Guidelines for the use of the SIPS URI Scheme in the Session Initiation Protocol (SIP)

draft-audet-sip-sips-guidelines-04

Status of this Memo

Copyright Notice

Abstract

This document updates RFC3261 by providing clarifications, guidelines and new requirements concerning the use of SIPS URI Scheme in the Session Initiation Protocol (SIP).

Table of Contents

1. Introduction

The meaning and usage of the SIPS URI scheme and of TLS is at best underspecified in SIP [RFC3261] and has been the source of confusion for implementors.

This document provides clarifications, guidelines and new requirements concerning the use of the SIPS URI scheme.

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 [RFC2119].

3. Meaning of SIPS

[RFC3261]/19.1 describes a SIPS URI as follows:

> A SIPS URI specifies that the resource be contacted securely. This means, in particular, that TLS is to be used between the UAC and the domain that owns the URI. From there, secure communications are used to reach the user, where the specific security mechanism depends on the policy of the domain.

Section 26.2.2 re-iterates it, with regards to Request-URIs:

> When used as the Request-URI of a request, the SIPS scheme signifies that each hop over which the request is forwarded, until the request reaches the SIP entity responsible for the domain portion of the Request-URI, must be secured with TLS; once it reaches the domain in question it is handled in accordance with local security and routing policy, quite possibly using TLS for any last hop to a UAS. When used by the originator of a request (as would be the case if they employed a SIPS URI as the address-of-record of the target), SIPS dictates that the entire request path to the target domain be so secured.

Let's take the classic SIP trapezoid to explain the meaning of a sips:b@B URI. Instead of using real domain names like example.com and example.net, logical names like "A" and "B" are used, for clarity.

```
        ..........................         ...........................
        .                        .         .                         .
        .              +-------+ .         . +-------+               .
        .              |       | .         . |       |               .
        .              | Proxy |-----TLS---- | Proxy |               .
        .              |   A   | .         . |  B    |               .
        .              |       | .         . |       |               .
        .            / +-------+ .         . +-------+ \             .
        .           /            .         .            \            .
        .          /             .         .             \           .
        .        TLS             .         .        Policy-based     .
        .        /               .         .               \         .
        .       /                .         .                \        .
        .      /                 .         .                 \       .
        .   +-------+            .         .              +-------+  .
        .   |       |            .         .              |       |  .
        .   | UA a  |            .         .              | UA b  |  .
        .   |       |            .         .              |       |  .
        .   +-------+            .         .              +-------+  .
        .             Domain A   .         .   Domain B              .
        ..........................         ...........................
```

SIP trapezoid

In this case, if a@A is sending a request to sips:b@B, the following will apply:

- o TLS MUST be used between UA a@A and Proxy A
- o TLS MUST be used between Proxy A and Proxy B
- o TLS MAY be used between Proxy B and UA b@B, depending on local policy.

One may then wonder why TLS is mandatory between UA a@A and Proxy A but not between Proxy B and UA b@B. The main reason is that [RFC3261] [RFC3261] was written before [I-D.ietf-sip-outbound]. At that time, it was recognized that in many practical deployments, Proxy B may not be able to establish a TLS connection with UA b because only Proxy B would have a certificate to provide and UA b would have none. Since UA b would be the TLS Server, it would then not be able to accept the incoming TLS connection. The consequence is that an [RFC3261]-compliant UAS b, while it may not need to support TLS for incoming requests, will nevertheless have to support TLS for outgoing requests as it takes the UAC role. Contrary to what many believe erroneously, the last-hop exception was not created to allow for using a SIPS URI to address a UAS that does not support TLS : the last-hop exception was an attempt to allow for incoming requests TLS when a SIPS URI is used, and it does not apply to outgoing requests. The rationale for this was somewhat flawed, and since then, [I-D.ietf-sip-outbound] has provided a more satisfactory solution to this problem.

OPEN ISSUE: Many people have expressed the opinion that the "last hop exception" rule should be deprecated, and nobody so far objected to it. The author of this draft is one who favors deprecating the "last hop exception" rule. This is the single biggest open issue in this draft. If so, should it be done within this specification, or in a different specification? The ramainder of this draft assumes that the "last hop exception" is NOT deprecated.

Furthemore, consider the problem of using SIPS inside a dialog. If a@A sends a request to b@B using a SIPS Request-URI, according to RFC 3261/8.1.1.8, then the contact MUST contain a SIPS URI as well. This means that b@B, upon sending a new Request within the dialog (e.g., a BYE or re-INVITE), will have to use a SIPS URI. If there is no Record-Route entry, or if the last Record-Route entry consist of a SIPS URI, this implies that b@B must understand SIPS in the first place, and must also support TLS. If the last Record-Route entry however is a sip URI, then b would be able to send requests without using TLS. In either case however, the Request-URI would be a SIPS URI.

The SIPS scheme implies transitive trust. Obviously, there is nothing that prevents a proxy to cheat (see 26.4.4/[RFC3261]). While SIPS is useful to request that a resource be contacted securely, it is not useful as an indication that a resource was in fact contacted security. Therefore, it is not appropriate to infer that because an incoming request had a Request-URI (or To header) containing a SIPS URI, that it necessarily garantees that the request was in fact transmitted securely on each hop. Some have been tempted to believe that the SIPS scheme was equivalent to an HTTPS scheme in the sense that one could provide a visual indication to a user (e.g., a padlock icon) to the effect that the session is secured. This is obviously not the case, and one must therefore be careful not to oversell the meaning of a SIPS URI. There is currently no mechanism to provide an indication of end-to-end security for SIP. Other mechanisms may provide a more concrete indication of some level of security. For example, SIP Identity [RFC4474] describes an authenticated identity mechanism and a domain-to-domain integrity protection mechanism.

4. Routing

This specification mandates that SIP and SIPS URIs that are identical except for the scheme itself (e.g., sip:alice@example.com and sips:alice@example.com) MUST refer to the same resource. This requirement is implicit in [RFC3261]/19.1 which states that "Any resource described by a SIP URI can be "upgraded" to a SIPS URI by just changing the scheme, if it is desired to communicate with that

resource securily". Note that this does not mean that the SIPS URI will necessarily be reachable, in particular, if the proxy can not establish a secure connection to a client or another proxy. This does not suggest that proxies should arbitrarily "upgrade" SIP URIs to SIPS URIs. Rather, it means that when a proxy has a legitimate reason to do so, it MAY upgrade a SIP URI to a SIPS URI. An example of such a case is when the Contact binding to an AOR is a SIPS URI and a request was addressed to a SIP AOR, the proxy will "upgrade" the Request-URI to the SIPS Contact and forward the request to that address, as illustrated by message F13 in Section 9.2.

Although not mandated specifically in [RFC3261], the implication is that a resource described by a SIPS URI can not be "downgraded" to a SIP URI by just changing the scheme, unless it is the "last hop exception" described in Section 3. This specification mandates that a resource described by a SIPS URI MUST NOT be "downgraded" to a SIP URI by changing the scheme, or by sending the associated request over a non secure link, except for cases where the last hap when the "last-hop exception" rule is in effect (in which case the Request-URI would be replaced by a SIP URI).

For example, the sip:bob@example.com and sips:bob@example.com AORs MUST refer to the same user "Bob" in domain "example.com": the first URI is the SIP version, and the second one is the SIPS version. From the point of view of routing, requests to either sip:bob@example.com and sips:bob@example.com are treated the same way. Location services are therefore free to map from SIP to SIPS URIs as appropriate (see 26.4.4/[RFC3261]). When Bob registers, it therefore does not really matter if he is using a SIP or a SIPS AOR, since they both refer to the same user. It is the association of the AOR with the Contact in the REGISTER that will determine the reachability of the AOR. At first glance, section 19.1.4/[RFC3261] seems to contradict this idea by stating that a SIP and a SIPS URI are never equivalent. Specifically, it says that they are never equivalent for the purpose of comparing bindings in Contact URIs in REGISTER requests. The key point is that this statement applies to the Contact bindings in a registration: it is the association of the Contact with the AoR that will determine if the user is reachable or not with a SIPS URI.

Consider this example. If Bob registers with a SIPS contact (e.g., sips:bob@bobphone.example.com), the registar and the location service then knows that Bob (bob@example.com) is reachable at sips:bob@bobphone.example.com. If a request is sent to sips:bob@bobphone.example.com, Bob's proxy will route it to Bob at sips:bob@bobphone.example.com. If a request is sent to sip:bob@bobphone.example.com, Bob's proxy will also route it to Bob at sips:bob@bobphone.example.com (because of the "upgrade" scenario described above). However, if Bob had registered instead with a SIP

Contact (e.g., sip:bob@bobphone.example.com), then a request to sips:bob@example.com would not be routed to Bob, since there is no SIPS contact for Bob, and "downgrades" from SIPS to SIP are not allowed.

See Section 9 for illustrative call flows.

