Securing Real-Time Communications

Jon Peterson

Transport Area Director, IETF

ITU-T IETF Workshop

Geneva May 2005

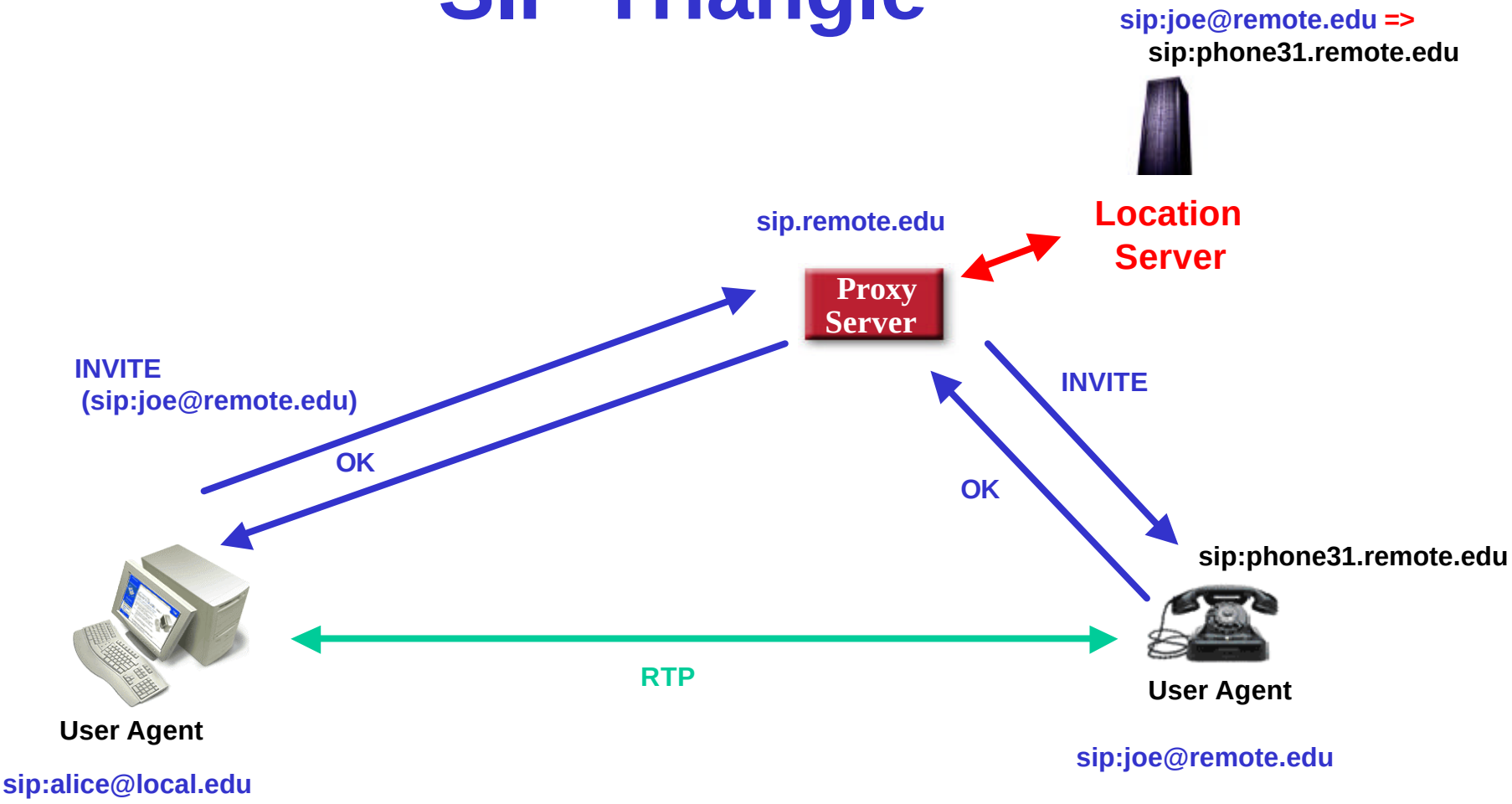
Telephony on the Internet

- On the Internet, telephony is an application
  - Not necessarily a service, no service must be provided
  - A coder can always just write something for talking…
- Like any other application, voice communications operates in accordance with the principles of the Internet
  - Threats against the Internet are applicable
- The IETF has built a suite of real-time communication tools
  - These tools can be used to instantiate telephony
  - Includes SIP, SDP, RTP and related protocols
- These tools are secured with the same tools used to secure other Internet applications

