Formal modeling of authentication in SIP

Anders Moen Hagalisletto and Lars Strand

Norwegian Computing Center

The Second International Conference on Emerging Security Information, Systems and Technologies (SECURWARE 2008)

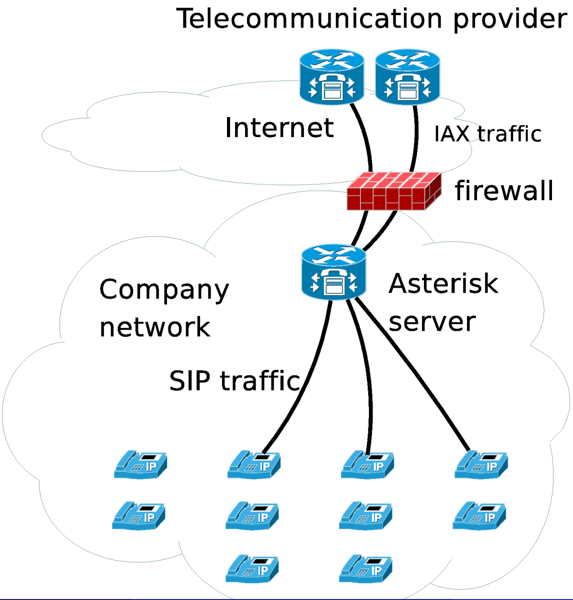
Outline

1. Voice-over-IP - Session Initiation Protocol (SIP)
2. Our test case scenario
3. Method and tools: Analyzing implementation rather than specification
4. Digest Access Authentication
5. Formal modeling PROSA
6. Results
7. Conclusion and further work.

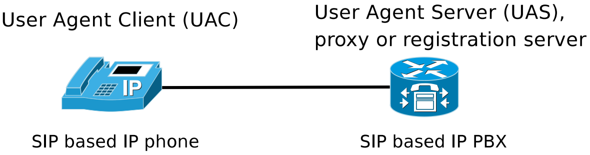
Voice over IP

- Voice over IP (VoIP) protocols and technology is a merge of telecom and data communication
- Industry have high focus on VoIP today.
- VoIP is known to be unsecure!
- Multiple attacks on SIP based VoIP exists
- We will focus on authentication in SIP
- Norwegian Computing Center evaluates various architectures and protocols of Voice over IP
    - Session Initiation Protocol (SIP) RFC 3261
    - Interasterisk Exchange IAX (RFC draft only)
- Project: EUX2010SEC, `http://eux2010sec.nr.no/`

VoIP case-study - three protocols: SIP, RTP and IAX



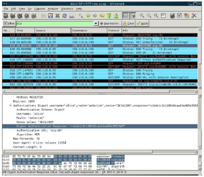
Method



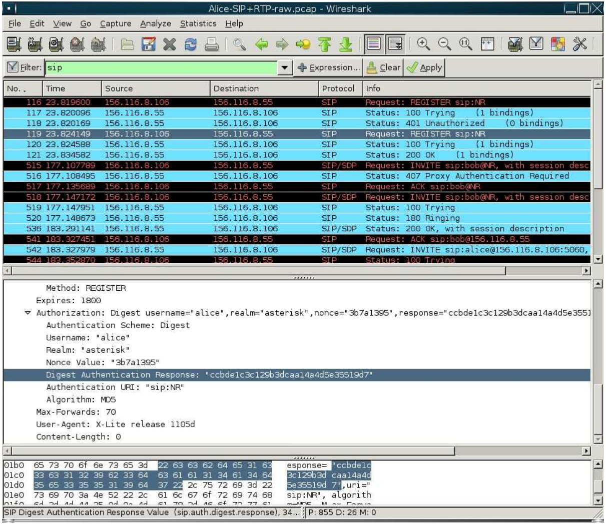
1. Experiment
   - *"Don't trust the documentation"*
   - Lab test stup: Replicate test scenario.
   - Software: Asterisk PBX and X-Lite softphones.
2. Active observation
   - Using the network monitoring tool "Wireshark".
3. Formal protocol analysis.
   - PROSA

Network tool Wireshark

- A network monitoring tool.
  - Sniff the network
  - Parse the result and compare against the standard.
- Why did we use Wireshark?
  - Compare implementation against the SIP standard.
  - Result used as basis for modelling in PROSA.

