1 enumerates the ways in which the different channels could be connected by an attacker and we analyze these ways in the following sections.

MAN-IN-THE-MIDDLE ATTACKS POSSIBLE

Phone side	Network side		
	SIP/AKAv1	*SIP/AKAv2*	*3G AKA*
Smart card	Y, §5.2	Y, §5.2	Y, §5.3
SIP/AKAv1	Y, §4	$N^{H,K}$	$N^{H,K}$
SIP/AKAv2	N^{H}	Y, §4	N^{H}
3G AKA	Y, §5.1	N^{K}	Y, [40], §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.

Table 1: Each cell indicates whether an attack is possible when connecting the mechanism used on the phone on the left side to the mechanism used int 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.

4 Man-in-the-middle attack on T-Mobile Wi-Fi Calling

T-Mobile's Wi-Fi Calling system is a large-scale deployed IMS system, which gave us a testbed to investigate these attack vectors. In this section, we look at how a network attacker on the second authentication data channel (between the phone and the SIP server) can perform a man-in-the-middle attack.

When connecting to Wi-Fi Calling, a DNS conversation takes place requesting a chain of information (including NAPTR and SRV records) about *wifi.msg.pc.t-mobile.com*. Then, a TLS connection is established to the host and port returned by DNS (*sba.sipgeo.t-mobile.com* and *5061*). The certificate chain returned by T-Mobile's server was somewhat non-standard. Two things stood out: first, the common name of the first certificate was simply the IP address of the server; second, the self-signed root certificate was not included in standard Certificate Authority (CA) distributions. Analysis of the Wi-Fi Calling binaries indeed showed no trace of the root certificate. This led us to believe that the TLS certificate was not being correctly validated, and indeed the client did not have any problems with sslsniff [31] intercepting the connection.

As hinted at by the DNS records and the port number, a SIP [3] dialog is initiated when the TLS connection is established. AKAv1 Digest authentication is used to secure the connection.

Because of the lack of proper validation, an attacker can man-in-the-middle the Wi-Fi Calling TLS connection and then eavesdrop or modify the SIP traffic that follows. Figure 5 shows a user's phone P connected to a wireless access point (AP) controlled by an attacker M. The attacker proxies the TLS traffic between P and the SIP server S using a self-signed certificate with a tool such as sslsniff, allowing the attacker to read and modify all traffic sent in the TLS session.

The attacker M can now continue to proxy the resulting decrypted SIP dialog between P and S, and could record all incoming and outgoing calls and text messages (collectively "SIP traffic"), and also block or reroute SIP traffic. The attacker could modify traffic by faking a sender or changing the real-time voice data or message content. The attacker could also fake incoming traffic or impersonate a client with forged outgoing traffic.

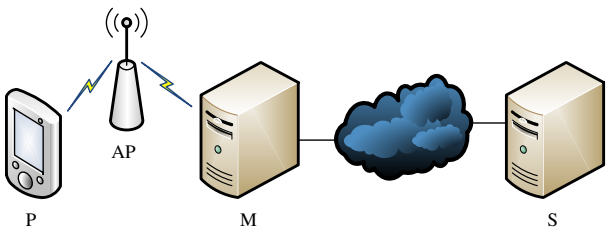

Figure 5: Wireless man-in-the-middle setup. (P) Phone. (AP) Access point. (M) Man-in-the-middle. (S) SIP server.

We verified the ability to record outgoing calls and incoming and outgoing text messages. We also verified changing the destination phone number on outgoing calls by modifying sslsniff to replace a single target phone number with a different one.

This attack would have allowed an attacker to eavesdrop on all of a user's voice calls and text messages sent over the Wi-Fi Calling service. An attacker could even reroute messages and phone calls (or create new ones) to premium numbers, which would cost the user money and could earn money for the attacker.

4.1 Scope

This attack affected all T-Mobile Wi-Fi Calling users that had the newer IMS-based application before December 2012—potentially millions of users. Not all versions of T-Mobile Wi-Fi calling were necessarily vulnerable to this attack. According to T-Mobile's website, the IMS stack is used on the Samsung Galaxy S II, HTC Amaze 4G, myTouch and myTouch Q. We have tested the attack on a Samsung Galaxy S Relay 4G and a Samsung Galaxy Note 2. Users of T-Mobile Wi-Fi calling for Business might not be vulnerable to this specific vulnerability, since it uses GAN, not IMS technology [20].

4.2 Solution

T-Mobile has open-sourced most of its Android IMS stack [38]. Using the code and by reverse engineering the binaries from a Wi-Fi Calling enabled phone, we were able to identify the vulnerable TLS validation code [17].

