

Internet Engineering Task Force (IETF)
Request for Comments: 7904
Category: Standards Track
ISSN: 2070-1721

C. Jennings
Cisco
B. Lowekamp
Skype
E. Rescorla
RTFM, Inc.
S. Baset
IBM
H. Schulzrinne
Columbia University
T. Schmidt, Ed.
HAW Hamburg
October 2016

A SIP Usage for REsource LOcation And Discovery (RELOAD)

Abstract

This document defines a SIP Usage for REsource LOcation And Discovery (RELOAD). The SIP Usage provides the functionality of a SIP proxy or registrar in a fully distributed system and includes a lookup service for Address of Records (AORs) stored in the overlay. It also defines Globally Routable User Agent URIs (GRUUs) that allow the registrations to map an AOR to a specific node reachable through the overlay. After such initial contact of a Peer, the RELOAD AppAttach method is used to establish a direct connection between nodes through which SIP messages are exchanged.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in [Section 2 of RFC 7841](#).

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at <http://www.rfc-editor.org/info/rfc7904>.

Copyright Notice

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

Table of Contents

1.	Introduction	4
2.	Terminology	6
3.	Registering AORs in the Overlay	6
3.1.	Overview	6
3.2.	Data Structure	7
3.3.	Access Control	9
3.4.	Overlay Configuration Document Extension	10
4.	Looking Up an AOR	11
4.1.	Finding a Route to an AOR	11
4.2.	Resolving an AOR	12
5.	Forming a Direct Connection	12
5.1.	Setting Up a Connection	12
5.2.	Keeping a Connection Alive	13
6.	Using GRUUs	13
7.	SIP-REGISTRATION Kind Definition	14
8.	Security Considerations	14
8.1.	RELOAD-Specific Issues	14
8.2.	SIP-Specific Issues	15
8.2.1.	Fork Explosion	15
8.2.2.	Malicious Retargeting	15
8.2.3.	Misuse of AORs	15
8.2.4.	Privacy Issues	16
9.	IANA Considerations	16
9.1.	Data Kind-ID	16
9.2.	XML Namespace Registration	16
10.	References	16
10.1.	Normative References	16
10.2.	Informative References	18
Appendix A.	Third-Party Registration	19
	Acknowledgments	19
	Authors' Addresses	20

1. Introduction

REsource LOcation And Discovery (RELOAD) [[RFC6940](#)] specifies a peer-to-peer (P2P) signaling protocol for general use on the Internet. This document defines a SIP Usage of RELOAD that allows SIP [[RFC3261](#)] user agents (UAs) to establish peer-to-peer SIP (or SIPS) sessions without the requirement for a permanent proxy or registration servers, e.g., a fully distributed telephony service. This service transparently supports SIP addressing including telephone numbers. In such a network, the RELOAD overlay itself performs the registration and rendezvous functions ordinarily associated with such servers.

The SIP Usage involves two basic functions:

Registration: SIP UAs can use the RELOAD data storage functionality to store a mapping from their Address of Record (AOR) to their Node-ID in the overlay and to retrieve the Node-ID of other UAs.

Rendezvous: Once a SIP UA has identified the Node-ID for an AOR it wishes to call, it can use the RELOAD message routing system to set up a direct connection for exchanging SIP messages.

Mappings are stored in the SipRegistration Resource Record defined in this document. All operations required to perform a SIP registration or rendezvous are standard RELOAD protocol methods.

For example, Bob registers his AOR, "bob@dht.example.com", for his Node-ID "1234". When Alice wants to call Bob, she queries the

overlay for "bob@dht.example.com" and receives Node-ID "1234" in return. She then uses the overlay routing to establish a direct connection with Bob and can directly transmit a standard SIP INVITE. In detail, this works along the following steps:

1. Bob, operating Node-ID "1234", stores a mapping from his AOR to his Node-ID in the overlay by applying a Store request for "bob@dht.example.com -> 1234".
2. Alice, operating Node-ID "5678", decides to call Bob. She retrieves Node-ID "1234" by performing a Fetch request on "bob@dht.example.com".
3. Alice uses the overlay to route an AppAttach message to Bob's Peer (ID "1234"). Bob responds with his own AppAttach and they set up a direct connection, as shown in Figure 1. Note that mutual Interactive Connectivity Establishment (ICE) checks are invoked automatically from the AppAttach message exchange.