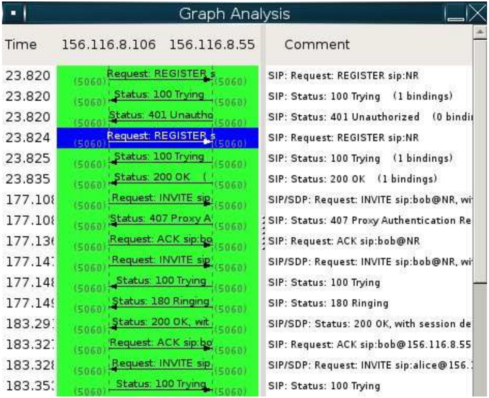


Wireshark

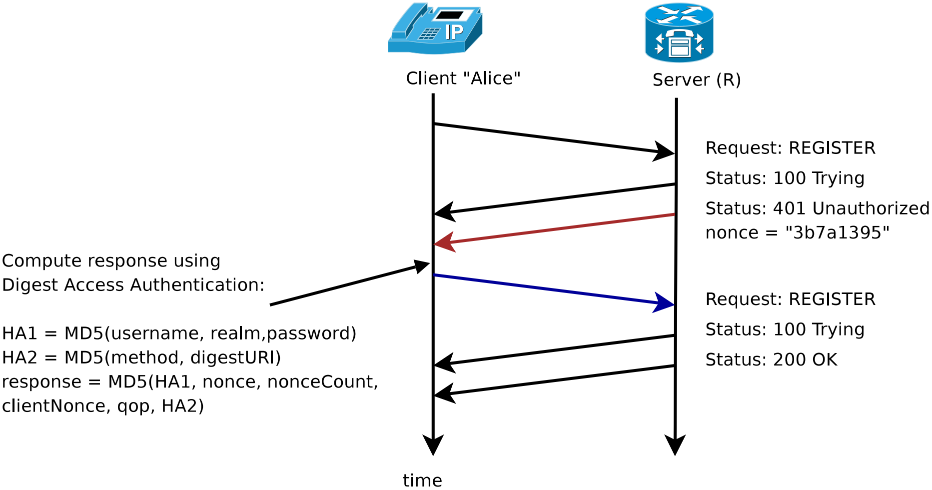


Alice-SIP+RTP-raw.pcap - Wireshark
File Edit View Go Capture Analyze Statistics Help
Filter: sip
Expression... Clear Apply
No. Time Source Destination Protocol Info
116 23.819600 156.116.8.106 156.116.8.55 SIP Request: REGISTER sip:NR
117 23.820096 156.116.8.55 156.116.8.106 SIP Status: 100 Trying (1 bindings)
118 23.820169 156.116.8.55 156.116.8.106 SIP Status: 401 Unauthorized (0 bindings)
119 23.824149 156.116.8.106 156.116.8.55 SIP Request: REGISTER sip:NR
120 23.824588 156.116.8.55 156.116.8.106 SIP Status: 100 Trying (1 bindings)
121 23.834582 156.116.8.55 156.116.8.106 SIP Status: 200 OK (1 bindings)
515 177.107789 156.116.8.106 156.116.8.55 SIP/SDP Request: INVITE sip:bob@NR, with session desc
516 177.108495 156.116.8.55 156.116.8.106 SIP Status: 407 Proxy Authentication Required
517 177.135689 156.116.8.106 156.116.8.55 SIP Request: ACK sip:bob@NR
518 177.147172 156.116.8.106 156.116.8.55 SIP/SDP Request: INVITE sip:bob@NR, with session desc
519 177.147951 156.116.8.55 156.116.8.106 SIP Status: 100 Trying
520 177.148673 156.116.8.55 156.116.8.106 SIP Status: 180 Ringing
536 183.291141 156.116.8.55 156.116.8.106 SIP/SDP Status: 200 OK, with session description
541 183.327451 156.116.8.106 156.116.8.55 SIP Request: ACK sip:bob@156.116.8.55
542 183.327979 156.116.8.55 156.116.8.106 SIP/SDP Request: INVITE sip:alice@156.116.8.106:5060,
544 183.352870 156.116.8.106 156.116.8.55 SIP Status: 100 Trying
Method: REGISTER
Expires: 1800
Authorization: Digest username="alice",realm="asterisk",nonce="3b7a1395",response="ccbde1c3c129b3dcaa14a4d5e3551
Authentication Scheme: Digest
Username: "alice"
Realm: "asterisk"
Nonce Value: "3b7a1395"
Digest Authentication Response: "ccbde1c3c129b3dcaa14a4d5e35519d7"
Authentication URI: "sip:NR"
Algorithm: MD5
Max-Forwards: 70
User-Agent: X-Lite release 1105d
Content-Length: 0
01b0 65 73 70 6f 6e 73 65 3d 22 63 63 62 64 65 31 63 esponse= "ccbde1c
01c0 33 63 31 32 39 62 33 64 63 61 61 31 34 61 34 64 3c129b3d caa14a4d
01d0 35 65 33 35 35 31 39 64 37 22 2c 75 72 69 3d 22 5e35519d 7",uri="
01e0 73 69 70 3a 4e 52 22 2c 61 6c 67 6f 72 69 74 68 sip:NR", algorith
SIP Digest Authentication Response Value (sip.auth.digest.response), 34...
P: 855 D: 26 M: 0