Since upgrading from SIP to SIPS is allowed it other circumstances (e.g., a user "guessing" a SIPS AOR from a SIP AOR on a business card), it is quite possible that a request will be rejected with response code 416 (either because TLS or SIPS is not supported). When 416 is received, the request MAY be re-attempted with a SIP URI, but the user SHOULD be informed.

Although "downgrading" from SIPS to SIP is disallowed, it is possible that a redirect server or UAS sends a 3XX response to a request to a SIPS URI with a Contact containing a SIP URI. Section 8.1.3.4/ [RFC3261] recommends that if the UAC decide to recurse to the SIP URI, it SHOULD inform the user. When a proxy is handling the 3XX, it can obviously not indicate anything to the user that it is being redirected from SIPS to SIP: therefore, proxies that conform to this specification MUST forwards the 3XX to the UAC instead of recursing, in order to allow for the UAC to take the appropriate action.

OPEN ISSUE: Should forwarding the 3XX to the UAC be a RECOMMENDED strength instead? If so, what would be good qualifiers for not doing so?

4.1. Detection of end-to-end security

The presence of a SIPS Request-URI does not necessarily indicate that the request was sent end-to-end securely. As described in 26.4.4/ [RFC3261], a proxy may legitimately retarget a request from SIP to SIPS. Therefore, a UAS MUST NOT assume on the basis of the Request-URI alone that SIPS was used for the entire request path. An example of a case where a proxy legitimally retargets from SIP to SIPS shown in Section 9.2.

So how does a UAS know if the SIPS was used for the entire request path to secure the request end-to-end? Effectively, the UAS can not know for sure. However, 26.4.4/[RFC3261] recommends how a UAS may make some checks to validate the security. Here is a summary of a potential algorithm:

- o If the URI in the To header is a SIPS URI and the Request-URI is a SIPS, then the dialog is "tentatively" secure. See below.

o If the URI in the To header is SIPS and the Request-URI is SIP and there is some other security mechanism (e.g., IPsec) securing the last hop, then the dialog may be "tentatively" secure. See below.
o Otherwise the dialog is insecure.
o If the dialog was "tentatively" secure, it is RECOMMENDED that the security be checked by checking both the Via headers and the Record-route, as described in 26.4.4/[RFC3261].

Again, it should be restated that all the checking may be circumvented by any proxy on the path that does not follow the rules and recommendations of this document and of [RFC3261].

Proxies MAY have their own policy regarding routing of requests to SIP or SIPS URIs. For example, a proxy in some environment may be configured to only route SIPS. Some proxies MAY be configured to detect uncompliancies and reject unsecure requests. For example, it could inspect Request-URIs, Path, Record-Route, To, From, Contacts and Via headers to enforce SIPS.

26.4.4/[RFC3261] also explains that S/MIME may also be used by the originating UAC to ensure that the original form of the To header field is carried end-to-end. While not specifically mentioned in 26.4.4/[RFC3261], this is meant to imply that [RFC3893] would be used to "tunnel" important headers (such as To and From) in an encrypted and signed S/MIME body, replicating the information in the SIP message, and allowing the UAS to validate the content of those important headers. While this approach is certainly legal, another approach is to use the SIP Identity mechanism defined in [RFC4474]. SIP Identity creates a signed identity digest which includes, amongst other things, the AOR of the sender (from the From header) and the AOR of the original destination (from the To header). It is RECOMMENDED that a UAC use the mechanism in [RFC4474] instead of the one defined in [RFC3893].

4.2. Loose and strict routing

Using strict or loose routing has a huge impact on sips and TLS. Some of the advantages of using loose routing have been discussed in Section 3, regarding mid-dialog requests.

When a proxy inserts a Record-Route entry, it must take care in using the proper scheme so that furher in-dialog requests are sent to the proper URI. This is particularly important when a proxy changes the transport from TLS to non-TLS of an incoming request (when the last hop exception rule is used) or from non-TLS to TLS (when "upgrading"). [RFC3261] sections 16.6 and 16.7 describe how this can be done by having the proxy modifying the Record-Route in the response. However, as described in [RFC3608], this is problematic.

This specification therefore adopts the procedure of [RFC3608], and instead of following the procedure in [RFC3261], proxies that are inserting Record-Route or Path header field URIs MUST record not one but two route URIs when processing the request. The first value recorded indicates the receiving interface, and the second indicates the sending interface. When processing the response, no modification of the recorded route is required. This optimization provides for fully invertible routes that can be effectively used in construction of service routes. It is illustrated as follows:

```
    UA a                           Proxy                          UA b

      =======REQUEST/TLS==========>-------REQUEST/non-TLS------>
                                     Record-Route: <sip:p;lr>,
                                                   <sips:p;lr>

      <======Response/TLS=========<-------Response/non-TLS------
       Record-Route: <sip:p;lr>      Record-Route: <sip:p;lr>,
                     <sips:p;lr>                   <sips:p;lr>
```

Record routing from SIPS to SIP

Similarly, if a proxy receives a request on npn-TLS and forwards it over TLS, then the Record-Route entry it inserts MUST be a sips URI. A response to the Request will then be sent over TLS, and forwarded back on non-TLS, with the proxy rewriting the Record-Route to be a sip URI.This is illustrated as follows:

```
    UA a                           Proxy                          UA b

      -------REQUEST/non-TLS------>=======REQUEST/TLS==========>
                                     Record-Route: <sips:p;lr>,
                                                   <sip:p;lr>

      <------Response/non-TLS-----<=======Response/TLS==========
       Record-Route: <sips:p;lr>      Record-Route: <sips:p;lr>,
                     <sip:p;lr>                    <sip:p;lr>
```

Record routing from SIP to SIPS

Note that the same rules apply to the Path Header [RFC3327].

When a UAC is using a Service-Route (e.g., as in [RFC3608]), and sending a request to a SIPS Request-URI, it MUST ensure that the Route header URIs it includes are all SIPS URIs. If the Service route included SIP URI, the UAC MUST upgrade the SIP URIs to SIPS URIs simply by changing the scheme from "sip" to "sips" before sending the request. Note that this allows for configuring or

discovering one Service Route with all SIP URIs and allowing sending requests to both SIP and SIPS URIs.

5. Registration

This section describes the registration procedures of SIPS versus SIP Contacts that follows from the discussion in Section 4.

The USC registers either a SIPS or a SIP AOR. From a routing perspective, it does not matter which one is used for registration as they identify the same resource.

If all the Contacts are SIPS, a SIPS AOR MUST also be used by the UAC. If at least one of the Contacts is SIP or is neither SIP nor SIPS (e.g., mailto, tel, http, https), a SIP AOR MUST also be used by the UAC. However, the UAS (the Registrar), MUST treat the SIP and SIPS schemes of the AOR the same way (i.e., it MUST NOT care if it is SIP or SIPS). Those are mechanical rules with no influence on routing.

Furthermore, it is a matter of local policy for a UA to accept incoming requests addressed to a URI scheme that does not correspond to what it used for registration. For example, a UA with a policy of "always secure" MUST address the Registrar using a SIPS Request-URI over TLS, MUST register with a SIPS Contact, and must NOT accept requests addressed to a SIP Request-URI. A UA with a policy of "best-effort security" MUST address the Registrar using a SIPS Request-URI over TLS, MUST register with a SIPS Contact, and MUST accept requests addressed to either SIP or SIPS Request-URIs. A UA with a policy of "No security" MUST address the Registrar using a SIP Request-URI, MUST NOT use TLS, MUST register with a SIP AOR and SIP Contact, and MUST accept requests addressed only to a SIP Request-URI.

If proxies (such as outbound proxies) are present in the path between the UA and the registrar, they SHOULD insert the Path header [RFC3327].

A registrar MUST only accept a binding to a SIPS Contact if all the appropriate URIs are of the SIPS scheme: i.e., the Request-URI, the AOR (i.e., To header), the From header, the Contacts and all the Path headers. If the URIs are not of the proper SIPS scheme, it MUST reject the REGISTER with a 403 "Forbidden".

OPEN ISSUE: Is 403 "Forbidden" the right error code? Should there be additional information for specific problems? For example, if one of the path headers is wrong?

The usage of the "transport" URI parameter in Contacts in registration is of dubious usefulnes. The assumption is that a UAC may choose one transport for the registration itself, and a different transport for receiving requests. Using the transport URI parameters also results in some complex problems. For example, should all the transports be listed as separate contacts (e.g, udp, tcp, sctp, tls over tcp, tls over sctp)? If so, there is no way to signal tls over sctp defined yet. Furthermore, how should they be prioritized using a q-value? If so, it is possible that certain proxies will interpret this as a forking scenario and they might decide to send one incoming request per transport! Another issue is what happens if a UAC fetches bindings by sending an empty REGISTER message. Would the proxy respond with one or all the possible transport? All this would generate unwarranted complexity.

It is therefore RECOMMENDED that UACs do not use any transport URI parameters in Contacts in REGISTER.

For backward compatibility, a registrar MUST accept a REGISTER message with a transport URI parameter in the Contact. It is RECOMMENDED that a registrar ignores that parameter, i.e., that it will not influence routing.

However, a registrar MUST record the scheme of the Contact.