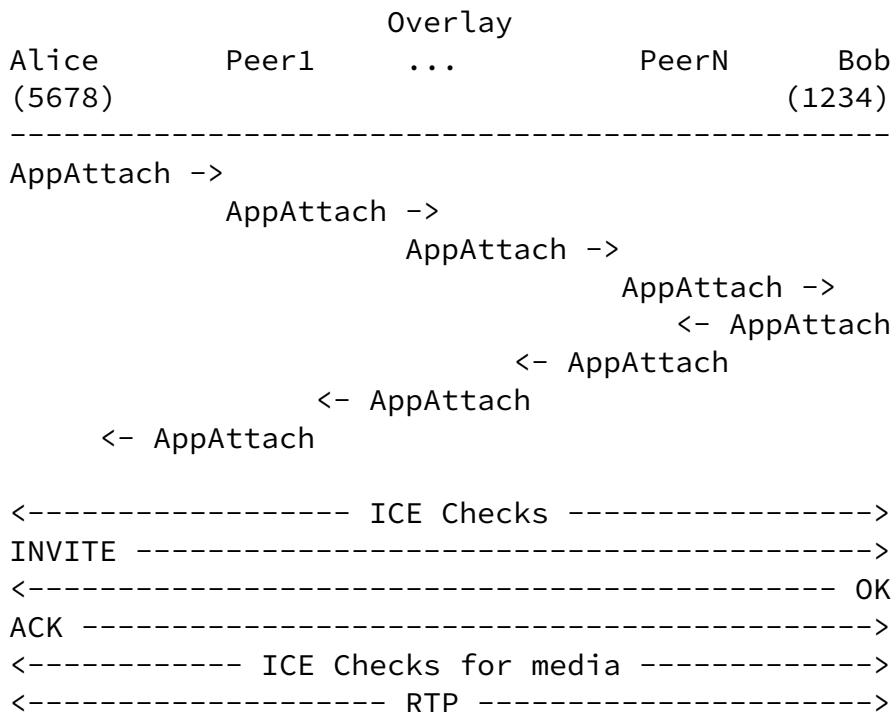


Figure 1: Connection Setup in P2P SIP Using the RELOAD Overlay

It is important to note that the only role of RELOAD in this example is to set up the direct SIP connection between Alice and Bob. As soon as the ICE checks complete and the connection is established, ordinary SIP or SIPS is used. In particular, the establishment of the media channel for a phone call happens via the usual SIP mechanisms, and RELOAD is not involved. Media never traverses the overlay. After the successful exchange of SIP messages, communicating Peers run ICE connectivity checks for media.

In addition to mappings from AORs to Node-IDs, the SIP Usage also allows mappings from AORs to other AORs. This enables an indirection useful for call forwarding. For instance, if Bob wants his phone calls temporarily forwarded to Charlie, he can store the mapping "bob@dht.example.com -> charlie@dht.example.com". When Alice wants to call Bob, she retrieves this mapping and can then fetch Charlie's AOR to retrieve his Node-ID. These mechanisms are described in [Section 3](#).

Alternatively, Globally Routable User Agent URIs (GRUUs) [[RFC5627](#)] can be used for directly accessing Peers. They are handled via a separate mechanism, as described in [Section 6](#).

Concepts used in this document can be extended to include tel URIs [[RFC3966](#)], but this will require further specifications to ensure semantic interoperability of implementations.

The SIP Usage for RELOAD addresses a fully distributed deployment of session-based services among overlay Peers. This RELOAD Usage may be relevant in a variety of environments, including a tightly controlled environment of a single provider that admits parties using AORs with domains from controlled namespace(s) only, or an open, multi-party infrastructure that liberally allows a registration and rendezvous for various or any domain namespace. It is noteworthy in this context that -- in contrast to regular SIP -- domain names play no role in routing to a proxy server. Once connectivity to an overlay is given, the technology allows any name registration, possibly constrained by overlay domain restrictions.