Wireshark


Graph Analysis
Time
156.116.8.106
156.116.8.55
Comment
23.820
23.820
23.820
23.824
23.825
23.835
Request: REGISTER
Status: 100 Trying
Status: 401 Unautho
Request: REGISTER
Status: 100 Trying
Status: 200 OK
Request: INVITE sip
Status: 407 Proxy A
Request: ACK sip:bo
Request: INVITE sip
Status: 100 Trying
Status: 180 Ringing
Status: 200 OK, wit
Request: ACK sip:bo
Request: INVITE sip
Status: 100 Trying
SIP: Request: REGISTER sip:NR
SIP: Status: 100 Trying (1 bindings)
SIP: Status: 401 Unauthorized
SIP: Request: REGISTER sip:NR
SIP: Status: 100 Trying (1 bindings)
SIP: Status: 200 OK (1 bindings)
SIP/SDP: Request: INVITE sip:bob@NR, wi
SIP: Status: 407 Proxy Authentication Re
SIP: Request: ACK sip:bob@NR
SIP/SDP: Request: INVITE sip:bob@NR, wi
SIP: Status: 100 Trying
SIP: Status: 180 Ringing
SIP/SDP: Status: 200 OK, with session de
SIP: Request: ACK sip:bob@156.116.8.55
SIP/SDP: Request: INVITE sip:alice@156.
SIP: Status: 100 Trying

SIP REGISTER



Why use formal methods?

Because

1. the only way to prove or verify that protocols fulfills their goals!
2. has been used to find new attacks on protocols
3. implicitly gives a unambigous specification of
   1. the protocol's interactions and entities
   2. the *functional* and *security goals*
4. the protocol specification can be analyzed automatically

The Dolev Yao model

A Dolev Yao attacker

1. controls the entire network
2. does not have access to secret entities (keys)
3. can intercept any message
4. can send any message (based on her knowledge)

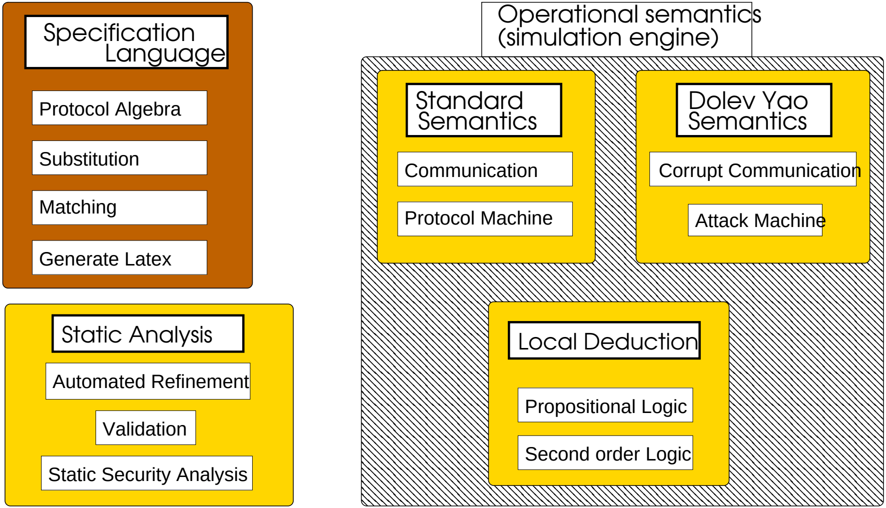
The latter means that it can inject anything into a concrete message, even the entire message content can be changed.

The PROSA tool - specification of protocols

1. Formal language PROSA contains
   1. all necessary primitives and operators for cryptography
   2. contains opertor: *Agent A believes that ...*
2. The PROSA tool includes a static validation module
   1. *automated refinement*
   2. *validation of refined specs*
3. simulation and analysis

Note: Both tools and theory rely on the Dolev Yao model.