6. SIPS in a Dialog

There MUST be only one Contact in any request resulting in the establishment of a dialog (e.g., INVITE, SUBSCRIBE, REFER). As mandated by 8.1.1.8/[RFC3261], if the Request-URI (or top Route header field) contains a SIPS URI, the Contact header MUST be a SIPS URI as well. This poses a very significant problem if the topmost Record-Route entry is not a SIP URI since because the remote UAS does not support SIPS, it will not be able to send a mid-dialog request to the client.

In the response, the Contact header MUST also include a SIPS URI if the Request-URI contained a SIPS URI or if the topmost Record-Route header contained a SIPS URI or if the Contact header contained one and there was no Record-Route header.

If a UAS does not support SIPS, it MUST reject a request to a SIPS Request-URI with response code 416 "Unsupported URI scheme". Upon

receiving a 416 a UAC SHOULD NOT re-attempt the request by automatically replacing the SIPS scheme with a SIP scheme. If the UAC does re-attempt the call with a SIP URI, it SHOULD inform to the user that the security level is downgraded.

If a UAS does not support SIP, it MUST reject a request to a SIP Request-URI with response code 416 "Unsupported URI scheme". Upon receiving a 416 a UAC SHOULD re-attempt the request by automatically replacing the SIP scheme with a SIPS scheme.

If the Request-URI is a SIP URI, then the UAC needs to be careful about what to use in the Contact (in case Record-Route is not used for this hop). If the Contact was a SIPS URI, it would mean that it would only accept mid-dialog requests that are over secure transport end-to-end. Since the Request-URI is in this case a SIP URI, it is quite possible that the UA sending a request to that URI may not be able to send requests to SIPS URIs. It is therefore RECOMMENDED that in this case, the Contact be a SIP URI, even if the request is sent over a secure transport (e.g., the first hop could be re-using a TLS connection to the proxy as would be the case with [I-D.ietf-sip-outbound]).

When a target refresh occurs within a dialog (e.g., re-INVITE, UPDATE), unless there is a need to change it, the UAC SHOULD include a Contact header with a SIPS URI if the original request used a SIPS Request-URI.

OPEN ISSUE: Handling of annomalies are not very well defined in [RFC3261]. What if a UAS receives a SIP Contact replacing a SIPS contact in a target refresh? Should the UAC tear down the dialog if it can not cope with the unexpected response?

7. Usage of tls transport parameter and TLS Via parameter

26.2.2/[RFC3261] makes it clear that the use of the "transport=tls" URI transport parameter in SIPS or SIP URIs has been deprecated:

> Note that in the SIPS URI scheme, transport is independent of TLS, and thus "sips:alice@atlanta.com;transport=tcp" and "sips:alice@atlanta.com;transport=sctp" are both valid (although note that UDP is not a valid transport for SIPS). The use of "transport=tls" has consequently been deprecated, partly because it was specific to a single hop of the request. This is a change since RFC 2543.
> Users that distribute a SIPS URI as an address-of-record may elect to operate devices that refuse requests over insecure transports.

However, the "tls" parameter has not been eliminated from the ABNF in 25/[RFC3261], and 26.2.1/[RFC3261] has a vague reference to it. This has been a source of confusion. Those omissions are errors in [RFC3261].

NOTE: This needs to be in corrected in [RFC3261].

This specification mandates that the "transport=tls" parameter MUST NOT be used.

However, for backward compatibility, if a "transport=tls" parameter is received, it SHOULD be interpreted as per the following guidelines:

- o 16.7/[RFC3261] states the transport parameter (e.g., with tcp or udp) SHOULD NOT be used in Record-Route unless it has knowledge that the next upstream element that will be in the path of subsequent supports this transport. Generally, it is RECOMMENDED that the transport parameter never be used in a Record-Route, Route or Path header. Since the transport=tls URI parameter has been deprecated, it MUST NOT be used in Route, Record-Route or Path headers, and MUST be ignored.
- o In a Contact in a dialog, it MAY be interpreted as a request to send incoming mid-dialog requests using TLS. Note that this would only have a significance if [I-D.ietf-sip-outbound] and Record-Route are not used, and if that URI is nevertheless reachable with TLS which is extremely unlikely. If it was the case that it was reachable with TLS, say because there is an active TLS connection, then that connection could be re-used anyways, regardless of the presense of the transport parameter. It is RECOMMENDED that the "transport=tls" parameter be ignored by the UAS.
- o In a Contact in a REGISTER, it tells the registrar that the UAC is reachable through TLS. If the registrar and proxy are co-located, and are the proxy of that UAC, it tells what is already known because the request was sent over TLS (i.e., that it is reachable using TLS), and is therefore redundant. If the registrar is not co-located with the proxy, then it is useless because transport=tls is hop-by-hop and therefore not applicable in this case. The transport=tls parameter MUST therefore be ignored.
- o In a Request-URI, the transport parameter is problematic. On the last hop, it is useless because the transport is evident. Before being resolved to the last hop (with loose routing), it is not clear what it would mean (hop-by-hop?). A proxy MUST ignore the "transport=tls" parameter in a Request-URI.
- o In a Contact in a 3XX response, it would essentially mean a request to attempt to re-send the request, using TLS transport. Since the transport=tls parameter only has local significance, it will only be successful if the 3XX is recursed by the last hop.

It MAY be ignored by the recursing entity, or the recursing entity MAY re-attempt the request using TLS transport.

For Via headers, the following transport "UDP", "TCP", "TLS", "SCTP", and "TLS-SCTP" [RFC4168] are supported.

8. GRUU

GRUU [I-D.ietf-sip-gruu] specifies that when a GRUU is assigned to an instance ID/AOR pair, both SIP and SIPS GRUUs will be assigned. It also specificies that when a GRUU is obtained through registration, if the To header in the REGISTER request contains a SIP URI, the SIP version of the GRUU is returned. If the To header filed in the REGISTER request contains a SIPS URI, the SIPS version of the GRUU is returned. GRUU therefore follows the same logic as the one described in Section 5.

OPEN ISSUE How should the UAC react if the returned GRUU is SIP but the To was SIPS?

OPEN ISSUE How should the UAC react if the returned GRUU is SIPS but the To was SIP?

9. Call Flows

In the following examples, Bob has two clients, one is a SIP PC client running on his computer, and the other one is a SIP Phone. The PC client does not support SIPS (and does not support TLS either) and consequently only registers with a SIP address. The SIP phone however does support SIPS and TLS, and consequently registers with a SIPS address. Both of Bob's devices are going through Outbound Proxy B, and consequently, they include a Route header indicating Proxy B. Proxy B removes the Route header corresponding to itself, and adds itself in a Path header.

After registration, there are 2 contact bindings associated with Bob's AOR of bob@example.com: sips:bob@bobphone.example.com and sip:bob@bobpc.example.com.

Alice then calls Bob through her own Oubound Proxy A, including a Route header for Proxy A. Proxy A locates Bob's domain example.com. In this example, that domain is co-located with Bob's outbound proxy, but it could easily have been a separate proxy. Outbound Proxy A removes the Route header corresponding to itself, and inserts itself in the Record-Route and forwards the request to Proxy B.

The following subsections illustrates two examples. In the first

one, Alice calls Bob using Bob's SIPS URI, and in the second one, Alice calls Bob's SIP AOR.

9.1. Alice Calls Bob's SIPS AOR

In this first example, Alice calls Bob's SIPS address (sips:bob@example.com). Proxy B consults the binding in the registration database, and finds the 2 Contact bindings. Alice had addressed Bob with a SIPS Request-URI (sips:bob@example.com), so Proxy B determines that the calls needs to be routed only to a SIPS Contact, and therefore the request is only sent to sips:bob@bobphone.example.com. Proxy B inserts itself in the Record-Route. Bob answers.

```
                         Outbound                  Outbound
        Bob@bobpc        Proxy B     Registrar     Proxy A     Alice
         |                 |            |            |            |
         |   REGISTER F1   |            |            |            |
         |---------------->|REGISTER F2 |            |            |
         |                 |----------->|            |            |
         |                 |   200 F3   |            |            |
         |     200 F4      |<-----------|            |            |
         |---------------->|            |            |            |
         |                 |            |            |            |
         |   Bob@phone     |            |            |            |
         |    |            |            |            |            |
         |    |REGISTER F5 |            |            |            |
         |    |----------->|REGISTER F6 |            |            |
         |    |            |----------->|            |            |
         |    |            |   200 F7   |            |            |
         |    |   200 F8   |<-----------|            |            |
         |    |----------->|            |            |            |
         |    |            |                         |  INVITE F9 |
         |    |            |        INVITE F11       |<-----------|
         |    | INVITE F13 |<------------------------|   100 F10  |
         |    |<-----------|         100 F12         |----------->|
         |    |   100 F14  |------------------------>|            |
         |    |----------->|                         |            |
         |    |   200 F15  |                         |            |
         |    |----------->|         200 F16         |            |
         |    |            |------------------------>|   200 F17  |
         |    |            |                         |----------->|
         |    |            |                         |   ACK F18  |
         |    |            |         ACK F19         |<-----------|
         |    |  ACK F20   |<------------------------|            |
         |    |<-----------|                         |            |
```