[2](#). Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT",

"SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC 2119](#) [[RFC2119](#)].

We use the terminology and definitions from "Concepts and Terminology for Peer-to-Peer SIP (P2PSIP)" [[RFC7890](#)] and the RELOAD Base Protocol [[RFC6940](#)] extensively in this document.

In addition, terms defined by SIP [[RFC3261](#)] apply to this memo. The term AOR is the SIP "Address of Record" used to identify a user in SIP. For example, "alice@example.com" could be the AOR for Alice. For the purposes of this specification, an AOR is considered not to include the scheme (e.g., sip:), as the AOR needs to match the rfc822Name in the X.509 v3 certificates [[RFC5280](#)]. It is worth noting that SIP and SIPS are distinguished in P2PSIP by the Application-ID.

[3.](#) Registering AORs in the Overlay

[3.1.](#) Overview

In ordinary SIP, a UA registers the user's AOR and its network location with a registrar. In RELOAD, this registrar function is provided by the overlay as a whole. To register its location, a RELOAD peer stores a SipRegistration Resource Record under its own AOR using the SIP-REGISTRATION Kind, which is formally defined in [Section 7](#). Note that the registration lifetime known from the regular SIP REGISTER method is inherited from the lifetime attribute of the basic RELOAD StoredData structure (see [Section 7 in](#) [\[RFC6940\]](#)).

A RELOAD overlay MAY restrict the storage of AORs. Namespaces (i.e., the right-hand side of the AOR) that are supported for registration and lookup can be configured for each RELOAD deployment as described in [Section 3.4](#).

As a simple example, consider Alice with an AOR "alice@dht.example.org" at Node-ID "1234". She might store the mapping "alice@dht.example.org -> 1234" telling anyone who wants to

call her to contact node "1234".

RELOAD peers can store two kinds of SIP mappings,

- o from an AOR to a destination list (a single Node-ID is just a trivial destination list), or
- o from one AOR to another.

The meaning of the first kind of mapping is "in order to contact me, form a connection with this Peer." The meaning of the second kind of mapping is "in order to contact me, dereference this AOR". The latter allows for forwarding. For instance, if Alice wants her calls to be forwarded to her secretary, Sam, she might insert the following mapping, "alice@dht.example.org -> sam@dht.example.org".

[3.2.](#) Data Structure

This section defines the SipRegistration Resource Record as follows:

```
enum {
    sip_registration_uri(1),
    sip_registration_route(2),
    (255)
} SipRegistrationType;

select (SipRegistration.type) {
    case sip_registration_uri:
        opaque          uri<0..2^16-1>;

    case sip_registration_route:
        opaque          contact_prefs<0..2^16-1>;
        Destination    destination_list<3..2^16-1>;

    /* This type can be extended */
} SipRegistrationData;
```

```
struct {
```



```
        SipRegistrationType    type;
        uint16                  length;
        SipRegistrationData     data;
    } SipRegistration;
```

The contents of the SipRegistration Resource Record are:

type

the type of the registration

length

the length of the rest of the PDU

data

the registration data

- o If the registration is of type "sip_registration_uri", then the contents are an opaque string containing the AOR.
- o If the registration is of type "sip_registration_route", then the contents are an opaque string containing the registrant's contact preferences and a destination list for the Peer.

The callee expresses its capabilities within the contact preferences as specified in [\[RFC3840\]](#). It encodes a media feature set comprised of its capabilities as a contact predicate, i.e., a string of feature parameters that appear as part of the Contact header field. Feature parameters are derived from the media feature set syntax of [\[RFC2533\]](#) (see also [\[RFC2738\]](#)) as described in [\[RFC3840\]](#).