The PROSA tool - specification of protocols



Standard notation: Security Protocols

A protocol clause is written:

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

meaning *"agent A sends a message M to the agent B"*

$A, B, C, S, I, I(A)$	agent terms
K_{AB}	symmetric key shared by A, B
K_A	A's public key
K_A^{-1}	A's private key
N_A	nonce generated by agent A
W_A^Y	string containing the text Y related to agent A
X_A	miscellaneous entities

Composition operators:

- concatenation of message content denoted by "," (comma),
- hashing $\mathrm{H}[M]$, and
- encryption $E(K : M)$, where K is a key and M a message content.

Digest Access authentication specified precisely

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:

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

Registration sub-protocol

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

Establish a (possibly new) contact point,

$$W_C^{\mathrm{Contact}}$$

a phone number, email address, etc.

Analyzing SIP authentication

We analyzed the registration sub-protocol in Case Study.
SIP authentication on registration $=$

registration $\boxplus$ *Digest authentication*

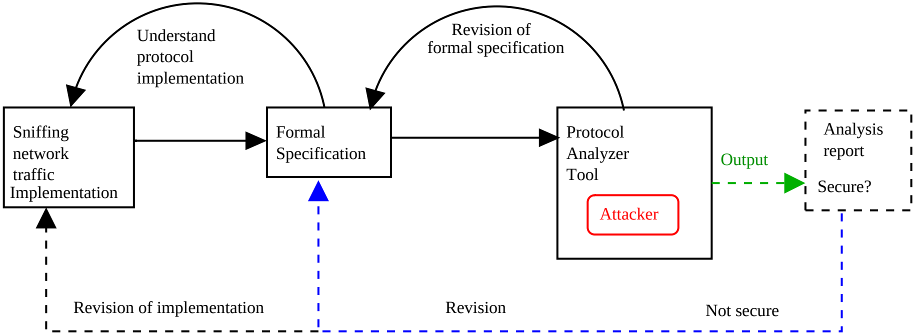
But what exactly means the composition $\boxplus$*?*

The exact behaviour not specified explicitly:
We used RFC and Wireshark to find out!

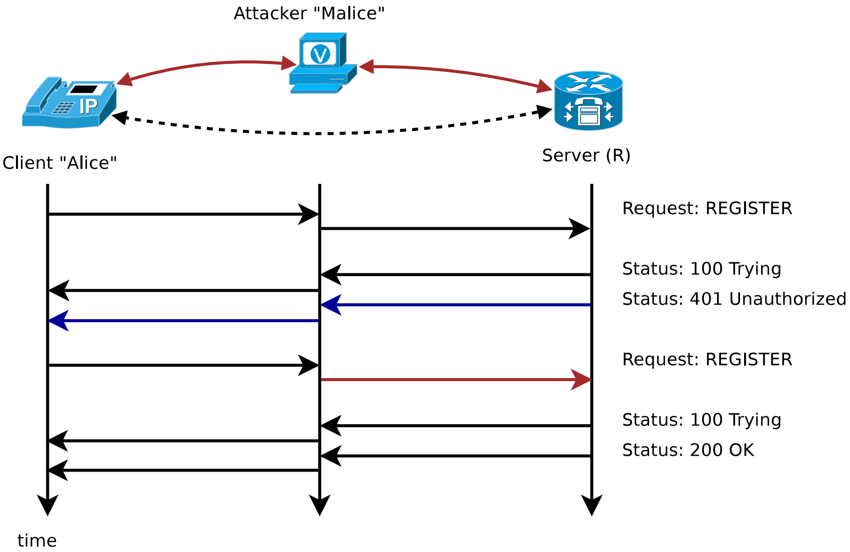
Registration with Digest Access authentication (Wireshark)