Alice Calls Bob's SIPS AOR

Message details

F1 REGISTER Bob's PC Client -> Proxy B

```
REGISTER sip:registrar.example.com SIP/2.0
Via: SIP/2.0/TCP bobspc.example.com:5060;branch=z9hG4bKnashds
Max-Forwards: 70
To: Bob <sip:bob@example.com>
From: Bob <sip:bob@example.com>;tag=456248
Call-ID: 843817637684230@998sdasdh09
CSeq: 1826 REGISTER
Supported: path
Route: <sip:proxyb.example.com;lr>
Contact: <sip:bob@bobpc.example.com>
   ;+sip.instance="<urn:uuid:0C67446E-F1A1-11D9-94D3-000A95A0E128>"
   ;reg-id=1
Expires: 7200
Content-Length: 0
```

F2 REGISTER Proxy B -> Registrar

```
REGISTER sip:registrar.example.com SIP/2.0
Via: SIP/2.0/TCP proxyb.example.com:5060;branch=z9hG4bK87asdks7
Via: SIP/2.0/TCP bobspc.example.com:5060;branch=z9hG4bKnashds
Max-Forwards: 69
To: Bob <sip:bob@example.com>
From: Bob <sip:bob@example.com>;tag=456248
Call-ID: 843817637684230@998sdasdh09
CSeq: 1826 REGISTER
Supported: path
Path: <sip:laksdyjanseg237+fsdf@proxyb.example.com;lr>
Contact: <sip:bob@bobpc.example.com>
   ;+sip.instance="<urn:uuid:0C67446E-F1A1-11D9-94D3-000A95A0E128>"
   ;reg-id=1
Expires: 7200
Content-Length: 0
```

F3 200 (REGISTER) Registrar -> Proxy B

```
SIP 2.0 200 OK
Via: SIP/2.0/TCP proxyb.example.com:5060;branch=z9hG4bK87asdks7
Via: SIP/2.0/TCP bobspc.example.com:5060;branch=z9hG4bKnashds
To: Bob <sip:bob@example.com>;tag=2493K59K9
From: Bob <sip:bob@example.com>;tag=456248
Call-ID: 843817637684230@998sdasdh09
CSeq: 1826 REGISTER
Supported: outbound
Path: <sip:laksdyjanseg237+fsdf@proxyb.example.com;lr>
Contact: <sip:bob@bobphone.example.com>
   ;+sip.instance="<urn:uuid:0C67446E-F1A1-11D9-94D3-000A95A0E128>"
   ;reg-id=1
   ;expires=7200
Date: Mon, 12 Jun 2006 16:43:12 GMT
Content-Length: 0
```

F4 200 (REGISTER) Proxy B -> Bob's PC Client

```
SIP 2.0 200 OK
Via: SIP/2.0/TCP bobspc.example.com:5060;branch=z9hG4bKnashds
To: Bob <sip:bob@example.com>;tag=2493K59K9
From: Bob <sip:bob@example.com>;tag=456248
Call-ID: 843817637684230@998sdasdh09
CSeq: 1826 REGISTER
Supported: outbound
Path: <sip:laksdyjanseg237+fsdf@proxyb.example.com;lr>
Contact: <sip:bob@bobphone.example.com>
   ;+sip.instance="<urn:uuid:0C67446E-F1A1-11D9-94D3-000A95A0E128>"
   ;reg-id=1
   ;expires=7200
Date: Mon, 12 Jun 2006 16:43:12 GMT
Content-Length: 0
```

F5 REGISTER Bob's Phone -> Proxy B

```
REGISTER sips:registrar.example.com SIP/2.0
Via: SIP/2.0/TLS bobphone.example.com:5061;branch=z9hG4bK9555
Max-Forwards: 70
To: Bob <sips:bob@example.com>
From: Bob <sips:bob@example.com>;tag=90210
Call-ID: faif9a@qwefnwdclk
CSeq: 12 REGISTER
Supported: path
Route: <sips:proxyb.example.com;lr>
Contact: <sips:bob@bobphone.example.com>
   ;+sip.instance="<urn:uuid:6F85D4E3-E8AA-46AA-B768-BF39D5912143>"
   ;reg-id=1
Expires: 7200
Content-Length: 0
```

F6 REGISTER Proxy B -> Registrar

```
REGISTER sips:registrar.example.com SIP/2.0
Via: SIP/2.0/TLS proxyb.example.com:5061;branch=z9hG4bK876354
Via: SIP/2.0/TLS bobphone.example.com:5061;branch=z9hG4bK9555
Max-Forwards: 69
To: Bob <sips:bob@example.com>
From: Bob <sips:bob@example.com>;tag=90210
Call-ID: faif9a@qwefnwdclk
CSeq: 12 REGISTER
Supported: path
Path: <sips:psodkfsj+34+kkls@proxyb.example.com;lr>
Contact: <sips:bob@bobphone.example.com>
   ;+sip.instance="<urn:uuid:6F85D4E3-E8AA-46AA-B768-BF39D5912143>"
   ;reg-id=1
Expires: 7200
Content-Length: 0
```

F7 200 (REGISTER) Registrar -> Proxy B

```
SIP 2.0 200 OK
Via: SIP/2.0/TLS proxyb.example.com:5061;branch=z9hG4bK876354
Via: SIP/2.0/TLS bobphone.example.com:5061;branch=z9hG4bK9555
To: Bob <sips:bob@example.com>;tag=5150
From: Bob <sips:bob@example.com>;tag=90210
Call-ID: faif9a@qwefnwdclk
CSeq: 12 REGISTER
Supported: outbound
Path: <sips:psodkfsj+34+kkls@proxyb.example.com;lr>
Contact: <sips:bob@bobphone.example.com>
   ;+sip.instance="<urn:uuid:6F85D4E3-E8AA-46AA-B768-BF39D5912143>"
   ;reg-id=1
   ;expires=7200
Date: Mon, 12 Jun 2006 16:43:50 GMT
Content-Length: 0
```

F8 200 (REGISTER) Proxy B -> Bob's Phone

```
SIP 2.0 200 OK
Via: SIP/2.0/TLS bobphone.example.com:5061;branch=z9hG4bK9555
To: Bob <sips:bob@example.com>;tag=5150
From: Bob <sips:bob@example.com>;tag=90210
Call-ID: faif9a@qwefnwdclk
CSeq: 12 REGISTER
Supported: outbound
Path: <sips:psodkfsj+34+kkls@proxyb.example.com;lr>
Contact: <sips:bob@bobphone.example.com>
   ;+sip.instance="<urn:uuid:6F85D4E3-E8AA-46AA-B768-BF39D5912143>"
   ;reg-id=1
   ;expires=7200
Date: Mon, 12 Jun 2006 16:43:50 GMT
Content-Length: 0
```

F9 INVITE Alice -> Proxy A

```
INVITE sips:bob@example.com SIP/2.0
Via: SIP/2.0/TLS alice-1.example.net:5061;branch=z9hG4bKprout
Max-Forwards: 70
To: Bob <sips:bob@example.com>
From: Alice <sips:alice@example.net>;tag=8675309
Call-ID: lzksjf8723k@sodk6587
CSeq: 1 INVITE
Route: <sips:proxya.example.net;lr>
Contact: <sips:alice@alice-1.example.net>
Content-Type: application/sdp
Content-Length: {as per SDP}
{SDP not shown}
```

F10 100 (INVITE) Proxy A -> Alice

```
SIP 2.0 100 Trying
Via: SIP/2.0/TLS alice-1.example.net:5061;branch=z9hG4bKprout
Max-Forwards: 70
To: Bob <sips:bob@example.com>
From: Alice <sips:alice@example.net>;tag=8675309
Call-ID: lzksjf8723k@sodk6587
CSeq: 1 INVITE
Content-Length: 0
```

F11 INVITE Proxy A -> Proxy B

```
INVITE sips:bob@example.com SIP/2.0
Via: SIP/2.0/TLS proxya.example.net:5061;branch=z9hG4bKpouet
Via: SIP/2.0/TLS alice-1.example.net:5061;branch=z9hG4bKprout
Max-Forwards: 69
To: Bob <sips:bob@example.com>
From: Alice <sips:alice@example.net>;tag=8675309
Call-ID: lzksjf8723k@sodk6587
CSeq: 1 INVITE
Record-Route: <sips:KFndf+47KsFH@proxya.example.net;lr>
Contact: <sips:alice@alice-1.example.net>
Content-Type: application/sdp
Content-Length: {as per SDP}
{SDP not shown}
```

F12 100 (INVITE) Proxy B -> Proxy A

```
SIP 2.0 100 Trying
Via: SIP/2.0/TLS proxya.example.net:5061;branch=z9hG4bKpouet
Via: SIP/2.0/TLS alice-1.example.net:5061;branch=z9hG4bKprout
To: Bob <sips:bob@example.com>
From: Alice <sips:alice@example.net>;tag=8675309
Call-ID: lzksjf8723k@sodk6587
CSeq: 1 INVITE
Content-Length: 0
```

