Security Analysis of Real-time Protocols: SIP, Secured SIP, and SRTP

Alexander L. Churchill achur@stanford.edu

Emmanouel Liodakis liodakis@stanford.edu

Mar. 14, 2011

Contents

1 Introduction

The Session Initiation Protocol (SIP) is a signaling protocol designed to control multimedia communication sessions. The basic structure of SIP involves registration with a proxy server, calls to invite, ring, and ready, a media session, and bye messages with acknowledgements. Because SIP is designed for flexibility across devices and simplicity, there is no implicit security in the SIP protocol. The SIP standard described in RFC 3621 [1] includes suggested security protocols designed to provide user-to-user security across the protocol. The Secure Real-time Transport Protocol (SRTP) is a profile of RTP designed to provide message security, authentication, integrity, and replay protection across media sessions. In this paper, we analyze the security of the standard implementation of SIP, the recommended secure implementation of SIP as per RFC 3621, and the security of a media session sent over SRTP.

2 Protocol Description

2.1 SIP

The SIP protocol first defines the protocol for registration with a proxy. An agent sends a REGISTER message to a SIP proxy. The proxy registers the agent and responds with an OK message. After agents have registered with a proxy, they can begin the call flow shown in Figure 1 [2]. Alice sends an INVITE message to the proxy to be passed to Bob; the proxy passes along the INVITE and responds to Alice with a TRYING message. Bob responds to the proxy with a RINGING message, then an OK message, both of which are passed by the proxy to Alice. Alice then sends an ACK to the proxy which gets passed to Bob. At this point, Alice and Bob engage in a media session over RTP or SRTP. When Bob is ready to terminate the media session, he sends a BYE message to which Alice replies with an OK and both agents terminate the call.

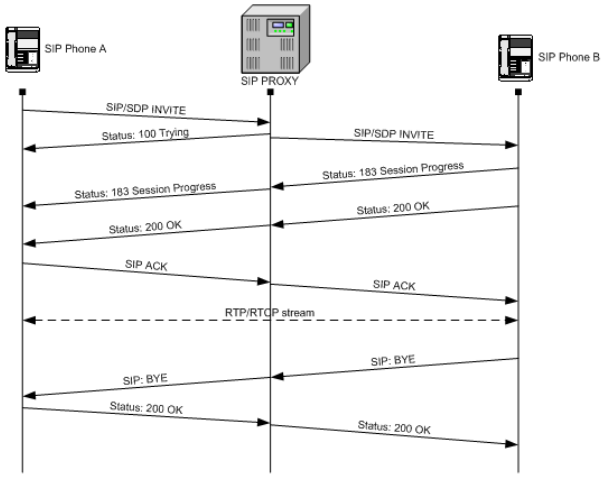

Figure 1: SIP Call Flow

2.2 Secured SIP

The SIP protocol described in RFC 3621 describes a proscribed secure setup for SIP. Proxies are required to support SSL/TLS. Agents are recommended but not required to send all messages over TLS. The SIP specification suggests that the media session be encrypted, but on the premise that SIP is only a signaling protocol, does not engage a method to secure the media session.

2.3 SRTP

SRTP defines a profile of RTP that is intended to provide (1) encryption of the payload, (2) message authentication of packets, (3) transmission integrity and (4) replay authentication in both unicast and multicast applications. As a profile of RTP, the goal of SRTP is to add minimal overhead to RTP packets while still securing the connection. Therefore, the SRTP packet header (which includes the padding, sequence number, timestamp, SSRC, CSRC, and RTP extension) is identical to the RTP packet header. However, SRTP packets differ in the bottom three sections of the packet. In SRTP the payload is encrypted, there is an option STRP master key index, and a *recommended*, but not required, authentication tag as shown in Figure 2 [3].

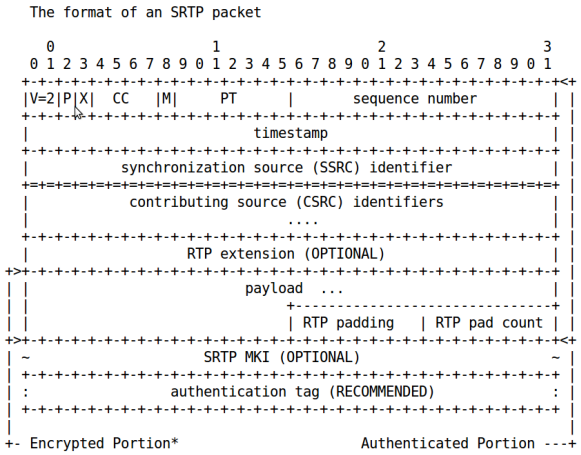

Figure 2: SRTP Packet

The key exchange protocol is not discussed in the SRTP specification; however, it is assumed that a cryptographically-secure public key exchange protocol is used in the initial handshake and then after session keys are determined the payload is encrypted with these keys. The protocol recommends an AES-CRT encryption scheme, which we know is currently considered cryptographically-secure. Additionally, the protocol specifies that the *recommended* authentication tag takes the entire packet and MACs it.