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

Typical Workflow: Analysis of implementation



A large picture on the attack



Attack on registration

$$
\begin{array}{lll}
(R^{\mathrm{D}}_{1.1.a}) & C \longrightarrow I(R) : & W^{\mathrm{REGISTER}}, W^{\mathrm{Contact}}_{C}, N^{\mathrm{callid}}_{C} \\
(R^{\mathrm{D}}_{1.1.b}) & I(C) \longrightarrow R : & W^{\mathrm{REGISTER}}, \color{red}{W^{\mathrm{Contact}}_{I}}, N^{\mathrm{callid}}_{C} \\
(R^{\mathrm{D}}_{1.2.a}) & R \longrightarrow I(C) : & W^{\mathrm{Trying}}, N^{\mathrm{callid}}_{C} \\
(R^{\mathrm{D}}_{1.2.b}) & I(R) \longrightarrow C : & W^{\mathrm{Trying}}, N^{\mathrm{callid}}_{C} \\
(R^{\mathrm{D}}_{1.3.a}) & R \longrightarrow I(C) : & W^{\mathrm{Unauth}}, W^{\mathrm{auth}}, W^{\mathrm{realm}}, N_{R}, N^{\mathrm{callid}}_{C} \\
(R^{\mathrm{D}}_{1.3.b}) & I(R) \longrightarrow C : & W^{\mathrm{Unauth}}, W^{\mathrm{auth}}, W^{\mathrm{realm}}, N_{R}, N^{\mathrm{callid}}_{C} \\
(R^{\mathrm{D}}_{1.4.a}) & C \longrightarrow I(R) : & W^{\mathrm{REGISTER}}, N^{\mathrm{callid}}_{C}, W^{\mathrm{uname}}_{C}, \\
& & W^{\mathrm{realm}}, N_{R}, W^{\mathrm{URI}}_{C}, X_{\mathrm{nc}}, N_{C}, W^{\mathrm{qop}} \\
& & \mathsf{H}[\mathsf{H}[W^{\mathrm{uname}}_{C}, W^{\mathrm{realm}}, K^{\mathrm{pwd}}_{CR}], N_{C}, X_{\mathrm{nc}}, \\
& & N_{R}, W^{\mathrm{qop}}, \mathsf{H}[W^{\mathrm{REGISTER}}, W^{\mathrm{URI}}_{C}]] \\
(R^{\mathrm{D}}_{1.4.b}) & I(C) \longrightarrow R : & W^{\mathrm{REGISTER}}, N^{\mathrm{callid}}_{C}, W^{\mathrm{uname}}_{C}, \\
& & W^{\mathrm{realm}}, N_{R}, W^{\mathrm{URI}}_{C}, X_{\mathrm{nc}}, N_{C}, W^{\mathrm{qop}} \\
& & \mathsf{H}[\mathsf{H}[W^{\mathrm{uname}}_{C}, W^{\mathrm{realm}}, K^{\mathrm{pwd}}_{CR}], N_{C}, X_{\mathrm{nc}}, \\
& & N_{R}, W^{\mathrm{qop}}, \mathsf{H}[W^{\mathrm{REGISTER}}, W^{\mathrm{URI}}_{C}]] \\
(R^{\mathrm{D}}_{1.5.a}) & R \longrightarrow I(C) : & W^{\mathrm{Trying}}, \color{red}{W^{\mathrm{Contact}}_{I}}, N^{\mathrm{callid}}_{C} \\
(R^{\mathrm{D}}_{1.5.b}) & I(R) \longrightarrow C : & W^{\mathrm{Trying}}, W^{\mathrm{Contact}}_{C}, N^{\mathrm{callid}}_{C} \\
(R^{\mathrm{D}}_{1.6.a}) & R \longrightarrow I(C) : & W_{\mathrm{OK}}, \color{red}{W^{\mathrm{Contact}}_{I}}, N^{\mathrm{callid}}_{C} \\
(R^{\mathrm{D}}_{1.6.b}) & I(R) \longrightarrow C : & W_{\mathrm{OK}}, W^{\mathrm{Contact}}_{C}, N^{\mathrm{callid}}_{C}
\end{array}
$$

$$
\begin{array}{ll}
\mathsf{Bel}_C(\mathsf{Bel}_R(\mathsf{Bel}_C(W^{\mathrm{Contact}}_{C}))) & \mathsf{TRUE} \\
\mathsf{Bel}_R(\mathsf{Bel}_C(W^{\mathrm{Contact}}_{C})) & \color{red}{\mathsf{FALSE}}
\end{array}
$$

Discussion

1. Contact address of Alice is compromized (attack on authenticity/integrity)
2. Easy to spot security errors when we have a precise specification
3. Easy to fix attack in theory:

The attack can be prevented by changing the Digest response to include the contact address(es):

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

Hence: the specification must be changed!

Conclusion

- SIP is a huge and feature-rich protocol standard
- But SIP REGISTRATION ⊞ Digest authentication = leads to REGISTRATION attack
- This attack can be prevented by modifying the Digest.
- Formalizing protocols with tools support aids in discover new attacks
- Future work: Deploy same procedure for IAX protocol - compare SIP and IAX