F13 INVITE Proxy B -> Bob's Phone

```
INVITE sips:bob@bobphone.example.com SIP/2.0
Via: SIP/2.0/TLS proxyb.example.com:5061;branch=z9hG4bKbalouba
Via: SIP/2.0/TLS proxya.example.net:5061;branch=z9hG4bKpouet
Via: SIP/2.0/TLS alice-1.example.net:5061;branch=z9hG4bKprout
Max-Forwards: 68
To: Bob <sips:bob@example.com>
From: Alice <sips:alice@example.net>;tag=8675309
Call-ID: lzksjf8723k@sodk6587
CSeq: 1 INVITE
Record-Route: <sips:UJH-hUdvb65@proxyb.example.com;lr>,
              <sips:KFndf+47KsFH@proxya.example.net;lr>
Contact: <sips:alice@alice-1.example.net>
Content-Type: application/sdp
Content-Length: {as per SDP}
{SDP not shown}
```

F14 100 (INVITE) Bob's Phone -> Proxy B

```
SIP 2.0 100 Trying
Via: SIP/2.0/TLS proxyb.example.com:5061;branch=z9hG4bKbalouba
Via: SIP/2.0/TLS proxya.example.net:5061;branch=z9hG4bKpouet
Via: SIP/2.0/TLS alice-1.example.net:5061;branch=z9hG4bKprout
To: Bob <sips:bob@example.com>
From: Alice <sips:alice@example.net>;tag=8675309
Call-ID: lzksjf8723k@sodk6587
CSeq: 1 INVITE
Content-Length: 0
```

F15 200 (INVITE) Bob's Phone -> Proxy B

```
SIP 2.0 200 OK
Via: SIP/2.0/TLS proxyb.example.com:5061;branch=z9hG4bKbalouba
Via: SIP/2.0/TLS proxya.example.net:5061;branch=z9hG4bKpouet
Via: SIP/2.0/TLS alice-1.example.net:5061;branch=z9hG4bKprout
To: Bob <sips:bob@example.com>;tag=5551212
From: Alice <sips:alice@example.net>;tag=8675309
Call-ID: lzksjf8723k@sodk6587
CSeq: 1 INVITE
Record-Route: <sips:UJH-hUdvb65@proxyb.example.com;lr>,
              <sips:KFndf+47KsFH@proxya.example.net;lr>
Contact: <sips:bob@bobphone.example.com>
Content-Length: 0
```

F16 200 (INVITE) Proxy B -> Proxy A

```
SIP 2.0 200 OK
Via: SIP/2.0/TLS proxya.example.net:5061;branch=z9hG4bKpouet
Via: SIP/2.0/TLS alice-1.example.net:5061;branch=z9hG4bKprout
To: Bob <sips:bob@example.com>;tag=5551212
From: Alice <sips:alice@example.net>;tag=8675309
Call-ID: lzksjf8723k@sodk6587
CSeq: 1 INVITE
Record-Route: <sips:UJH-hUdvb65@proxyb.example.com;lr>,
              <sips:KFndf+47KsFH@proxya.example.net;lr>
Contact: <sips:bob@bobphone.example.com>
Content-Length: 0
```

F17 200 (INVITE) Proxy A -> Alice

```
SIP 2.0 200 OK
Via: SIP/2.0/TLS alice-1.example.net:5061;branch=z9hG4bKprout
To: Bob <sips:bob@example.com>;tag=5551212
From: Alice <sips:alice@example.net>;tag=8675309
Call-ID: lzksjf8723k@sodk6587
CSeq: 1 INVITE
Record-Route: <sips:UJH-hUdvb65@proxyb.example.com;lr>,
              <sips:KFndf+47KsFH@proxya.example.net;lr>
Contact: <sips:bob@bobphone.example.com>
Content-Length: 0
```

```
F18 ACK Alice -> Proxy A

ACK sips:bob@bobphone.example.com SIP/2.0
Via: SIP/2.0/TLS alice-1.example.net:5061;branch=z9hG4bKksdjf
Max-Forwards: 70
To: Bob <sips:bob@example.com>;tag=5551212
From: Alice <sips:alice@example.net>;tag=8675309
Call-ID: lzksjf8723k@sodk6587
CSeq: 1 ACK
Route: <sips:KFndf+47KsFH@proxya.example.net;lr>,
       <sips:UJH-hUdvb65@proxyb.example.com;lr>
Content-Lenght: 0


F19 ACK Proxy A -> Proxy B

ACK sips:bob@bobphone.example.com SIP/2.0
Via: SIP/2.0/TLS proxya.example.net:5061;branch=z9hG4bKplo7hy
Via: SIP/2.0/TLS alice-1.example.net:5061;branch=z9hG4bKksdjf
Max-Forwards: 69
To: Bob <sips:bob@example.com>;tag=5551212
From: Alice <sips:alice@example.net>;tag=8675309
Call-ID: lzksjf8723k@sodk6587
CSeq: 1 ACK
Route: <sips:UJH-hUdvb65@proxyb.example.com;lr>
Content-Lenght: 0


F20 ACK Proxy B -> Bob's Phone

ACK sips:bob@bobphone.example.com SIP/2.0
Via: SIP/2.0/TLS proxyb.example.com:5061;branch=z9hG4bK8msdu2
Via: SIP/2.0/TLS proxya.example.net:5061;branch=z9hG4bKplo7hy
Via: SIP/2.0/TLS alice-1.example.net:5061;branch=z9hG4bKksdjf
Max-Forwards: 68
To: Bob <sips:bob@example.com>;tag=5551212
From: Alice <sips:alice@example.net>;tag=8675309
Call-ID: lzksjf8723k@sodk6587
CSeq: 1 ACK
Content-Lenght: 0
```

9.2. Alice Calls Bob's SIP AOR

In the second example, Alice calls Bob's SIP address instead (sip:bob@example.com). Proxy B consults the binding in the registration database, and finds the 2 Contact bindings. Alice had addressed Bob with a SIP Request-URI (sip:bob@example.com), so Proxy B determines that the calls needs to be routed both to the SIP

Contact and the SIPS Contact, and therefore the request is forked sent to sip:bob@boppc.example.com and sips:bob@bobphone.example.com. Proxy B inserts itself in the Record-Route. Bob's phone's policy is to accept calls to SIP and SIPS (i.e., "best effort") so both his PC Client and his SIP Phone ring simultaneously. Bob answers on his SIP phone, and the forked call leg to the PC client is canceled.

```
                        Outbound                  Outbound
        Bob@bobpc        Proxy B     Registrar     Proxy A     Alice
         |                 |            |            |            |
         |   REGISTER F1   |            |            |            |
         |---------------->|REGISTER F2 |            |            |
         |                 |----------->|            |            |
         |                 |   200 F3   |            |            |
         |     200 F4      |<-----------|            |            |
         |---------------->|            |            |            |
         |                 |            |            |            |
         |   Bob@phone     |            |            |            |
         |    |            |            |            |            |
         |    |REGISTER F5 |            |            |            |
         |    |----------->|REGISTER F6 |            |            |
         |    |            |----------->|            |            |
         |    |            |   200 F7   |            |            |
         |    |   200 F8   |<-----------|            |            |
         |    |----------->|            |            |            |
         |                 |                         |  INVITE F9 |
         |                 |        INVITE F11       |<-----------|
         |   INVITE F13'   |<------------------------|   100 F10  |
         |<----------------|         100 F12         |----------->|
         |    100 F14'     |------------------------>|            |
         |---------------->|                         |            |
         |    180 F15'     |                         |            |
         |---------------->|         180 F16'        |            |
         |                 |------------------------>|   180 F17' |
         |    | INVITE F13 |                         |----------->|
         |    |<-----------|                         |            |
         |    |  100 F14   |                         |            |
         |    |----------->|                         |            |
         |    |   200 F15  |                         |            |
         |    |----------->|         200 F16         |            |
         |    |            |------------------------>|   200 F17  |
         |    |            |                         |----------->|
         |    |            |                         |   ACK F18  |
         |    |            |         ACK F19         |<-----------|
         |    |  ACK F20   |<------------------------|            |
         |    |<-----------|                         |            |
         |                 |                         |            |
         |   CANCEL F20'   |                         |            |
         |<----------------|                         |            |
         |    200 F21'     |                         |            |
         |---------------->|                         |            |
         |    487 F22'     |                         |            |
         |---------------->|                         |            |
```

                         Alice Calls Bob's SIP AOR

Messages F1-F8 are identical to the ones in Section 9.1. The other messages are as follows.