This encoding covers all SIP User Agent capabilities, as defined in [\[RFC3840\]](#) and registered in the SIP feature tag registration tree. In particular, a callee can indicate that it prefers contact via a particular SIP scheme -- SIP or SIPS -- by using one of the following contact_prefs attributes:

```
(sip.schemes=SIP)
(sip.schemes=SIPS)
```

RELOAD explicitly supports multiple registrations for a single AOR. The registrations are stored in a dictionary with Node-IDs as the dictionary keys. Consider, for instance, the case where Alice has two Peers:

- o her desk phone (1234)
- o her cell phone (5678)

Alice might store the following in the overlay at resource "alice@dht.example.com":

- o a SipRegistration of type "sip_registration_route" with dictionary key "1234" and value "1234", both referring to Node-IDs
- o a SipRegistration of type "sip_registration_route" with dictionary key "5678" and value "5678"

Note that this structure explicitly allows one Node-ID to forward to another Node-ID. For instance, Alice could set calls to her desk phone to ring at her cell phone by storing a SipRegistration of type "sip_registration_route" with a dictionary key "1234" and a value "5678".

[3.3.](#) Access Control

In order to prevent hijacking or other misuse, registrations are subject to access control rules. Two kinds of restrictions apply:

- o A Store is permitted only for AORs with domain names that fall into the namespaces supported by the RELOAD Overlay Instance.
- o Storing requests are performed according to the USER-NODE-MATCH access control policy of RELOAD.

Before issuing a Store request to the overlay, any Peer SHOULD verify that the AOR of the request is a valid Resource Name with respect to its domain name and the namespaces defined in the overlay configuration document (see [Section 3.4](#)).

Before a Store is permitted, the Storing Peer MUST check that:

- o The AOR of the request is a valid Resource Name with respect to the namespaces defined in the overlay configuration document.
- o The certificate contains a username that is a SIP AOR that hashes

to the Resource-ID it is being stored at.

- o The certificate contains a Node-ID that is the same as the dictionary key it is being stored at.

If any of these checks fail, the request MUST be rejected with an `Error_Forbidden` error.

Note that these rules permit Alice to forward calls to Bob without his permission. However, they do not permit Alice to forward Bob's calls to her. See [Section 8.2.2](#) for additional details.

[3.4.](#) Overlay Configuration Document Extension

The use of a SIP-enabled overlay MAY be restricted to users with AORs from specific domains. When deploying an overlay service, providers can implement such restrictions by defining a set of namespaces for admissible domain names. This section extends the overlay configuration document by defining new elements for patterns that describe a corresponding domain name syntax.

A RELOAD overlay can be configured to accept store requests for any AOR, or to apply domain name restrictions. To apply restrictions, the overlay configuration document needs to contain a `<domain-restrictions>` element. The `<domain-restrictions>` element serves as a container for zero to multiple `<pattern>` sub-elements. A `<pattern>` element MAY be present if the "enable" attribute of its parent element is set to true. Each `<pattern>` element defines a pattern for constructing admissible resource names. It is of type `xsd:string` and interpreted as a regular expression according to "POSIX Extended Regular Expression" (see the specifications in [[IEEE-Posix](#)]). Encoding of the domain name adheres to the restricted ASCII character set without character escaping as defined in [Section 19.1 of \[RFC3261\]](#).

Inclusion of a `<domain-restrictions>` element in an overlay configuration document is OPTIONAL. If the element is not included, the default behavior is to accept any AOR. If the element is included and the "enable" attribute is not set or set to false, the overlay MUST only accept AORs that match the domain name of the overlay. If the element is included and the "enable" attribute is

set to true, the overlay MUST only accept AORs that match patterns specified in the <domain-restrictions> element.

Example of Domain Patterns:

```
dht\.example\.com
.*\.my\.example
```

In this example, any AOR will be accepted that is either of the form <user>@dht.example.com, or ends with the domain "my.example".

The RELAX NG grammar for the AOR Domain Restriction reads:

```
# AOR DOMAIN RESTRICTION URN SUB-NAMESPACE
namespace sip = "urn:ietf:params:xml:ns:p2p:config-base:sip"
# AOR DOMAIN RESTRICTION ELEMENT
Kind-parameter &= element sip:domain-restriction {
    attribute enable { xsd:boolean }
    # PATTERN ELEMENT
    element sip:pattern { xsd:string }*
}?
```

[4.](#) Looking Up an AOR

[4.1.](#) Finding a Route to an AOR

A RELOAD user, member of an overlay, who wishes to call another user with a given AOR SHALL proceed in the following way:

AOR is a GRUU? If the AOR is a GRUU for this overlay, the callee can be contacted directly as described in [Section 6](#).