A Crash Course in SIP

- SIP is a protocol that provides two functions
  - Discovery: allows endpoints in the Internet that want to share a session to discover one another
  - Session Management: allows endpoints to exchange session framing messages and other context information about sessions
- SIP is a rendezvous protocol for setting up real-time communications sessions
  - Common applications include voice (telephony), video, presence and instant messaging
- SIP is a control layer used in concert with other protocols that instantiate the session
  - Session Description Protocol (SDP)
  - Real-Time Protocol (RTP)

SIP Triangle



Alice uses DNS to find the deference the Request-URI (which points to the Proxy-Server)

User agent sip:phone31.remote.edu had previously registered itself as a contact for Joe

Four-Tier Security Model for RTC

- Transaction Security Layer
  - Use of TLS and Digest
- Identity Layer
  - SIP header and body signature
- Body Security Layer
  - S/MIME and certificate stores
  - Also encompasses sdescriptions and MIKEY
- Media Security Layer
  - SRTP

- Each layer relies on the one above
- Together, they provide a full security suite for RTC

Security mechanisms in baseline SIP

- SIP is difficult to secure
  - Has many end-to-end and hop-by-hop security requirements
- Digest
  - Based on HTTP Digest (Basic has been deprecated)
  - Can be used to derive authentication properties (based on shared secrets) and some integrity properties
  - Useful for the SIP registration function (likely to share a secret with the registrar)
- TLS
  - Which is of course the new name for SSL
  - Gets canonical properties: integrity, confidentiality, mutual authN, replay protection)
  - Authentication requires certificates
- S/MIME
  - SIP used to support PGP…
  - Authentication also requires certificates

Transaction Security Layer

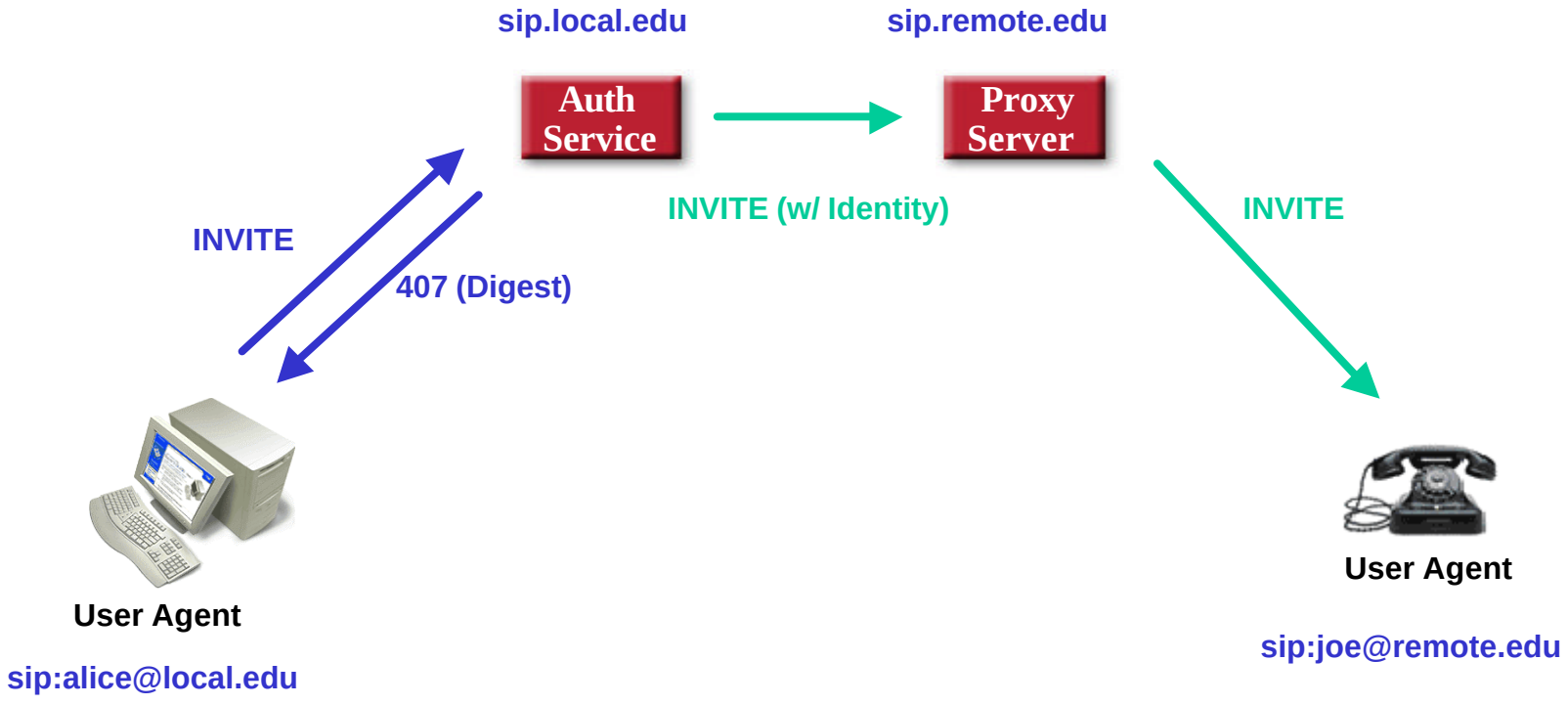
- Digest
  - How a user agent authenticates itself to a proxy server or registrar
  - Can also be leveraged to provide limited integrity (over message body)
  - Must share a secret (i.e. password) with server for this to work
- TLS
  - How a proxy server authenticates itself
  - How a proxy server authenticates itself to another proxy server
  - Also provides integrity and confidentiality of SIP transactions
  - Hop-by-hop only

