Formal modeling of authentication in SIP registration

Anders Moen Hagalisletto
Norwegian Computing Center
and Department of Informatics,
University of Oslo, Norway
anders.moen@nr.no

Lars Strand
Norwegian Computing Center
and Department of Informatics,
University of Oslo, Norway
lars.strand@nr.no

Abstract

The Session Initiation Protocol (SIP) is increasingly used as a signaling protocol for administrating Voice over IP (VoIP) phone calls. SIP can be configured in several ways so that different functional and security requirements are met. Careless configuration of the SIP protocol is known to lead to a large set of attacks.

In this paper we show how SIP can be specified in a protocol centric formal language. Both static analysis and simulations can be performed on the resulting specifications by the recently developed tool PROSA. In particular, we analyze the VoIP architecture of a medium size Norwegian company, and show that several of the well known threats can be found.

1 Introduction

It was not until the late 1990s that phone calls over the Internet reached a significant number of users. The convergence of voice, video and data over the same IP infrastructure, reduces installation and maintainance cost since there is less need for separate networks. However, securing a VoIP system is challenging: Since VoIP shares the same infrastructure as traditional data networks, it also inherits the security problems of data communication. VoIP does not have a dominant standard that has been scrutinized over the years by security researchers. VoIP also has high requirements to the network with respect to Quality of Service since its a duplex communication with low tolerance for latency, jitter, and packet loss.

Many will likely expect VoIP to meet the same service level as the Public Switched Telephony System (PSTN), the traditional circuit-switched telephone networks. People have become accustomed to 99,999% availability on PSTN [10].

The Session Initiation Protocol (SIP) [12] is an application layer protocol for handling multimedia sessions. SIP is used to negotiate and establish a *context* for the media flow, where other protocols are used for the media transport. The media protocol Real-time Transport Protocol (RTP) is often used in combination with SIP. SIP is a text based protocol, similar to HTTP.

Originally the focus on the SIP specification has been on functionality for providing additional services rather than security features [13]. Security was soon recognized to be an area of further investigation and improvement [2, 4, 6].

The rest of the paper is organized as follows: In Section 2 we give an introduction to authentication in SIP. In Section 3, we present the VoIP architecture of the case study show how the formal specifications can obtained. Section 4, the formalization of the protocol is presented, including a formalization of Digest Access Authentication, and the registration protocol. In Section 5, results from the analysis is presented, including an example of a call-hijacking attack on the registration sub-protocol and a description of the application of the tool PROSA.

2 Authentication in SIP

Whether a given VoIP configuration is considered secure depends on two factors: (1) the requirements specified by the given security policy for a particular installation, and (2) whether these requirements are covered by the implemented security mechanisms. The security requirements for telephony connections depends on the application area: for some companies might connectivity be enough, while others would require strong confidentiality, integrity and authenticity.

According to the RFC3261 [12], there are three ways to configure SIP authentication: plaintext authentication, weak authentication, and strong authentication. Plaintext authentication sends the authentication credentials unprotected. Weak authentication is an adaptation of the HTTP Digest Access Authentication [7] that requires a shared secret between the two participants. Strong authentication uses certificates in the same way as web browsers and

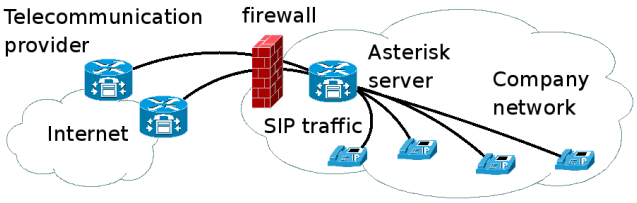

Figure 1. Scenario for VoIP architecture.

servers use them. Since it is currently common to use weak authentication, our paper discusses only Digest Access Authentication.

Digest Access Authentication method works like this: When receiving a request, the server may challenge the client with a random *nonce*. The client then hashes the nonce, secret password, username and other parameters. The server, upon receiving the hash from the client, does the same computation and compares the two results. If the server generated hash equals the one received from the client, the client is authenticated.

The digest authentication provides message authentication and replay protection only. It does not cover message integrity or confidentiality and does not provide strong authentication when compared to the Kerberos protocol or other public key based mechanism, see for instance [11].