AOR domain is hosted in overlay? If the domain part of the AOR matches a domain pattern configured in the overlay, the user can continue to resolve the AOR in this overlay. The user MAY choose to query the DNS service records to search for additional support of this domain name.

AOR domain not supported by overlay? If the domain part of the AOR is not supported in the current overlay, the user might query the DNS (or other discovery services at hand) to search for an alternative overlay that services the AOR under request. Alternatively, standard SIP procedures for contacting the callee might be used.

AOR inaccessible? If all of the above contact attempts fail, the call fails.

The procedures described above likewise apply when nodes are simultaneously connected to several overlays.

[4.2.](#) Resolving an AOR

A RELOAD user that has discovered a route to an AOR in the current overlay SHALL execute the following steps:

1. Perform a Fetch for Kind SIP-REGISTRATION at the Resource-ID corresponding to the AOR. This Fetch SHOULD NOT indicate any dictionary keys, so that it will fetch all the stored values.
2. If any of the results of the Fetch are non-GRUU AORs, then repeat step 1 for that AOR.
3. Once only GRUUs and destination lists remain, the Peer removes duplicate destination lists and GRUUs from the list and initiates SIP or SIPS connections to the appropriate Peers as described in the following sections. If there are also external AORs, the Peer follows the appropriate procedure for contacting them as well.

[5.](#) Forming a Direct Connection

[5.1.](#) Setting Up a Connection

Once the Peer has translated the AOR into a set of destination lists, it then uses the overlay to route AppAttach messages to each of those

Peers. The "application" field MUST be either 5060 to indicate SIP or 5061 to indicate SIPS. If certificate-based authentication is in use, the responding Peer MUST present a certificate with a Node-ID matching the terminal entry in the destination list. Otherwise, the connection MUST NOT be used and MUST be closed. Note that it is possible that the Peers already have a RELOAD connection mutually established. This MUST NOT be used for SIP messages unless it is a SIP connection. A previously established SIP connection MAY be used for a new call.

Once the AppAttach succeeds, the Peer sends plain or (D)TLS-encrypted SIP messages over the connection as in normal SIP. A caller MAY choose to contact the callee using SIP or SIPS, but SHOULD follow a preference indicated by the callee in its contact_prefs attribute (see [Section 3.2](#)). A callee MAY choose to listen on both SIP and SIPS ports and accept calls from either SIP scheme, or select a single one. However, a callee that decides to accept SIPS calls only, SHOULD indicate its choice by setting the corresponding attribute in its contact_prefs. It is noteworthy that, according to [\[RFC6940\]](#), all overlay links are built on (D)TLS-secured transport.

SIP messages carry the SIP URIs of actual overlay endpoints (e.g., "sip:alice@dht.example.com") in the Via and Contact headers, while the communication continues via the RELOAD connection. However, a UA can redirect its communication path by setting an alternate Contact header field like in ordinary SIP.

[5.2](#). Keeping a Connection Alive

In many cases, RELOAD connections established from ICE [\[RFC5245\]](#) negotiations will traverse stateful NATs and firewalls. It is the responsibility of the Peers to send messages with a frequency sufficient to maintain the necessary state in these NATs and firewalls and thus keep the connection alive. Keepalives are a mandatory component of ICE (see [Section 10 of \[RFC5245\]](#)) and no further operations are required. Applications that want to assure maintenance of sessions individually need to follow regular SIP means. Accordingly, a SIP Peer MAY apply keep-alive techniques in agreement with its transport binding as defined in [Section 3.5 of](#)

[\[RFC5626\]](#).