In December 2012, we notified T-Mobile of these vulnerabilities. In subsequent months, they added proper certificate validation to the T-Mobile Wi-Fi Calling app, 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.

We note that using AKAv2 would not have protected against this attack.

5 Attacks on the AKA Protocol

In this section we show three attacks on AKA, which are caused by poor cross-protocol interaction and implemen-

tation issues. These attacks are based on the poor use and handling of the AKA session keys.

5.1 False base station attack

We discovered that AKA is secure if and only if the confidentiality and integrity keys produced by AKA will subsequently be used to protect the communication channel. The Digest Authentication with AKA RFC [2] 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* [11]. In this attack, the attacker controls the false base station F and an Internet-connected host M, as shown in Figure 7. Now consider the sequence of events in Figure 6. 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

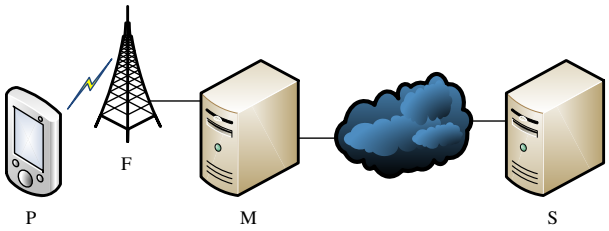

Figure 7: Setup for the false base station attack. (P) Phone. (F) False base station. (M) (On-path) attacker. (S) SIP server.

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.

5.2 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 al-

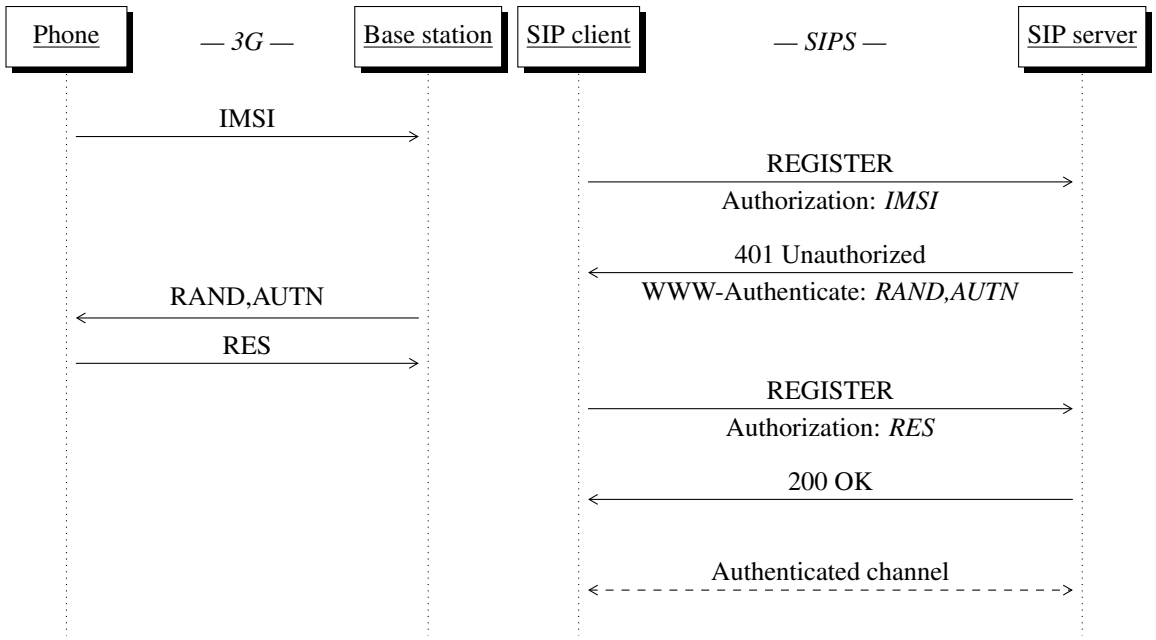

Figure 6: False base station attack, exploiting interactions between 3G and Digest Authentication AKA. Both the base station and the SIP client are under the attacker's control. In reality, the Authorization and WWW-Authenticate headers are much more verbose.

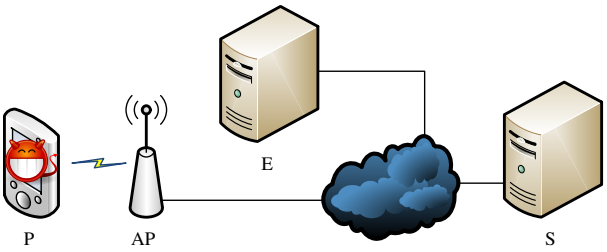

Figure 8: Setup for the malware attack. (P) Phone with malware installed. (AP) Access point. (E) (Off-path) attacker. (S) SIP server. In this figure, the phone is connected to the Internet via Wi-Fi, but it could also be connected via a 3G data connection instead.