F9 INVITE Alice -> Proxy A

```
INVITE sip:bob@example.com SIP/2.0
Via: SIP/2.0/TCP alice-1.example.net:5060;branch=z9hG4bKprout
Max-Forwards: 70
To: Bob <sip:bob@example.com>
From: Alice <sip:alice@example.net>;tag=8675309
Call-ID: lzksjf8723k@sodk6587
CSeq: 1 INVITE
Route: <sip:proxya.example.net;lr>
Contact: <sip:alice@alice-1.example.net>
Content-Type: application/sdp
Content-Length: {as per SDP}
{SDP not shown}
```

F10 100 (INVITE) Proxy A -> Alice

```
SIP 2.0 100 Trying
Via: SIP/2.0/TCP alice-1.example.net:5060;branch=z9hG4bKprout
Max-Forwards: 70
To: Bob <sips:bob@example.com>
From: Alice <sips:alice@example.net>;tag=8675309
Call-ID: lzksjf8723k@sodk6587
CSeq: 1 INVITE
Content-Length: 0
```

F11 INVITE Proxy A -> Proxy B

```
INVITE sip:bob@example.com SIP/2.0
Via: SIP/2.0/TCP proxya.example.net:5060;branch=z9hG4bKpouet
Via: SIP/2.0/TCP alice-1.example.net:5060;branch=z9hG4bKprout
Max-Forwards: 69
To: Bob <sip:bob@example.com>
From: Alice <sip:alice@example.net>;tag=8675309
Call-ID: lzksjf8723k@sodk6587
CSeq: 1 INVITE
Record-Route: <sip:KFndf+47KsFH@proxya.example.net;lr>
Contact: <sip:alice@alice-1.example.net>
Content-Type: application/sdp
Content-Length: {as per SDP}
{SDP not shown}
```

F12 100 (INVITE) Proxy B -> Proxy A

```
SIP 2.0 100 Trying
Via: SIP/2.0/TCP proxya.example.net:5060;branch=z9hG4bKpouet
Via: SIP/2.0/TCP alice-1.example.net:5060;branch=z9hG4bKprout
To: Bob <sip:bob@example.com>
From: Alice <sip:alice@example.net>;tag=8675309
Call-ID: lzksjf8723k@sodk6587
CSeq: 1 INVITE
Content-Length: 0
```

F13' INVITE Proxy B -> Bob's PC Client

```
INVITE sip:bob@bobphone.example.com SIP/2.0
Via: SIP/2.0/TCP proxyb.example.com:5060;branch=z9hG4bKbalouba.2
Via: SIP/2.0/TCP proxya.example.net:5060;branch=z9hG4bKpouet
Via: SIP/2.0/TCP alice-1.example.net:5060;branch=z9hG4bKprout
Max-Forwards: 68
To: Bob <sip:bob@example.com>
From: Alice <sip:alice@example.net>;tag=8675309
Call-ID: lzksjf8723k@sodk6587
CSeq: 1 INVITE
Record-Route: <sip:UJH-hUdvb65@proxyb.example.com;lr>,
              <sip:KFndf+47KsFH@proxya.example.net;lr>
Contact: <sip:alice@alice-1.example.net>
Content-Type: application/sdp
Content-Length: {as per SDP}
{SDP not shown}
```

F14' 100 (INVITE) Bob's PC Client -> Proxy B

```
SIP 2.0 100 Trying
Via: SIP/2.0/TCP proxyb.example.com:5060;branch=z9hG4bKbalouba.2
Via: SIP/2.0/TCP proxya.example.net:5060;branch=z9hG4bKpouet
Via: SIP/2.0/TCP alice-1.example.net:5060;branch=z9hG4bKprout
To: Bob <sip:bob@example.com>
From: Alice <sip:alice@example.net>;tag=8675309
Call-ID: lzksjf8723k@sodk6587
CSeq: 1 INVITE
Content-Length: 0
```

```
F15' 180 (INVITE) Bob's PC Client -> Proxy B

SIP 2.0 200 OK
Via: SIP/2.0/TCP proxyb.example.com:5060;branch=z9hG4bKbalouba.2
Via: SIP/2.0/TCP proxya.example.net:5060;branch=z9hG4bKpouet
Via: SIP/2.0/TCP alice-1.example.net:5060;branch=z9hG4bKprout
To: Bob <sip:bob@example.com>;tag=963258
From: Alice <sip:alice@example.net>;tag=8675309
Call-ID: lzksjf8723k@sodk6587
CSeq: 1 INVITE
Record-Route: <sip:UJH-hUdvb65@proxyb.example.com;lr>,
              <sip:KFndf+47KsFH@proxya.example.net;lr>
Contact: <sip:bob@bobpc.example.com>
Content-Length: 0


F16' 180 (INVITE) Proxy B -> Proxy A

SIP 2.0 200 OK
Via: SIP/2.0/TCP proxya.example.net:5060;branch=z9hG4bKpouet
Via: SIP/2.0/TCP alice-1.example.net:5060;branch=z9hG4bKprout
To: Bob <sip:bob@example.com>;tag=963258
From: Alice <sip:alice@example.net>;tag=8675309
Call-ID: lzksjf8723k@sodk6587
CSeq: 1 INVITE
Record-Route: <sip:UJH-hUdvb65@proxyb.example.com;lr>,
              <sip:KFndf+47KsFH@proxya.example.net;lr>
Contact: <sip:bob@bobpc.example.com>
Content-Length: 0


F17' 180 (INVITE) Proxy A -> Alice

SIP 2.0 200 OK
Via: SIP/2.0/TCP alice-1.example.net:5060;branch=z9hG4bKprout
To: Bob <sip:bob@example.com>;tag=963258
From: Alice <sip:alice@example.net>;tag=8675309
Call-ID: lzksjf8723k@sodk6587
CSeq: 1 INVITE
Record-Route: <sip:UJH-hUdvb65@proxyb.example.com;lr>,
              <sip:KFndf+47KsFH@proxya.example.net;lr>
Contact: <sip:bob@bobpc.example.com>
Content-Length: 0
```

F13 INVITE Proxy B -> Bob's Phone

```
INVITE sips:bob@bobphone.example.com SIP/2.0
Via: SIP/2.0/TLS proxyb.example.com:5061;branch=z9hG4bKbalouba.1
Via: SIP/2.0/TCP proxya.example.net:5060;branch=z9hG4bKpouet
Via: SIP/2.0/TCP alice-1.example.net:5060;branch=z9hG4bKprout
Max-Forwards: 68
To: Bob <sip:bob@example.com>
From: Alice <sip:alice@example.net>;tag=8675309
Call-ID: lzksjf8723k@sodk6587
CSeq: 1 INVITE
Record-Route: <sips:UJH-hUdvb65@proxyb.example.com;lr>,
              <sip:UJH-hUdvb65@proxyb.example.com;lr>,
              <sip:KFndf+47KsFH@proxya.example.net;lr>
Contact: <sip:alice@alice-1.example.net>
Content-Type: application/sdp
Content-Length: {as per SDP}
{SDP not shown}
```

F14 100 (INVITE) Bob's Phone -> Proxy B

```
SIP 2.0 100 Trying
Via: SIP/2.0/TLS proxyb.example.com:5061;branch=z9hG4bKbalouba.1
Via: SIP/2.0/TCP proxya.example.net:5060;branch=z9hG4bKpouet
Via: SIP/2.0/TCP alice-1.example.net:5060;branch=z9hG4bKprout
To: Bob <sip:bob@example.com>
From: Alice <sip:alice@example.net>;tag=8675309
Call-ID: lzksjf8723k@sodk6587
CSeq: 1 INVITE
Content-Length: 0
```

F15 200 (INVITE) Bob's Phone -> Proxy B

```
SIP 2.0 200 OK
Via: SIP/2.0/TLS proxyb.example.com:5061;branch=z9hG4bKbalouba.1
Via: SIP/2.0/TCP proxya.example.net:5060;branch=z9hG4bKpouet
Via: SIP/2.0/TCP alice-1.example.net:5060;branch=z9hG4bKprout
To: Bob <sip:bob@example.com>;tag=5551212
From: Alice <sip:alice@example.net>;tag=8675309
Call-ID: lzksjf8723k@sodk6587
CSeq: 1 INVITE
Record-Route: <sips:UJH-hUdvb65@proxyb.example.com;lr>,
              <sip:UJH-hUdvb65@proxyb.example.com;lr>,
              <sip:KFndf+47KsFH@proxya.example.net;lr>
Contact: <sips:bob@bobphone.example.com>
Content-Length: 0
```