6. Using GRUUs

Globally Routable User Agent URIs (GRUUs) [\[RFC5627\]](#) have been designed to allow direct routing to a specific UA instance without the need for dereferencing by a domain-specific SIP proxy function. The concept is transferred to RELOAD overlays as follows. GRUUs in RELOAD are constructed by embedding a base64-encoded destination list in the "gr" URI parameter of the GRUU. The base64 encoding is done with the alphabet specified in Table 1 of [\[RFC4648\]](#) with the exception that "~" is used in place of "=".

Example of a RELOAD GRUU:

```
alice@example.com;gr=MDEyMzQ1Njc4OTAxMjM0NTY3ODk~
```

GRUUs do not require storing data in the Overlay Instance. Rather, when a Peer needs to route a message to a GRUU in the same P2P overlay, it simply uses the destination list and connects to that Peer. Because a GRUU contains a destination list, it can have the same contents as a destination list stored elsewhere in the resource dictionary.

Anonymous GRUUs [\[RFC5767\]](#) are constructed analogously, but require either that the enrollment server issues a different Node-ID for each anonymous GRUU required, or that a destination list be used that includes a Peer that compresses the destination list to stop the Node-ID from being revealed.

7. SIP-REGISTRATION Kind Definition

This section defines the SIP-REGISTRATION Kind.

Name: SIP-REGISTRATION

Kind IDs: The Resource Name for the SIP-REGISTRATION Kind-ID is the AOR of the user as specified in [Section 2](#). The data stored is a SipRegistration, which can contain either another URI or a destination list to the Peer that is acting for the user.

Data Model: The data model for the SIP-REGISTRATION Kind-ID is a dictionary. The dictionary key is the Node-ID of the Storing Peer. This allows each Peer (presumably corresponding to a single device) to store a single route mapping.

Access Control: USER-NODE-MATCH. Note that this matches the SIP AOR against the rfc822Name in the X.509 v3 certificate. The rfc822Name does not include the scheme so that the "sip:" prefix needs to be removed from the SIP AOR before matching. Escaped characters ('%' encoding) in the SIP AOR also need to be decoded prior to matching (see [[RFC3986](#)]).

Data stored under the SIP-REGISTRATION Kind is of type SipRegistration, containing one of two data types:

sip_registration_uri

A URI that the user can be reached at.

sip_registration_route

A destination list that can be used to reach the user's Peer.

[8.](#) Security Considerations

[8.1.](#) RELOAD-Specific Issues

This Usage for RELOAD does not define new protocol elements or operations. Hence, no new threats arrive from message exchanges in RELOAD.

This document introduces an AOR domain restriction function that must be compared against the registration attempt by the Storing Peer. A misconfigured or malicious Peer could cause frequent rejects of illegitimate storing requests. However, domain name control relies on a lightweight pattern matching and can be processed prior to

validating certificates. Hence, no extra burden is introduced for RELOAD peers beyond loads already present in the base protocol.

[8.2.](#) SIP-Specific Issues

[8.2.1.](#) Fork Explosion

Because SIP includes a forking capability (the ability to retarget to multiple recipients), fork bombs (i.e., attacks using SIP forking to amplify the effect on the intended victims) are a potential DoS concern. However, in the SIP Usage of RELOAD, fork bombs are a much lower concern than in a conventional SIP Proxy infrastructure, because the calling party is involved in each retargeting event. It can therefore directly measure the number of forks and throttle at some reasonable number.

[8.2.2.](#) Malicious Retargeting

To launch a DoS attack, the owner of a popular AOR could retarget all calls to the victim. This attack is common to SIP and is difficult to ameliorate without requiring the target of a SIP registration to authorize all stores. The overhead of that requirement would be excessive and, in addition, there are good use cases for retargeting to a Peer without its explicit cooperation.

[8.2.3.](#) Misuse of AORs

A RELOAD overlay and enrollment service that liberally accepts registrations for AORs of domain names unrelated to the overlay instance and without further authorization could store presence state for AORs without the consent of the owner of the AOR. An attacker could hijack names, register a bogus presence, and attract calls dedicated to a victim that resides within or outside the Overlay Instance.