3 Analysis

Based on the given voice over IP architecture, the documentation of SIP [12, 14], and Digest Access Authentication [7], we specified the particular instance of of SIP formally. Unfortunately, it was not possible to get a reasonable interpretation of the scenario, since the SIP RFC lacked several details of the message interactions and message content. Therefore we monitored the implementation by logging and analyzing network traffic using the network monitoring tool *Wireshark*[1].

3.1 VoIP in a medium size company

Our case study is taken from a medium sized Norwegian company with 100 employees. Most of them have a VoIP "hard phone" from Cisco. Some also have a "soft phone" installed on their computer, an application that supports VoIP phone calls. The configuration files counts 127 SIP phones, including both soft and hard phones. One VoIP server running the open source software *Asterisk*[2] on Linux provides the functionality of a Private Branch eXchange (PBX).

[1]Wireshark: Go deep. `http://www.wireshark.org/`.

[2]Asterisk: The Open Source PBX & Telephony Platform `http://www.asterisk.org/`

Phone calls destined to the outside, or incoming calls, are handled by an external telecommunication provider. The external connections use the Inter-Asterisk eXchange v2 protocol (IAX) [1]. An overview of the architecture is given in Figure 1.

The hard phones use DHCP to obtain and configure network settings. This simplifies network management. But since dynamic IP addresses are used, the phones need to register to the Asterisk server at startup time. During the registration they perform Digest authentication.

Two soft-phones were used to place the SIP calls. All network traffic between the phones and the Asterisk server was then intercepted and logged using *Wireshark*.

3.2 Method

It is well known that security protocols can be hard to specify formally [8, 3]. Based on information obtained from monitoring network traffic, a precise specification of SIP registration with Digest authentication was generated. This specification was used as input to the PROSA protocol analyzer in order to validate the specification and simulate uncompromized as well as compromised protocol instances. A severe call-hijacking attack was found eventually, which shows that formalization, simulation and automated analysis can reveal potential and real weaknesses with the implementation of VoIP systems.

We learned that the informal specifications in RFC 3261 were incomplete: Message interaction was not specified explicitly, neither was the the ordering of elements, and concrete adaptation of Digest Access Authentication.

Moreover the most important implications were that (i) formalization of complex protocols is much faster using network monitoring tools than without, and (ii) taking traces from such tools give realistic specifications that are close to the implementation.

The security analysis was performed by first obtaining a formal specification that could be studied in itself and by automated tool support. We used the protocol analyzer PROSA to specify a register session of SIP. The tool consists of a formal language based on temporal epistemic logic, a static analyzer that can automatically refine protocol specifications, and a simulator that can execute protocols as well as attacks on protocols. By using PROSA typical errors like misprints of sender/receiver names or incomplete and incorrect message contents were easily discovered. Then several simulations were configured, attack-free, eavesdropping attacks and finally a severe call-hijacking attack that breaks integrity of the clients address.

A consequence of the latter is that the attacker intercepts phone calls addressed to the client, but neither the client itself, nor the responder might know that the SIP channel is redirected and corrupted.

4 Formal specification of SIP

In this section we describe the formal specification of the SIP registration and signaling sub-protocols. First we derive specifications in high level protocol specifications in the form used in the literature. A protocol clause is of the form

$$(P) \qquad A \longrightarrow B \quad : \quad M$$

meaning "agent A sends a message M to the agent B" The messages in the protocols consist of basic entities as follows:

$A, B, C, S, I, I(A)$	agent terms
K_{AB}	symmetric key shared by A, B
N_A	nonce generated by agent A
W_A^Y	string containing the text Y related to agent A
X_A	miscellaneous entities

There are three composition operators in the notation: concatenation of message content denoted by "," (comma), hashing $\mathsf{H}[M]$, and encryption denoted by $E(K \ : \ M)$, where K denotes a key and M a message content.