F16 200 (INVITE) Proxy B -> Proxy A

```
SIP 2.0 200 OK
Via: SIP/2.0/TCP proxya.example.net:5060;branch=z9hG4bKpouet
Via: SIP/2.0/TCP alice-1.example.net:5060;branch=z9hG4bKprout
To: Bob <sip:bob@example.com>;tag=5551212
From: Alice <sip:alice@example.net>;tag=8675309
Call-ID: lzksjf8723k@sodk6587
CSeq: 1 INVITE
Record-Route: <sips:UJH-hUdvb65@proxyb.example.com;lr>,
              <sip:UJH-hUdvb65@proxyb.example.com;lr>,
              <sip:KFndf+47KsFH@proxya.example.net;lr>
Contact: <sips:bob@bobphone.example.com>
Content-Length: 0
```

```
F17 200 (INVITE) Proxy A -> Alice

SIP 2.0 200 OK
Via: SIP/2.0/TCP alice-1.example.net:5060;branch=z9hG4bKprout
To: Bob <sip:bob@example.com>;tag=5551212
From: Alice <sip:alice@example.net>;tag=8675309
Call-ID: lzksjf8723k@sodk6587
CSeq: 1 INVITE
Record-Route: <sips:UJH-hUdvb65@proxyb.example.com;lr>,
              <sip:UJH-hUdvb65@proxyb.example.com;lr>,
              <sip:KFndf+47KsFH@proxya.example.net;lr>
Contact: <sips:bob@bobphone.example.com>
Content-Length: 0


F18 ACK Alice -> Proxy A

ACK sips:bob@bobphone.example.com SIP/2.0
Via: SIP/2.0/TCP alice-1.example.net:5060;branch=z9hG4bKprout
Max-Forwards: 70
To: Bob <sips:bob@example.com>;tag=5551212
From: Alice <sips:alice@example.net>;tag=8675309
Call-ID: lzksjf8723k@sodk6587
CSeq: 1 ACK
Route: <sip:KFndf+47KsFH@proxya.example.net;lr>,
       <sip:UJH-hUdvb65@proxyb.example.com;lr>,
       <sips:UJH-hUdvb65@proxyb.example.com;lr>
Content-Lenght: 0


F19 ACK Proxy A -> Proxy B

ACK sips:bob@bobphone.example.com SIP/2.0
Via: SIP/2.0/TCP proxya.example.net:5060;branch=z9hG4bKpouet
Via: SIP/2.0/TCP alice-1.example.net:5060;branch=z9hG4bKprout
Max-Forwards: 69
To: Bob <sip:bob@example.com>;tag=5551212
From: Alice <sip:alice@example.net>;tag=8675309
Call-ID: lzksjf8723k@sodk6587
CSeq: 1 ACK
Route: <sip:UJH-hUdvb65@proxyb.example.com;lr>,
       <sips:UJH-hUdvb65@proxyb.example.com;lr>
Content-Lenght: 0
```

F20 ACK Proxy B -> Bob's Phone

```
ACK sips:bob@bobphone.example.com SIP/2.0
Via: SIP/2.0/TLS proxyb.example.com:5061;branch=z9hG4bKbalouba.1
Via: SIP/2.0/TCP proxya.example.net:5060;branch=z9hG4bKpouet
Via: SIP/2.0/TCP alice-1.example.net:5060;branch=z9hG4bKprout
Max-Forwards: 68
To: Bob <sip:bob@example.com>;tag=5551212
From: Alice <sip:alice@example.net>;tag=8675309
Call-ID: lzksjf8723k@sodk6587
CSeq: 1 ACK
Content-Lenght: 0
```

F20' CANCEL Proxy B -> Bob's PC Client

```
CANCEL sip:bob@bobpc.example.com SIP/2.0
Via: SIP/2.0/TCP proxyb.example.com:5060;branch=z9hG4bKbalouba.2
Max-Forwards: 70
To: Bob <sip:bob@example.com>;tag=5551212
From: Alice <sip:alice@example.net>;tag=8675309
Call-ID: lzksjf8723k@sodk6587
CSeq: 1 CANCEL
Content-Lenght: 0
```

F21' 200 (CANCEL) Proxy B -> Bob's PC Client

```
SIP 2.0 200 OK
Via: SIP/2.0/TCP proxyb.example.com:5060;branch=z9hG4bKbalouba.2
To: Bob <sip:bob@example.com>;tag=5551212
From: Alice <sip:alice@example.net>;tag=8675309
Call-ID: lzksjf8723k@sodk6587
CSeq: 1 CANCEL
Content-Lenght: 0
```

F22' 487 (INVITE) Proxy B -> Bob's PC Client

```
SIP 2.0 487 Request Terrminated
Via: SIP/2.0/TCP proxyb.example.com:5060;branch=z9hG4bKbalouba.2
Via: SIP/2.0/TCP proxya.example.net:5060;branch=z9hG4bKpouet
Via: SIP/2.0/TCP alice-1.example.net:5060;branch=z9hG4bKprout
To: Bob <sip:bob@example.com>
From: Alice <sip:alice@example.net>;tag=8675309
Call-ID: lzksjf8723k@sodk6587
CSeq: 1 INVITE
Content-Length: 0
```

10. Conclusion

The restrictions described in this document have consequences on the applicability of the SIPS URI scheme.

SIP [RFC3261] itself introduces some complications with using SIPS, for example when using strict routing instead of loose routing. When a SIPS URI is used in a Contact in a dialog initiating request and Record-Route is not used, that SIPS URI may not be usable by the other end. If the other end does not support SIPS and/or TLS, it will not be able to use it. The "last-hop exception" is an example of when this may occur. In this case, using Record-Route so that the requests are sent through proxies helps in making it work. Another example of issues with strict routing is that even in a case where the Contact is a SIPS URI, no Record-Route is used, and the far end supports SIPS and TLS, it may still not be possible for the far end to establish a TLS connection with the SIP originating end if the certificate can not be validated by the far end. This could typically be the case if the originating end was using server-side authentication as described below, or even if the originating end is not using a certificate that can be validated. In both cases, [I-D.ietf-sip-outbound] and Record-Route may be used to solve the problem.

TLS itself has a significant impact on how SIPS may be used. "server-side authentication" (where the server side provides its certificate but the client side does not) is typically used between a SIP end-user device acting as the TLS client side (like a phone or a personal computer), and it's SIP server (proxy or registrar) acting as the TLS server side. "Mutual TLS" (where both the client and the server side provide their respective certificate) is typically used between SIP servers (proxies, registrars), or statically configured devices such as PSTN gateways or media servers. In the mutual TLS model, for two entities to be able to establish a TLS connection requires that both side be able to validate each other's certificates, either by static

configuration or by being able to recurse to a valid root certificate. With server-side authentication, only the client side is capable of validating the server side's certificate, as the client side does not provide a certificate. The consequences of all this are that whenever a SIPS URI is used to establish a TLS connection, it must be possible for the entity establishing the connection (the client) to validate the certificate from the server side. For server-side authentication, [I-D.ietf-sip-outbound] is the RECOMMENDED approach. For mutual TLS, it means that one should be very careful that the architecture of the network is such that connections are made between entities that have access to each other's credential. Record-Route [RFC3261] and Path [RFC3327] are very useful in ensuring that previously established TLS connections can be re-used. Other mechanism may also be used in certain circumstances: for example, using root-certificates that are widely recognized may allow for more easily creqated TLS connections.

The "last hop exception" introduces significant potential vulnerabilities in SIP. Obviously, there is no garantee on the type of security that will be used on that last hop as it will be completely up to the target domain. Annother vulnerability is that there is no way to ensure that the last hop will really be the last hop: that hop could redirect or retarget to more hops. These hops could even be outside of the original target domain, and it is possible that the fact that it was retargeted by an entity that was not secured through TLS may be undetectable.

11. Security Considerations

Most of this document can be considered to be security considerations since it applies to the usage of the SIPS URI.

12. IANA Considerations

There are no IANA considerations.

13. IAB Considerations

There are no IAB considerations.

14. Acknowledgments

The author would like to thank Jon Peterson, Cullen Jennings, and Jonathan Rosenberg for their help. The author would like to thank

John Elwell, Paul Kyzivat, Eric Rescorla, Rifaat Shekh-Yusef, Peter Reissner, Samir Srivastava and Tina Tsou for their careful review and input.

15. References

15.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.

[RFC3327] Willis, D. and B. Hoeneisen, "Session Initiation Protocol (SIP) Extension Header Field for Registering Non-Adjacent Contacts", RFC 3327, December 2002.

15.2. Informational References

[RFC3263] Rosenberg, J. and H. Schulzrinne, "Session Initiation Protocol (SIP): Locating SIP Servers", RFC 3263, June 2002.

[RFC3515] Sparks, R., "The Session Initiation Protocol (SIP) Refer Method", RFC 3515, April 2003.

[RFC3608] Willis, D. and B. Hoeneisen, "Session Initiation Protocol (SIP) Extension Header Field for Service Route Discovery During Registration", RFC 3608, October 2003.

[RFC3725] Rosenberg, J., Peterson, J., Schulzrinne, H., and G. Camarillo, "Best Current Practices for Third Party Call Control (3pcc) in the Session Initiation Protocol (SIP)", BCP 85, RFC 3725, April 2004.

[RFC3891] Mahy, R., Biggs, B., and R. Dean, "The Session Initiation Protocol (SIP) "Replaces" Header", RFC 3891, September 2004.

[RFC3892] Sparks, R., "The Session Initiation Protocol (SIP) Referred-By Mechanism", RFC 3892, September 2004.

[RFC3893] Peterson, J., "Session Initiation Protocol (SIP) Authenticated Identity Body (AIB) Format", RFC 3893,

September 2004.

[RFC3911] Mahy, R. and D. Petrie, "The Session Initiation Protocol (SIP) "Join" Header", RFC 3911, October 2004.