A hijacking of AORs can be mitigated by restricting the name spaces admissible in the Overlay Instance, or by additional verification actions of the enrollment service. To prevent an (exclusive) routing to a bogus registration, a caller can in addition query the DNS (or other discovery services at hand), search for an alternative presence of the callee in another overlay or a SIP infrastructure using [\[RFC3263\]](#) for name resolution.

8.2.4. Privacy Issues

All RELOAD SIP registration data is visible to all nodes in the overlay. Location privacy can be gained from using anonymous GRUUs. Methods of providing anonymity or deploying pseudonyms exist, but are beyond the scope of this document.

9. IANA Considerations

9.1. Data Kind-ID

IANA has registered the following code point in the "RELOAD Data Kind-ID" Registry (cf., [[RFC6940](#)]) to represent the SIP-REGISTRATION Kind, as described in [Section 7](#).

Kind	Kind-ID	Reference
SIP-REGISTRATION	0x1	RFC 7904

9.2. XML Namespace Registration

This document registers the following URI for the config XML namespace in the IETF XML registry defined in [[RFC3688](#)]:

URI: urn:ietf:params:xml:ns:p2p:config-base:sip

Registrant Contact: The IESG

XML: N/A; the requested URI is an XML namespace

10. References

10.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.
- [RFC6940] Jennings, C., Lowekamp, B., Ed., Rescorla, E., Baset, S., and H. Schulzrinne, "REsource LOcation And Discovery (RELOAD) Base Protocol", [RFC 6940](#), DOI 10.17487/RFC6940, January 2014, <<http://www.rfc-editor.org/info/rfc6940>>.

[RFC 7904](#)

RELOAD SIP Usage

October 2016

- [RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", [RFC 3261](#), DOI 10.17487/RFC3261, June 2002, <<http://www.rfc-editor.org/info/rfc3261>>.
- [RFC2533] Klyne, G., "A Syntax for Describing Media Feature Sets", [RFC 2533](#), DOI 10.17487/RFC2533, March 1999, <<http://www.rfc-editor.org/info/rfc2533>>.
- [RFC2738] Klyne, G., "Corrections to "A Syntax for Describing Media Feature Sets"", [RFC 2738](#), DOI 10.17487/RFC2738, December 1999, <<http://www.rfc-editor.org/info/rfc2738>>.
- [RFC3688] Mealling, M., "The IETF XML Registry", [BCP 81](#), [RFC 3688](#), DOI 10.17487/RFC3688, January 2004, <<http://www.rfc-editor.org/info/rfc3688>>.
- [RFC3840] Rosenberg, J., Schulzrinne, H., and P. Kyzivat, "Indicating User Agent Capabilities in the Session Initiation Protocol (SIP)", [RFC 3840](#), DOI 10.17487/RFC3840, August 2004, <<http://www.rfc-editor.org/info/rfc3840>>.
- [RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, [RFC 3986](#), DOI 10.17487/RFC3986, January 2005, <<http://www.rfc-editor.org/info/rfc3986>>.
- [RFC4648] Josefsson, S., "The Base16, Base32, and Base64 Data Encodings", [RFC 4648](#), DOI 10.17487/RFC4648, October 2006, <<http://www.rfc-editor.org/info/rfc4648>>.
- [RFC5245] Rosenberg, J., "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols", [RFC 5245](#), DOI 10.17487/RFC5245, April 2010, <<http://www.rfc-editor.org/info/rfc5245>>.
- [RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S.,

Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", [RFC 5280](#), DOI 10.17487/RFC5280, May 2008, <<http://www.rfc-editor.org/info/rfc5280>>.

- [RFC5626] Jennings, C., Ed., Mahy, R., Ed., and F. Audet, Ed., "Managing Client-Initiated Connections in the Session Initiation Protocol (SIP)", [RFC 5626](#), DOI 10.17487/RFC5626, October 2009, <<http://www.rfc-editor.org/info/rfc5626>>.
- [RFC5627] Rosenberg, J., "Obtaining and Using Globally Routable User Agent URIs (GRUUs) in the Session Initiation Protocol (SIP)", [RFC 5627](#), DOI 10.17487/RFC5627, October 2009, <<http://www.rfc-editor.org/info/rfc5627>>.
- [IEEE-Posix] IEEE, "International Standard - Information technology Portable Operating System Interface (POSIX) Base Specifications, Issue 7", ISO/IEC/IEEE 9945:2009, DOI 10.1109/IEEESTD.2009.5393893, September 2009.