4.1 Digest access authentication

Digest access authentication uses hashing, where nonces are used to protect against crypto analysis. We let $\mathsf{H}[\mathrm{C}]$, denote the hashing of content C. Digest access authentication is then given by

$$\begin{aligned} H_1 &= \mathsf{H}[W_C^{\mathrm{uname}}, W^{\mathrm{realm}}, K_{CR}^{\mathrm{pwd}}] \\ H_2 &= \mathsf{H}[W^{\mathrm{meth}}, W_C^{\mathrm{URI}}] \\ \text{response} &= \mathsf{H}[H_1, N_R, X_{\mathrm{nc}}, N_C, W^{\mathrm{qop}}, H_2] \end{aligned}$$

Written out explicitly the response yields:

$$\mathsf{H}[\mathsf{H}[W_C^{\mathrm{uname}}, W^{\mathrm{realm}}, K_{CR}^{\mathrm{pwd}}], N_R, X_{\mathrm{nc}}, N_C, W^{\mathrm{qop}}, \mathsf{H}[W^{\mathrm{meth}}, W_C^{\mathrm{URI}}]]$$

A typical application is then given by a challenger R requesting a client C to authenticate as described in the following protocol skeleton:

$$\begin{aligned} &(\mathrm{D}_1) \quad R \longrightarrow C : N_R \\ &(\mathrm{D}_2) \quad C \longrightarrow R : W_C^{\mathrm{uname}}, W^{\mathrm{realm}}, N_R, W_C^{\mathrm{URI}}, X_{\mathrm{nc}}, N_C, W^{\mathrm{qop}}, \mathsf{H}[H_1, N_R, X_{\mathrm{nc}}, N_C, W^{\mathrm{qop}}, H_2] \end{aligned}$$

Both parties, share a common secret, a password K_{CR}^{pwd}, playing the role of a symmetric key. Initially the challenger R sends a nonce N_R to the client C. The client responds by sending all the basic entities in the response in plaintext, except the password, and at the end the response itself. The challenger R can then perform hashing according to the response scheme above on the received entities and the password to check whether the response corresponds with the verification. The entities involved in digest access authentication can be explained as follows:

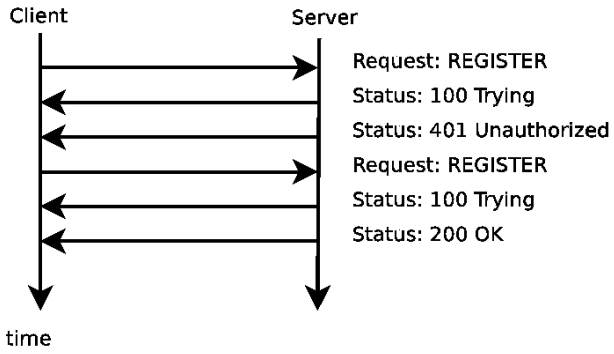

Figure 2. Digest authentication in SIP.

W_C^{uname}	authentication username of C
W^{realm}	defines a protective domain
K_{CR}^{pwd}	shared password between client C and challenger R
W^{meth}	main method of message (like HTTP)
W_C^{URI}	Digest URI for client C
N_R	nonce of the challenger R
X_{nc}	nonce counter
N_C	client C's nonce
W^{qop}	quality of protection

A "realm" is a protection domain on the server which is globally unique. Each realm on the server are partitioned into a set of logical protection spaces, each with its own authentication scheme.

In case of the SIP protocol, the URI is interpreted as the SIP URI, which have the same construction as an email address $<$ sip : user@domain $>$.

The authentication is one-way: The client C is authenticated to the challenger R. Authenticity of the client C is guaranteed by the secrecy of the shared key K_{CR}^{pwd}: Agent R can be certain that the message comes from C, since C is the only agent except C that possesses the key. Integrity of the message entities involved is provided by the fact that the hash could only be generated by C and freshness of the message is provided by the challenger nonce N_R.