- Together, they provide building blocks for security
- But, they do not themselves solve all our problems

Identity Layer

- The baseline SIP From header field contains an identity
  - However, most user agents can change the From field arbitrarily (e.g., ‘sip:fake@example.com’)
    - In email this is a critical enabler for spam
  - There is no built-in assurance that the user can legitimately claim this identity
- Thus, the Identity header was developed
  - Identity provides a signature over portions of SIP messages
    - Including the From
  - Can be verified by recipients to determine that the originating domain vouched for this request
  - Prevents certain classes of impersonation, provides domain-based authentication and partial integrity
    - Integrity does include message body integrity
- TLS and Digest are used to validate users before an Identity header is applied

Routing Requests through an Authentication Service



Joe can now inspect the Identity header created by local.edu for Alice

Body Security Layer

- The bodies of SIP requests frequently contain SDP
- Confidentiality of message bodies is thus very significant
  - Can prevent eavesdropping attacks
  - Exchange symmetric media session keys in SDP
    - ‘sdescriptions’ being one example
    - Confidentiality bestowed by S/MIME
- Key discovery for confidentiality is problematic
  - Keys can be exchanged in SIP, thanks to Identity
    - Also helpful to discover keys beforehand
    - Accordingly, a cert retrieval mechanism was developed
- However, support for S/MIME is not universal in SIP
  - MAY strength requirement in RFC3261
- Fortunately, some media security key exchange schemes do not require confidentiality
  - MIKEY key exchange is one example

Media Security Layer

- For telephony applications, RTP is commonly the session protocol
  - Typically secured with SRTP
    - Provides confidentiality for media and source authentication/integrity
- Other session protocols may use other security mechanisms
  - MSRP, for example, has its own security story
  - Also negotiated in SDP
- Lower-layer protocols may be used to secure media
  - If not negotiated in SIP/SDP, application (and hence user) will have no direct assurance of security

Privacy and Security

- In SIP, privacy is the withholding of an identity from potential recipients of a SIP message
  - Private requests can still lead to a dialog, but should not allow the originator of a message to be contacted outside of the dialog by the recipient
- Recipients of requests might also keep information such as their contact addresses private
- Important questions:
  - When is privacy necessary?
    - How much privacy do you need?
  - When can the user agent provide privacy itself?
    - Intermediaries may need to provide some privacy functions
- As we understand identity better, we are coming to a new understanding of privacy requirements

Summary: Securing RTC

- **Requires security at a variety of layers**
  - **Layers have interdependencies**
  - **Taken as a whole, they meet can threats**
- **Security needs to live in the endpoints because that is where the application logic resides**

- **On the Internet, applications cannot rely on the network to just make security happen**