[10.2.](#) Informative References

- [RFC3263] Rosenberg, J. and H. Schulzrinne, "Session Initiation Protocol (SIP): Locating SIP Servers", [RFC 3263](#), DOI 10.17487/RFC3263, June 2002, <<http://www.rfc-editor.org/info/rfc3263>>.
- [RFC3966] Schulzrinne, H., "The tel URI for Telephone Numbers", [RFC 3966](#), DOI 10.17487/RFC3966, December 2004, <<http://www.rfc-editor.org/info/rfc3966>>.
- [RFC7890] Bryan, D., Matthews, P., Shim, E., Willis, D., and S. Dawkins, "Concepts and Terminology for Peer-to-Peer SIP (P2PSIP)", [RFC 7890](#), DOI 10.17487/RFC7890, June 2016, <<http://www.rfc-editor.org/info/rfc7890>>.

[RFC5767] Munakata, M., Schubert, S., and T. Ohba, "User-Agent-Driven Privacy Mechanism for SIP", [RFC 5767](#), DOI 10.17487/RFC5767, April 2010, <<http://www.rfc-editor.org/info/rfc5767>>.

[SHARE] Knauf, A., Schmidt, T., Hege, G., and M. Waehlich, "A Usage for Shared Resources in RELOAD (ShaRe)", Work in Progress, [draft-ietf-p2psip-share-08](#), March 2016.

[Appendix A](#). Third-Party Registration

Non-peer-to-peer SIP defines third-party registration (e.g., an assistant acting for a manager or a changing set of users registering under a role-based AOR) in [Section 10.2 of \[RFC3261\]](#). This is a REGISTER that uses the URI of the third party in its From header and cannot be translated directly into a P2PSIP registration because only the owner of the certificate can store a SIP-REGISTRATION in a RELOAD overlay.

Third-party registration can be implemented by using the extended access control mechanism USER-CHAIN-ACL defined in [\[SHARE\]](#). Creating a new Kind "SIP-3P-REGISTRATION" that is ruled by USER-CHAIN-ACL allows the owner of the certificate to delegate the right for registration to individual third parties. This way, the SIP third-party registration functionality can be regained without weakening the security controls of RELOAD.

Acknowledgments

This document was generated in parts from initial drafts and discussions in the early specification phase of the P2PSIP base protocol. We gratefully acknowledge the significant contributions made by (in alphabetical order) David A. Bryan, James Deverick, Marcin Matuszewski, Jonathan Rosenberg, and Marcia Zangrilli.

Additional thanks go to all those who helped with ideas, discussions, and reviews, in particular (in alphabetical order) Roland Bless, Michael Chen, Alissa Cooper, Marc Petit-Huguenin, Brian Rosen, Meral Shirazipour, and Matthias Waehlich.

Authors' Addresses

Cullen Jennings
Cisco
170 West Tasman Drive
MS: SJC-21/2
San Jose, CA 95134
United States of America
Phone: +1 408 421-9990
Email: fluffy@cisco.com

Bruce B. Lowekamp
Skype
Palo Alto, CA
United States of America
Email: bbl@lowekamp.net

Eric Rescorla
RTFM, Inc.

2064 Edgewood Drive
Palo Alto, CA 94303
United States of America
Phone: +1 650 678 2350
Email: ekr@rtfm.com

Salman A. Baset
IBM T. J. Watson Research Center
1101 Kitchawan Road
Yorktown Heights, NY 10598
United States of America
Email: sabaset@us.ibm.com

Henning Schulzrinne
Columbia University
1214 Amsterdam Avenue
New York, NY 10027
United States of America
Email: hgs@cs.columbia.edu

Thomas C. Schmidt (editor)
HAW Hamburg
Berliner Tor 7
Hamburg 20099
Germany
Email: t.schmidt@haw-hamburg.de