4.2 The registration sub-protocol

If a SIP client is roaming or, like in our case, use DHCP to obtain network configuration, the client must register itself to a registration server. The SIP registration sub-protocol accomplish this task. A registration server should be connected to a location server that handles the bindings

between the user's URI and contact addresses. SIP registration without any security mechanisms configured is given by the following specification:

$$
\begin{array}{lllll}
(\mathrm{P}_1) & C \longrightarrow R & : & W^{\mathrm{REGISTER}}, W_C^{\mathrm{Contact}}, N_C^{\mathrm{callid}} \\
(\mathrm{P}_2) & R \longrightarrow C & : & W^{\mathrm{Trying}}, W_C^{\mathrm{Contact}}, N_C^{\mathrm{callid}} \\
(\mathrm{P}_3) & R \longrightarrow C & : & W^{\mathrm{OK}}, W_C^{\mathrm{Contact}}, N_C^{\mathrm{callid}}
\end{array}
$$

Initially in message (P_1), agent C sends a registration request to the registrar server R. Agent R gives a receipt that the registration has been received in (P_2), and then finally if C's request is accepted by R, then a notification message is replied in (P_3). The message entities involved in the protocol scheme are:

N_C^{callid}	The session identifier for the current registration session
W^{Contact}	Contact host for client
W^{REGISTER}	REGISTER method that indicate a registration session
W^{Trying}	100 Trying, receipt to a previous SIP message
W^{OK}	200 OK method notifies successful registration

The N_C^{callid} is the session identifier for the current instance of the protocol. In the realization N_C^{callid} is built up by the host address of the client C and a nonce generated by C. In the context of the registration protocol each registration session from one client to one particular registration server should use the same CALLID N_C^{callid}, in order to prevent a delayed REGISTER request to arrive out of order [12, p. 58]. The available contact hosts can be more than one (e.g. several phones or email addresses), hence several potential bindings for the client can be specified by replacing W_C^{Contact}, with a sequence of potential hosts $W^{\mathrm{Contact}} = W_C^{\mathrm{Contact}_1}, \ldots, W_C^{\mathrm{Contact}_n}$.

If digest access authentication is used, as shown in Figure 2, then the registration sub-protocol can be specified as follows:

$$
\begin{array}{ll}
(\mathrm{P}_1^{\mathrm{D}}) & C \longrightarrow R : W^{\mathrm{REGISTER}}, W_C^{\mathrm{Contact}}, N_C^{\mathrm{callid}} \\
(\mathrm{P}_2^{\mathrm{D}}) & R \longrightarrow C : W^{\mathrm{Trying}}, N_C^{\mathrm{callid}} \\
(\mathrm{P}_3^{\mathrm{D}}) & R \longrightarrow C : W^{\mathrm{Unauth}}, W^{\mathrm{auth}}, W^{\mathrm{realm}}, N_R, N_C^{\mathrm{callid}} \\
(\mathrm{P}_4^{\mathrm{D}}) & C \longrightarrow R : W^{\mathrm{REGISTER}}, N_C^{\mathrm{callid}}, W_C^{\mathrm{uname}}, W^{\mathrm{realm}}, \\
 & \qquad N_R, W_C^{\mathrm{URI}}, X_{\mathrm{nc}}, N_C, W^{\mathrm{qop}} \\
 & \qquad \mathsf{H}[\mathsf{H}[W_C^{\mathrm{uname}}, W^{\mathrm{realm}}, K_{CR}^{\mathrm{pwd}}], N_C, X_{\mathrm{nc}}, \\
 & \qquad N_R, W^{\mathrm{qop}}, \mathsf{H}[W^{\mathrm{REGISTER}}, W_C^{\mathrm{URI}}]] \\
(\mathrm{P}_5^{\mathrm{D}}) & R \longrightarrow C : W^{\mathrm{Trying}}, W_C^{\mathrm{Contact}}, N_C^{\mathrm{callid}} \\
(\mathrm{P}_6^{\mathrm{D}}) & R \longrightarrow C : W_{\mathrm{OK}}, W_C^{\mathrm{Contact}}, N_C^{\mathrm{callid}}
\end{array}
$$

In this protocol the user (or client) requests to register at a registrar server R, then R gives a receipt of this message. The challenger R then demands an authentication $(\mathrm{P}_3^{\mathrm{D}})$ by combining message (D_1) with the appropriate SIP method, information attributes and CALLID to the client. The meaning of the latter message is to request the client to authenticate itself to the registrar R. The client does this by essentially combining messages (P_1) and (D_2) in constructing message $(\mathrm{P}_4^{\mathrm{D}})$. A receipt from the registrar is given for the authentication trial in $(\mathrm{P}_5^{\mathrm{D}})$, and if the server R is able to verify message $(\mathrm{P}_4^{\mathrm{D}})$ by hashing the received plaintext elements and the previously known shared secret K_{CR}^{pwd} according to the response scheme, then the registrar notifies the client about the successful registration in the final message $(\mathrm{P}_6^{\mathrm{D}})$. Note that $(\mathrm{P}_6^{\mathrm{D}})$ is interpreted stronger than (P_3), the former means that the client has successfully registered and is authenticated to R.