As each packet is sent across the media session the receiver keeps track of the packet sequence numbers. If the same packet comes across multiple times the receiver discards it. Also, if packets come out of order, the receiver is able to reorder them.

3 Objectives

The purpose of our analysis is to analyze the weaknesses presented by the unsecured SIP protocol and determine of those weaknesses are shored-up by the standard secure setup for SIP communications, using SRTP for a media session. A secure protocol would ideally provide a secure signaling and transmission method for a secured user, or failing that, provide a secure signaling and transmission protocol, assuming both users are secured.

4 Methodology

We modeled three separate protocols in Murϕ. We modeled the SIP protocol, unmodified, with no security properties, the secured SIP setup suggested by the SIP specification, and the SRTP protocol.

4.1 SIP and Secured SIP

For SIP, we model four types of agents: Initiators, Responders, Proxies, and Intruders. Initiators and Responders each keep track of their current state, their proxy, and their responder or initiator respectively. Proxies keep track of their state and a list of registered agents. Intruders keep track of a list of agents whose media session messages they have intercepted. In the secured setup, intruders also keep track of a key they use to spoof secure messages of an unsecured agent. Messages are sucinctly modeled with a source and destination, a via field that is used to specify the eventual target of a message (e.g. dest: proxy.com, via: Bob) or the original sender, rather than the proxy passing along the message, and a message type field. In Secured SIP, messages also include a "key" property, which should match the "from" field. Keyed messages can only be read by the owner of the key and the destination of the message. In Secured SIP, we modeled two cases: in one, the responder sends messages without security and we test the invariants pertaining to secure communications on behalf of the initiator; in the second, we model both the initiator and responder as using secured communications.

4.1.1 Simplifications

The standard SIP protocol involves many proxies – two users might be registered with the same proxy (also called domain) or might be registered with different proxies. For the purpose of modeling both SIP and secure SIP, we assume that Alice and Bob register the same proxy. Provided the assumptions below, this is a fair simplification, given that the SIP protocol requires that proxies implement secure communication, so inter-proxy communication is secured.

4.1.2 Assumptions

Based on prior analysis of SSL [4], we make the assumption that an agent can only read a message over SSL if it is either the intended recipiant or if it has stolen a private key or convinced the sender its public key belongs to another agent.

4.1.3 Invariants

There are several key invariants that must hold for the SIP protocol to be secure. Agents that try to register must be able to do so. No legitimate agent should double register. The responder's initator should be the responder, and likewise the initiator's responder should be the initiator. Two agents who complete a media session should concur on finish

– one should not remain open while the other closes the connection. Most importantly, the intruder should not be able to read messages sent during the media session:

```
invariant "Intruder doesn't see our messages"
  forall i : IntruderId do
    forall j : InitiatorId do
      multisetcount(l:int[i].intercepted, j = int[i].intercepted[l]) = 0
    end
  end;
```

4.2 SRTP

Our model of SRTP employs three agents: Initiator, Receiver, and Intruder. First, the Initiator and Receiver exchange public keys and establish a joint secret key, then a media session is established and packets are sent from the Initiator to the Receiver. The Initiator and Receiver keep track of their own state, while the Intruder keeps track of the list of messages exchanged between the the Initiator and Receiver.

4.2.1 Assumptions and Simplifications

In order to simplify the analysis of SRTP we make certain assumptions and simplifications. We specifically deal with unicast applications – those applications involving communication between a single sender and a single receiver over a network. Our reasoning for this simplification is that any security flaws found in a unicast system could be easily demonstrated in a multicast system. Additionally, we assume that the encryption scheme used by SRTP is cryptographically secure. Specifically, the key exchange protocols, implementation of AES counter mode, and pseudo-random number generation techniques are implemented in a cryptographically-secure fashion.

We also analyze SRTP both with and without the *recommended* message-authentication tag. The message authentication tag can increase the size of each packet in the SRTP stream by fifty percent [5]. It is therefore common for VoIP communication streams with bandwidth constraints to not include this tag.

4.2.2 Invariants

In order for SRTP to be secure the Receiver should not accept packets sent from the Initiator but modified by the Intruder:

```
invariant "Intruder was able to modify packets going to Client."
    forall i: ClientId do
        res[i].state = C_ACCEPT
        ->
        forall j: IntruderId do
```

```
            int[j].canModify[i] = false
        end
    end;
```

4.3 Attacker Model

In all cases, we model the attacker using the Dolev-Yao model, in which the attacker can overhear, intercept, and synthesize any message up to cryptographic limits. We assume that messages are cryptographically securely encrypted and an attacker can only read a packet provided it is either unencrypted or the attacker has the private key of the message. The attacker has a limited memory and can remember some number of previously sent messages as well as one key used to spoof an agent at a time.