[RFC4168] Rosenberg, J., Schulzrinne, H., and G. Camarillo, "The Stream Control Transmission Protocol (SCTP) as a Transport for the Session Initiation Protocol (SIP)", RFC 4168, October 2005.

[RFC4346] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.1", RFC 4346, April 2006.

[RFC4474] Peterson, J. and C. Jennings, "Enhancements for Authenticated Identity Management in the Session Initiation Protocol (SIP)", RFC 4474, August 2006.

[I-D.ietf-sip-outbound] Jennings, C. and R. Mahy, "Managing Client Initiated Connections in the Session Initiation Protocol (SIP)", draft-ietf-sip-outbound-04 (work in progress), June 2006.

[I-D.ietf-sip-gruu] Rosenberg, J., "Obtaining and Using Globally Routable User Agent (UA) URIs (GRUU) in the Session Initiation Protocol (SIP)", draft-ietf-sip-gruu-10 (work in progress), August 2006.

Appendix A. To-Be-Done

TBD: Need to look at Replaces [RFC3891], Join [RFC3911] and Target-Dialog. For example, what if this header field is received in a request to a SIPS URI but the dialog to which it relates has a SIP local target, or vice-versa?

TBD: Third-party call control [RFC3725] may also have its own set of issues to investigate.

REFER [RFC3515] and also [RFC3892] introduces its own set of issues with sips:

OPEN ISSUE: What if a UA with no support for TLS receives a SIPS URI in a Refer-to header in a REFER request? Does it reject the REFER, or accept REFER and send back a 416 in a NOTIFY (whouldn't work if norefersub is used)?

OPEN ISSUE How should the UAC sending a REFER react if it receives a 416 in response to the REFER?

OPEN ISSUE What if a UA with TLS support receives a SIP URI in a Refer-to header? Is it allowed to "upgrade" to a SIPS URI? It is probably a bad idea in most scenarios, unless it already knows that the other ends supports TLS (and has a SIPS URI).

Appendix B. Background

This section is included for reference purposes. It is intended that this appendix will be removed in a further revision of this draft.

The use of the SIPS URI scheme in SIP is scattered throughout the following sections of [RFC3261].

8.1.1.8 describes the use of the Contact header field. Of particular importance are the following statements:

> The Contact header field MUST be present and contain exactly one SIP or SIPS URI in any request that can result in the establishment of a dialog.
> If the Request-URI or top Route header field value contains a SIPS URI, the Contact header field MUST contain a SIPS URI as well.

8.1.3.4 describes processing of 3XX responses. Of particular importance is the following statement:

> If the original request had a SIPS URI in the Request-URI, the client MAY choose to recurse to a non-SIPS URI, but SHOULD inform the user of the redirection to an insecure URI.

8.1.3.5 and 8.2.2.1 implies that if a SIPS is not supported by UAS, it can reject it with a 416, and the UAC SHOULD retry the request with a SIP URI. However, although not discussed in [RFC3261], the user should be informed.

10.2.1 describes address binding of SIPS AOR during registration:

> If the address-of-record in the To header field of a REGISTER request is a SIPS URI, then any Contact header field values in the request SHOULD also be SIPS URIs. Clients should only register non-SIPS URIs under a SIPS address-of-record when the security of the resource represented by the contact address is guaranteed by other means. This may be applicable to URIs that invoke protocols other than SIP, or SIP devices secured by protocols other than TLS.

12.1.1 describes the UAS behavior when creating a dialog with a SIPS Request-URI or a top Record-Route header:

> If the request that initiated the dialog contained a SIPS URI in the Request-URI or in the top Record-Route header field value, if there was any, or the Contact header field if there was no Record-Route header field, the Contact header field in the response MUST be a SIPS URI.

12.1.2 describes the UAC behavior when creating a dialog with a SIPS Request-URI or a top Recored-Route header. Of particular importance are the following statements:

> If the request has a Request-URI or a topmost Route header field value with a SIPS URI, the Contact header field MUST contain a SIPS URI.
> If the request was sent over TLS, and the Request-URI contained a SIPS URI, the "secure" flag is set to TRUE.

12.2.1.1 expands on what this secure flag means when doing any target refresh requests within that dialog:

> A UAC SHOULD include a Contact header field in any target refresh requests within a dialog, and unless there is a need to change it, the URI SHOULD be the same as used in previous requests within the dialog. If the "secure" flag is true, that URI MUST be a SIPS URI.

16.6 bullet 4 describes Record Route processing for SIPS URIs by proxies:

> If the Request-URI contains a SIPS URI, or the topmost Route header field value [...] contains a SIPS URI, the URI placed into the Record-Route header field MUST be a SIPS URI. Furthermore, if the request was not received over TLS, the proxy MUST insert a Record-Route header field. In a similar fashion, a proxy that receives a request over TLS, but generates a request without a SIPS URI in the Request-URI or topmost Route header field value [...], MUST insert a Record-Route header field that is not a SIPS URI.

16.7 describes proxy response forwarding with Record-Route:

> If the proxy received the request over TLS, and sent it outover a non-TLS connection, the proxy MUST rewrite the URI in the Record-Route header field to be a SIPS URI. If the proxy received the request over a non-TLS connection, and sent it outover TLS, the proxy MUST rewrite the URI in the Record-Route header field to be

a SIP URI.

19.1 describes the SIP and SIPS URI in general. Of particular importance is the following statement:

A SIPS URI specifies that the resource be contacted securely. This means, in particular, that TLS is to be used between the UAC and the domain that owns the URI. From there, secure communications are used to reach the user, where the specific security mechanism depends on the policy of the domain. Any resource described by a SIP URI can be "upgraded" to a SIPS URI by just changing the scheme, if it is desired to communicate with that resource securely.

19.1.4 describes rules for URI comparisons. Of particular importance is the following statement:

Some operations in this specification require determining whether two SIP or SIPS URIs are equivalent. In this specification, registrars need to compare bindings in Contact URIs in REGISTER requests (see Section 10.3.). SIP and SIPS URIs are compared for equality according to the following rules:
o A SIP and SIPS URI are never equivalent.

20.42 describes indicating TLS transport in Via headers:

A Via header field value contains the transport protocol used to send the message, [...] Transport protocols defined here are "UDP", "TCP", "TLS", and "SCTP". "TLS" means TLS over TCP. When a request is sent to a SIPS URI, the protocol still indicates "SIP", and the transport protocol is TLS.

26.2.1 describes Transport Layer Security [RFC4346]. Of particular importance is the following statement:

"tls" (signifying TLS over TCP) can be specified as the desired transport protocol within a Via header field value or a SIP-URI.

26.2.2 is very important and describes the SIPS URI scheme. Of particular importance is the following statements:

When used as the Request-URI of a request, the SIPS scheme signifies that each hop over which the request is forwarded, until the request reaches the SIP entity responsible for the domain portion of the Request-URI, must be secured with TLS; once it reaches the domain in question it is handled in accordance with local security and routing policy, quite possibly using TLS for any last hop to a UAS. When used by the originator of a request

(as would be the case if they employed a SIPS URI as the address-of-record of the target), SIPS dictates that the entire request path to the target domain be so secured.
[...]
Note that in the SIPS URI scheme, transport is independent of TLS, and thus "sips:alice@atlanta.com;transport=tcp" and "sips:alice@atlanta.com;transport=sctp" are both valid (although note that UDP is not a valid transport for SIPS). The use of "transport=tls" has consequently been deprecated, partly because it was specific to a single hop of the request. This is a change since RFC 2543.
Users that distribute a SIPS URI as an address-of-record may elect to operate devices that refuse requests over insecure transports.

26.4.4 describes the limitations in what to infer from using SIPS URIs. Of particular importance are the the following important statement:

> Location services are not required to provide a SIPS binding for a SIPS Request-URI. Although location services are commonly populated by user registrations (as described in Section 10.2.1), various other protocols and interfaces could conceivably supply contact addresses for an AOR, and these tools are free to map SIPS URIs to SIP URIs as appropriate. When queried for bindings, a location service returns its contact addresses without regard for whether it received a request with a SIPS Request-URI. If a redirect server is accessing the location service, it is up to the entity that processes the Contact header field of a redirection to determine the propriety of the contact addresses.
> Actually using TLS on every segment of a request path entails that the terminating UAS must be reachable over TLS (perhaps registering with a SIPS URI as a contact address). This is the preferred use of SIPS. Many valid architectures, however, use TLS to secure part of the request path, but rely on some other mechanism for the final hop to a UAS, for example. Thus SIPS cannot guarantee that TLS usage will be truly end-to-end. [...]

The reader should also be familiar with [RFC3263] which describes the use of DNS with SIPS schemes.

Finally, because in practical implementations TLS will often be implemented using client-initiated connections, the reader should be familar with [I-D.ietf-sip-outbound].

Author's Address

Francois Audet
Nortel Networks
4655 Great America Parkway
Santa Clara, CA 95054
US

Phone: +1 408 495 3756
Email: audet@nortel.com

Acknowledgment

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).