In the digest registration protocol there are thus three new messages $\mathrm{P}_3^{\mathrm{D}}$, $\mathrm{P}_4^{\mathrm{D}}$, and $\mathrm{P}_5^{\mathrm{D}}$ and consequently additional four message entities. These entities are described below:

W^{Unauth}	401 Unauthorized method request for authentication
W^{auth}	WWW-authenticate message request
N_B	Challenger nonce for DAA
N_A	Client nonce for DAA

5 Attacks on registration

SIP communication is vulnerable to several types of attacks, including network layered attacks like denial of service or eavesdropping, and SIP specific attacks like registration hijacking or call redirection.

In our case study, a disgruntled employee could easily exploit the vulnerabilities of the company's VoIP system. By plugging in his laptop with VoIP attacker tool instead of his phone, he could easily launch attacks[3]. Once the attacker has access to the infrastructure, eavesdropping on phone calls are easy since VoIP-related communication in the company are transmitted unencrypted.

SIP calls routed externally to remote hosts, through several Internet domains might be subject to attacks as powerful as the Dolev Yao attacker [5]. The Dolev Yao models means that

(DY-1) cryptography is assumed to be perfect

(DY-2) the attacker controls the entire network

Since the underlying cryptographic operations works perfect (assumption DY-1) the attacker I never can use brute force to break the underlying hashing or encryption algorithms. Hence I cannot extract secret entities or decrypt encrypted messages if I does not possess the required secret entities. However, the Dolev Yao attacker is assumed

[3] In our case, the attacker would have an easy target. All the phones in the company used the last three digits of the phones phone number as the shared secret.

to know every cryptographic operation, like public key schemes, symmetric encryption and decryption, concrete hashing algorithms, ways of using HMAC's, etc.

Assumption DY-2 implies that the attacker acts like a router: all messages pass through I. The attacker has the capability to intercept any message, and fake any message. *Interception* means that the attacker knows what every honest agent is doing. Interception is specified formally as

$$(a) \quad A \longrightarrow I(B) \quad : \quad M,$$

which reads "the intended message $A \longrightarrow B \quad : \quad M$, is picked up by the attacker I". The attacker's capability of *faking* means that it can: (i) impersonate as any other agent, that is assumed to be a honest agent, (ii) inject any compromised entity into the message (limited to the entities that can be obtained by assumption DY1).

The first attack on registration is eavesdropping, which is the basis of most of the succeeding attacks. Several trivial denial of service attacks can be launched by the attacker, by changing plaintext strings, the session nonce N_C^{callid}, or replay an old digest authentication response used between another client C' and the registrar server R. On the server side the latter behavior of the "client" must be considered to be corrupt, and the honest client might be excluded from the service. From the eavesdropping attack, we can construct the following call-hijacking attack on registration that includes digest authentication:

$$\begin{array}{ll}
(R_{1.1.a}^{\text{D}}) & C \longrightarrow I(R) : W^{\text{REGISTER}}, W_C^{\text{Contact}}, N_C^{\text{callid}} \\
(R_{1.1.b}^{\text{D}}) & I(C) \longrightarrow R : W^{\text{REGISTER}}, W_I^{\text{Contact}}, N_C^{\text{callid}} \\
(R_{1.2.a}^{\text{D}}) & R \longrightarrow I(C) : W^{\text{Trying}}, N_C^{\text{callid}} \\
(R_{1.2.b}^{\text{D}}) & I(R) \longrightarrow C : W^{\text{Trying}}, N_C^{\text{callid}} \\
(R_{1.3.a}^{\text{D}}) & R \longrightarrow I(C) : W^{\text{Unauth}}, W^{\text{auth}}, W^{\text{realm}}, N_R, N_C^{\text{callid}} \\
(R_{1.3.b}^{\text{D}}) & I(R) \longrightarrow C : W^{\text{Unauth}}, W^{\text{auth}}, W^{\text{realm}}, N_R, N_C^{\text{callid}} \\
(R_{1.4.a}^{\text{D}}) & C \longrightarrow I(R) : W^{\text{REGISTER}}, N_C^{\text{callid}}, W_C^{\text{uname}}, \\
& \quad W^{\text{realm}}, N_R, W_C^{\text{URI}}, X_{\text{nc}}, N_C, W^{\text{qop}}, \\
& \quad \mathsf{H}[\mathsf{H}[W_C^{\text{uname}}, W^{\text{realm}}, K_{CR}^{\text{pwd}}], N_C, W^{X_{\text{nc}}}, \\
& \quad N_R, W^{\text{qop}}, \mathsf{H}[W^{\text{REGISTER}}, W_C^{\text{URI}}]] \\
(R_{1.4.b}^{\text{D}}) & I(C) \longrightarrow R : W^{\text{REGISTER}}, N_C^{\text{callid}}, W_C^{\text{uname}}, \\
& \quad W^{\text{realm}}, N_R, W_C^{\text{URI}}, X_{\text{nc}}, N_C, W^{\text{qop}}, \\
& \quad \mathsf{H}[\mathsf{H}[W_C^{\text{uname}}, W^{\text{realm}}, K_{CR}^{\text{pwd}}], N_C, X_{\text{nc}}, \\
& \quad N_R, W^{\text{qop}}, \mathsf{H}[W^{\text{REGISTER}}, W_C^{\text{URI}}]] \\
(R_{1.5.a}^{\text{D}}) & R \longrightarrow I(C) : W^{\text{Trying}}, W_I^{\text{Contact}}, N_C^{\text{callid}} \\
(R_{1.5.b}^{\text{D}}) & I(R) \longrightarrow C : W^{\text{Trying}}, W_C^{\text{Contact}}, N_C^{\text{callid}} \\
(R_{1.6.a}^{\text{D}}) & R \longrightarrow I(C) : W_{\text{OK}}, W_I^{\text{Contact}}, N_C^{\text{callid}} \\
(R_{1.6.b}^{\text{D}}) & I(R) \longrightarrow C : W_{\text{OK}}, W_C^{\text{Contact}}, N_C^{\text{callid}}
\end{array}$$

In the attack, the malicious agent I is able to manipulate the client C to believe that she has successfully registered the additional contact location W_C^{Contact}, while the registration server is fooled to believe that C should be contacted using the corrupt address W_I^{Contact}, which is a location that the attacker I controls. In future deployment of SIP signaling and phone calls, the call is routed to the attackers contact address W_I^{Contact}.

The attacker I is passive in the attack clauses $R_{1.2.a}^{\text{D}}$ to $R_{1.4.b}^{\text{D}}$, corresponding to the protocol clauses P_2^{D} and P_4^{D}, the part of the protocol where authentication occurs, while I is active and injecting the corrupt contact address in the protocol clauses $(\text{P}_1^{\text{D}}, \text{P}_5^{\text{D}}, \text{P}_6^{\text{D}})$. A timely question is what kind of authentication or integrity guarantees are given by the application of Digest Access Authentication. The shared secret does not prevent the attacker to compromise the contact address.The attack can be prevented by changing the Digest response to include the contact address(es):

$$\begin{array}{l}
\mathsf{H}[\mathsf{H}[W_C^{\text{uname}}, W^{\text{realm}}, K_{CR}^{\text{pwd}}], W_C^{\text{Contact}}, \\
\qquad N_R, X_{\text{nc}}, N_C, W^{\text{qop}}, \mathsf{H}[W^{\text{REGISTER}}, W_C^{\text{URI}}]]
\end{array}$$

When the registrar receives the response, the integrity of the contact address is preserved and fake contact addresses might be discovered. An alternative solution is to keep the specification unchanged but let $W_C^{\text{URI}} = W_C^{\text{Contact}}$ in order ascertain the integrity of the contact address.