5 Results

5.1 SIP

Unsurprisingly, SIP is completely insecure and vulnearable at literally every point of the protocol, assuming the Dolev-Yao model. Every invariant was triggered in runs of the program. SIP is vulnerable to the following:

1. False/Double/Blocked Registration: an agent is either registered without their knowledge, registered twice, or the proxy is not aware of an agent that believes it has registered.
2. False Initiation: an intruder can initiate a conversation on behalf of Alice.
3. False Response: an intruder can send a response to an initiation on behalf of Bob.
4. Early/Blocked Ready: an intruder can send one agent into the media session state before the other agent is ready or prevent an agent from entering the media session even when the other agent is ready.
5. Replay Attack: an intruder can replay messages sent in the media session
6. Session Termination (Early Bye): an intruder can force a media session to an early close.

as well as reading and modifying all packets of the media session.

5.2 Secure SIP

Secure SIP prevented almost all of the attacks against SIP, but the one-sided secured model (in which case Alice secures all her messages but Bob does not secure his) triggered an invariant: the intruder was able to read the messages sent in the media session (and in fact could modify said messages, provided the intruder pretended to be Bob).

5.2.1 The False Registration Attack

In the attack found by Murϕ, an attacker intercepts Bob's unsecured attempt to register with a proxy and registers in Bob's place. The intruder then has all privileges assigned to Bob (and since Bob is unsecured, can freely pass along messages to Bob, spoofing as Alice). The attack flow is as follows:

1. Alice registers securely with a proxy
2. Bob sends unsecured REGISTER message to proxy
3. Intruder intercepts, chooses a key to associate with Bob, and sends secure REGISTER message to proxy as Bob, then sends OK message to Bob with the proxy in the "from" field
4. Alice initiates a conversation with Bob
5. Proxy forwards INVITE to intruder who forwards invite to Bob
6. Bob replies to proxy with RINGING and READY messages
7. Intruder intercepts RINGING and READY and resends to proxy
8. Alice sends ACK to intruder, who sends ACK to Bob
9. Intruder reads, stores, and forwards all messages between Bob and Alice

In the two-sided secured model (in which both Alice and Bob use secured messages), Murϕ found no attacks after expanding X states and firing Y rules.

5.3 SRTP

In order to check the security of SRTP we modeled the protocol in Murϕ both with and without message authentication. When message authentication was provided the protocol was able to securely transfer packets from client 1 to client 2. An attacker could not decrypt payloads because of the encryption scheme and any modification of packets was noticed by the receiver because of the MACs. An attacker could stop packets from reaching the receiver, but that is expected when the protocol has to deal with a network attacker. Additionally, SRTP was vulnerable to data analysis attacks. Information can be attained by simply looking at the size of the packets sent across the network. The SRTP scheme does not help with these types of attacks; however, SRTP does provide security for 1, 2, 3, and 4 from the description.

If the message authentication tag is not included, the same security is not guaranteed. The payload will remain encrypted, thus the first security goal will be met; however, the next three will not. The SRTP sequence number is transferred in the clear, thus an attacker has the ability to modify packet sequence numbers. This destroys the integrity of

the transmission because he can modify packets in such a way that they are received and accepted out of order. This same analysis can be extended to show that neither message authentication nor transmission integrity can be guaranteed.

6 Conclusions and Recommendations

SIP was never designed for security, but its large number of messages and complexity of agents makes it an easy target for intruders – it's large number of messages is only exceeded by its large number of attacks. In our model, the security recommendations proposed in RFC 3621 were highly effective, shutting out the vast majority of attacks against SIP. We were unable to maintain one-sided security: that is, if Alice implements secured SIP correctly and communicates with an honest and secure proxy, she is still unable to guarantee the security of the signaling and communication within the protocol. However, provided that all parties are forced to securely implement the protocol, the secured recommendations prevent all the attacks our model found against SIP. Therefore, we recommend that a secured SIP setup use an overarching trusted authority acting as both the proxy and the distributer of SIP software (for example, Facebook Chat). This does as much as possible to ensure that users are forced to register securely – the only insecure registration must happen outside the protocol.

The SRTP message authentication tag is a prime example of the trade-off between security and packet size. Since the authentication tag can make the packets too large, it prevents this security scheme from always being implemented in practice. Additionally, the SRTP protocol description (RFC 3711) simply recommends message authentication; however, in order to guarantee that all security measures are guaranteed by SRTP are upheld, message authentication is not recommended; it is required.

7 Acknowledgements

We would like to thank Professor John Mitchell and TA Jason Bau from Stanford University. This analysis was prepared as a final project for CS259: Security Modeling and Analysis.

References

[1] Network Working Group. "SIP: Session Initiation Protocol". RFC 3261. http://www.ietf.org/rfc/rfc3261.txt.

[2] "Understanding SIP-Based VoIP". Packetizer. SIP Image. http://www.packetizer.com/ipmc/sip/papers/understanding_sip_voip/sip_call_flow.png.

[3] Network Working Group. "The Secure Real-time Transport Protocol (SRTP)". RFC 3711. http://tools.ietf.org/html/rfc3711.

[4] Mitchell, J.C. Shmatikov, V. Stern, U. (1998). "Finite-State Analysis of SSL 3.0". Proceedings of the 7th USENIX Security Symposium. [link]

[5] Interop Labs. "What is SRTP?" (2007). [link]