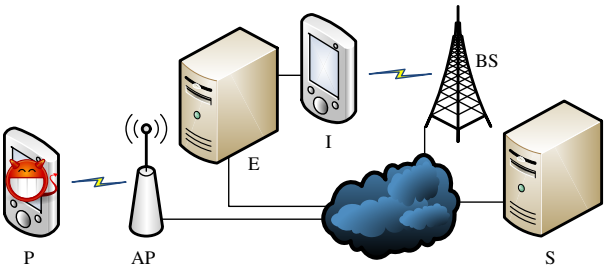

Figure 9: Setup for the imposter attack. (P) Phone with malware installed. (AP) Access point as in Figure 8. (E) (Off-path) attacker. (I) Imposter mobile device. (BS) Legitimate base station and (S) Service provider as in Figure 1.

lows 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 [23].

In this attack, as shown in Figure 8, 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 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.

5.3 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 9). 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.

5.4 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]

5.5 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 [7] 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

[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. 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.

keys, for example by replacing the phrase "they may be used" with "they MUST be used". 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. 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.

6 Related work

Zhang and Fang [40] 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 [34] 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 survey the VoIP security research space, with a focus on SIP. In the first review [28], six types of VoIP threats defined by the VoIP Security Alliance (VoIPSA) are described. The threats are: Social threats; Eavesdropping, interception, and modification; Denial of service; Service abuse; Physical access; and Interruption of services. Keromytis categorized over 50 papers, many of which addressed multiple categories. In the second review [29], over 200 known and/or disclosed security vulnerabilities are surveyed—almost all are mentioned in the Common Vulnerabilities and Exposures (CVE) database. 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 the third review [30], Keromytis tried to identify all VoIP security research papers. This work gathered 245 publications forming a closed cross-citation set being surveyed. Keromytis found 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) [15], 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. Grech and Eronen [26] give a high-level overview of the protocol and describe possible 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. [25] 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 Section 4.

Georgiev et al. [24] 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. [22] 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. [36] 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.

7 Conclusion

In this work we describe several attacks related to VoIP, three of which can be categorized as implementation errors, while one is a cross-protocol issue. The man-in-the-middle attack described in Section 4 is rather straightforward, and ideally should have been caught during development. The false base station attack on AKA/SIP in Section 5.1 is more subtle. The malware and imposter attacks in Sections 5.2 and 5.3 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 [2] and use the AKAv2 protocol [6].

References

[1] HTTP authentication: Basic and digest access authentication. IETF RFC 2617, June 1999.

[2] Hypertext Transfer Protocol (HTTP) digest authentication using Authentication and Key Agreement (AKA). IETF RFC 3310, September 2002.

[3] SIP: Session Initiation Protocol. IETF RFC 3261, June 2002.

[4] RTP: A transport protocol for real-time applications. IETF RFC 3550, July 2003.

[5] The Secure Real-time Transport Protocol (SRTP). IETF RFC 3711, March 2004.

[6] Hypertext Transfer Protocol (HTTP) digest authentication using Authentication and Key Agreement (AKA) version-2. IETF RFC 4169, November 2005.

[7] Pre-shared key ciphersuites for transport layer security (TLS). IETF RFC 4279, December 2005.

[8] SDP: Session Description Protocol. IETF RFC 4566, July 2006.

[9] Secure/Multipurpose Internet Mail Extensions (S/MIME) version 3.2 message specification. IETF RFC 5751, January 2010.

[10] 3RD GENERATION PARTNERSHIP PROJECT. 3GPP specifications.

[11] 3RD GENERATION PARTNERSHIP PROJECT. 3G security; security threats and requirements. 3GPP TS 21.133 version 4.1.0, January 2002.

[12] 3RD GENERATION PARTNERSHIP PROJECT. 3G security; access security for ip-based services. 3GPP TS 33.203 version 12.1.0, September 2012.

[13] 3RD GENERATION PARTNERSHIP PROJECT. 3G security; security architecture. 3GPP TS 33.102 version 11.4.0, September 2012.

[14] 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.

[15] 3RD GENERATION PARTNERSHIP PROJECT. Generic Access Network (GAN); stage 2. 3GPP TS 43.318 version 11.0.0, September 2012.