5.1 Analysis of SIP in PROSA

The translation from the standard protocol specifications in Section 4 is straightforward as shown in Figure 3. A concise specification of the registration sub-protocol was obtained quickly by using *Wireshark*. It took approximately six man hours of work to have a executable specification of the concrete setup of SIP, thanks to the advanced validation mechanism of PROSA [9]. A test-scenario with the user and the Asterisk server was configured: Each agent possesses the protocol and capability of constructing appropriate nonce. Three configurations of the scenario were applied, that is, simulations with: (a) perfect network and no attacker, (b) an eavesdropper sniffing all messages, (c) an active call-hijacking attack. In each case Digest authentication was used, with assumption (DY-1). The integrity of the contact address was not preserved in the active call-hijacking attack: A query on the final state shows that the attacker successfully had compromised the address and could thereby redirect all calls through its own device.

6 Conclusion

We showed that for large and complex protocols like SIP it is possible to formally specify the details of the message exchange on a level that permits automated security analysis. The formal specification and simulation of protocols from IETF, like SIP, reveals several potential and real deficiencies that are not easily spot reading informal descriptions like RFCs.

A severe attack on registration was discovered in this paper, an instance of the general call-hijacking attacks. Both

$$
\begin{aligned}
&\mathsf{protocol}[\mathrm{SIP-register}, 0,\\
&\quad \mathsf{role}(A) \wedge \mathsf{role}(R),\\
&\quad \mathsf{role}(A) \wedge \mathsf{role}(R), \mathsf{role}(A),\\
&\\
&\mathsf{Bel}_A(\mathsf{isKey}(\mathsf{key}(\mathsf{s}, A, R)))\\
&\mathscr{B}\ \mathsf{Transmit}\ (A, R, \mathsf{Text}(\text{REGISTER}) \wedge\\
&\quad \mathsf{Text}(\text{CONTACT}) \wedge \mathsf{isNonce}(\mathsf{n}(\text{CALLID}, A)))\\
&\mathscr{B}\ \mathsf{Transmit}\ (R, A, \mathsf{Text}(\text{100 TRYING}) \wedge \mathsf{isNonce}(\mathsf{n}(\text{CALLID}, A)))\\
&\mathscr{B}\ \mathsf{Transmit}\ (R, A, \mathsf{Text}(\text{401 Unauthorized}) \wedge \mathsf{Text}(\text{WWW-Authenticate}) \wedge\\
&\quad \mathsf{Text}(\text{Asterisk}) \wedge \mathsf{isNonce}(\mathsf{n}(\text{MD5}, R)) \wedge \mathsf{isNonce}(\mathsf{n}(\text{CALLID}, A)))\\
&\mathscr{B}\ \mathsf{Transmit}\ (A, R, \mathsf{Text}(\text{REGISTER}) \wedge \mathsf{Text}(\text{Username}) \wedge \mathsf{Text}(\text{Asterisk})\\
&\quad \wedge\ \mathsf{isNonce}\ (\mathsf{n}(\text{MD5}, R)) \wedge \mathsf{Agent}(R) \wedge \mathsf{isNonce}(\mathsf{n}(\text{MD5A}, A)) \wedge\\
&\quad \mathsf{Text}(\text{Nonce Counter}) \wedge \mathsf{Text}(\text{Auth}) \wedge\\
&\ \mathsf{Hash}[\mathsf{Hash}[\mathsf{Text}(\text{Username}) \wedge \mathsf{Text}(\text{Asterisk}) \wedge \mathsf{isKey}(\mathsf{key}(\mathsf{s}, A, R))] \wedge\\
&\quad \mathsf{isNonce}(\mathsf{n}(\text{MD5}, R)) \wedge \mathsf{Text}(\text{Nonce Counter}) \wedge\\
&\quad \mathsf{isNonce}(\mathsf{n}(\text{MD5A}, A)) \wedge \mathsf{Text}(\text{Auth})\\
&\wedge\ \mathsf{Hash}\ [\mathsf{Text}(\text{REGISTER}) \wedge \mathsf{Agent}(R)]] \wedge \mathsf{isNonce}(\mathsf{n}(\text{CALLID}, A)))\\
&\mathscr{B}\ \mathsf{Transmit}\ (R, A, \mathsf{Text}(\text{100 TRYING}) \wedge\\
&\quad \mathsf{Text}(\text{CONTACT}) \wedge \mathsf{isNonce}(\mathsf{n}(\text{CALLID}, A)))\\
&\mathscr{B}\ \mathsf{Transmit}\ (R, A, \mathsf{Text}(\text{200 OK}) \wedge\\
&\quad \mathsf{Text}(\text{CONTACT}) \wedge \mathsf{isNonce}(\mathsf{n}(\text{CALLID}, A)))\\
&\mathscr{B}\ \varepsilon]
\end{aligned}
$$

Figure 3. SIP registration specified in PROSA.

the discovery as well as the repair relied on the formal specification of the case study that was investigated. A practical problem discussed in this paper is that when mechanisms, like Digest Access Authentication, is combined with a given protocol, like registration, the security of the combined protocol is only clearly understood when it is formally analyzed. The approach taken in the paper can be used to rapidly specify and analyse different aspects and scenarios of SIP, closely related to implementations. Various configuration Call setup, Secure SIP, routing over several proxies can be explored with the same techniques as proposed in the paper.

Acknowledgments

This research is funded by the EUX2010SEC project in the VERDIKT framework of the Norwegian Research Council. The authors would like to thank Wolfgang Leister, Arne-Kristian Groven, Lothar Fritsch, Bjarte M. Østvold and the anonymous reviewers for comments on earlier drafts of this paper.

References

[1] IAX: Inter-Asterisk eXchange Version 2. Internt-Draft v04, mar 2008.

[2] J. Arkko, V. Torvinen, G. Camarillo, A. Niemi, and T. Haukka. Security Mechanism Agreement for the Session Initiation Protocol (SIP). RFC 3329 (Proposed Standard), January 2003.

[3] Steve Bishop, Matthew Fairbairn, Michael Norrish, Peter Sewell, Michael Smith, and Keith Wansbrough. Rigorous specification and conformance testing techniques for network protocols, as applied to TCP, UDP, and sockets. *SIGCOMM Comput. Commun. Rev.*, 35(4):265–276, 2005.

[4] Geneiatakis D., Kambourakis G., Dagiuklas T., Lambrinoudakis C., and Gritzalis S. SIP Security Mechanisms: A state-of-the-art review. In *Proceedings of the Fifth International Network Conference (INC 2005)*, pages 147–155, Jul 2005.

[5] Danny Dolev and Andrew Chi-Chih Yao. On the security of public key protocols. *IEEE Transactions on Information Theory*, 29(2):198–207, 1983.

[6] David Endler and Mark Collier. *Hacking Exposed VoIP: Voice over IP Security Secrets and Solutions*. McGraw-Hill Osborne Media, Nov 2006.

[7] J. Franks, P. Hallam-Baker, J. Hostetler, S. Lawrence, P. Leach, A. Luotonen, and L. Stewart. HTTP Authentication: Basic and Digest Access Authentication. RFC 2617 (Draft Standard), June 1999.

[8] Prateek Gupta and Vitaly Shmatikov. Security Analysis of Voice-over-IP Protocols. In *Computer Security Foundations Symposium, 2007. CSF '07. 20th IEEE*, pages 49–63, 2007.

[9] Anders Moen Hagalisletto. Validating Attacks on Authentication Protocols. In *Proceedings of the 12th IEEE Symposium on Computers and Communications - (ISCC 2007), July 1-4, Aveiro, Portugal*.

[10] D.R. Kuhn. Sources of failure in the public switched telephone network. *Computer*, 30:31–36, 1997.

[11] Wenbo Mao. *Modern Cryptography: Theory and Practice*. Prentice Hall, 2004.

[12] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler. SIP: Session Initiation Protocol. RFC 3261 (Proposed Standard), June 2002. Updated by RFCs 3265, 3853, 4320, 4916.

[13] S. Salsano, L. Veltri, and D. Papalilo. SIP security issues: The SIP authentication procedure and its processing load. *Network, IEEE*, 16:38–44, 2002.

[14] Henry Sinnreich and Alan B. Johnston. *Internet communications using SIP: Delivering VoIP and multimedia services with Session Initiation Protocol*. John Wiley & Sons, Inc., New York, NY, USA, second edition, Aug 2006.