[16] 3RD GENERATION PARTNERSHIP PROJECT. IP multimedia subsystem (IMS); stage 2. 3GPP TS 23.228 version 11.6.0, September 2012.

[17] BEEKMAN, J. G., AND THOMPSON, C. Man-in-the-middle attack on T-Mobile Wi-Fi Calling. Tech. Rep. UCB/EECS-2013-18, EECS Department, University of California, Berkeley, Mar 2013.

[18] BELLOVIN, S. M. Using the domain name system for system break-ins. In *Proceedings of the 5th conference on USENIX UNIX Security Symposium* (1995).

[19] BERNINGER, D. Proposal: HD Internetworking Committee. `http://vcxc.org/documents/HDICRev2.3.pdf`, 2011.

[20] CISCO SYSTEMS, INC. T-Mobile Wi-Fi Calling for Business with Cisco Unified Wireless Network. Cisco Partner Solution Profile, 2010.

[21] DAI ZOVI, D. A., AND MACAULAY, S. A. Attacking automatic wireless network selection. In *Proceedings from the 6th Annual IEEE SMC Information Assurance Workshop* (June 2005), pp. 365–372.

[22] FAHL, S., HARBACH, M., MUDERS, T., BAUMGÄRTNER, L., FREISLEBEN, B., AND SMITH, M. Why Eve and Mallory love Android: an analysis of Android SSL (in)security. In *Proceedings of the ACM conference on Computer and communications security* (2012), pp. 50–61.

[23] FELT, A. P., GREENWOOD, K., AND WAGNER, D. The effectiveness of application permissions. In *Proceedings of the 2nd USENIX conference on Web application development* (2011), pp. 75–86.

[24] GEORGIEV, M., IYENGAR, S., JANA, S., ANUBHAI, R., BONEH, D., AND SHMATIKOV, V. 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* (2012), pp. 38–49.

[25] GOLDE, N., REDON, K., AND BORGAONKAR, R. Weaponizing femtocells: The effect of rogue devices on mobile telecommunications. In *Annual Network & Distributed System Security Symposium* (Feb 2012).

[26] GRECH, S., AND ERONEN, P. 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* (September 2005), pp. 3–12.

[27] KELSEY, J., SCHNEIER, B., AND WAGNER, D. Protocol interactions and the chosen protocol attack. In *Proceedings of the 5th International Workshop on Security Protocols* (Apr 1997).

[28] KEROMYTIS, A. D. A survey of Voice over IP security research. In *Proceedings of the 5th International Conference on Information Systems Security* (2009), pp. 1–17.

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

[30] KEROMYTIS, A. D. A comprehensive survey of Voice over IP security research. *IEEE Communications Surveys and Tutorials 14*, 2 (2nd quarter 2012), 514–537.

[31] MARLINSPIKE, M. sslsniff. `http://www.thoughtcrime.org/software/sslsniff/`.

[32] MARLINSPIKE, M. More tricks for defeating SSL in practice. *Blackhat USA* (July 2009).

[33] MARLINSPIKE, M. New tricks for defeating SSL in practice. *BlackHat DC* (February 2009).

[34] MEYER, U., AND WETZEL, S. A man-in-the-middle attack on UMTS. In *Proceedings of the 3rd ACM workshop on Wireless security* (2004), pp. 90–97.

[35] 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.

[36] SCHRITTWIESER, S., FRHWIRT, P., KIESEBERG, P., LEITHNER, M., MULAZZANI, M., HUBER, M., AND WEIPPL, E. Guess whos texting you? evaluating the security of smartphone messaging applications. In *Proceedings of the 19th Annual Symposium on Network and Distributed System Security* (February 2012).

[37] SILES, R. Real World ARP Spoofing. SANS, http://pen-testing.sans.org/resources/papers/gcih/real-world-arp-spoofing-105411, 2003.

[38] T-MOBILE USA, INC. The IMS project for Android. https://code.google.com/p/the-ims-open-source-project-for-android/.

[39] TOORANI, M., AND BEHESHTI, A. Solutions to the GSM security weaknesses. In *Proceedings of the 2nd International Conference on Next Generation Mobile Applications, Services and Technologies* (September 2008), pp. 576–581.

[40] ZHANG, M., AND FANG, Y. Security analysis and enhancements of 3GPP authentication and key agreement protocol. *IEEE Transactions on Wireless Communications 4*, 2 (March 2005), 734–742.

[41] ZOURZOUVILLYS, T., AND RESCORLA, E. An introduction to standards-based VoIP: SIP, RTP, and friends. *IEEE Internet Computing 14*, 2 (March/April 2010), 69 –73.