TTC標準
Standard

JJ-22.15

企業 SIP 網に接続する SIP 端末⇔サーバ間 SIPS URI スキーム技術仕様

SIPS URI Scheme technical specifications
between SIP terminal <=> Servers linked to a Private SIP network

第 1.0 版

2018 年 11 月 15 日制定

一般社団法人
情報通信技術委員会

THE TELECOMMUNICATION TECHNOLOGY COMMITTEE

目　次

<参考>

1. 国際勧告等の関連

本標準に関する国際勧告はない。

2. 改版の履歴

版数	制定日	改版内容
第 1.0 版	2018 年 11 月 15 日	初版制定

3. 参照文書

3.1 必須文書

[1] JJ-22.01 “企業 SIP 網間における相互接続インタフェース”(Technical Specifications on Inter-connection Interface between Private SIP Networks)

[2] RFC5630 The Use of the SIPS URI Scheme in the Session Initiation Protocol (SIP)

[3] RFC3261 SIP: Session Initiation Protocol

[4] RFC5246 The Transport Layer Security (TLS) Protocol Version 1.2

[5] RFC5626 Managing Client-Initiated Connections in the Session Initiation Protocol (SIP)

3.2 参考文書

[6] RFC2543 SIP: Session Initiation Protocol

[7] RFC3327 Session Initiation Protocol (SIP) Extension Header Field for Registering Non-Adjacent Contacts

[8] RFC3515 The Session Initiation Protocol (SIP) Refer Method

[9] RFC3608 Session Initiation Protocol (SIP) Extension Header Field for Service Route Discovery During Registration

[10] RFC3725 Best Current Practices for Third Party Call Control (3pcc) in the Session Initiation Protocol (SIP)

[11] RFC3891 The Session Initiation Protocol (SIP) "Replaces" Header

[12] RFC3893 Session Initiation Protocol (SIP) Authenticated Identity Body (AIB) Format

[13] RFC3911 The Session Initiation Protocol (SIP) "Join" Header

[14] RFC4168 The Stream Control Transmission Protocol (SCTP) as a Transport for the Session Initiation Protocol (SIP)

[15] RFC4244 An Extension to the Session Initiation Protocol (SIP) for Request History Information

[16] RFC4474 Enhancements for Authenticated Identity Management in the Session Initiation Protocol (SIP)

[17] RFC5627 Obtaining and Using Globally Routable User Agent URIs (GRUU) in the Session Initiation Protocol (SIP)

4. 工業所有権

TTC の「工業所有権等の実施の権利に係る確認書」の提出状況は、TTC ホームページで公開されている。

5. 標準策定部門

企業ネットワーク専門委員会

1. 概説

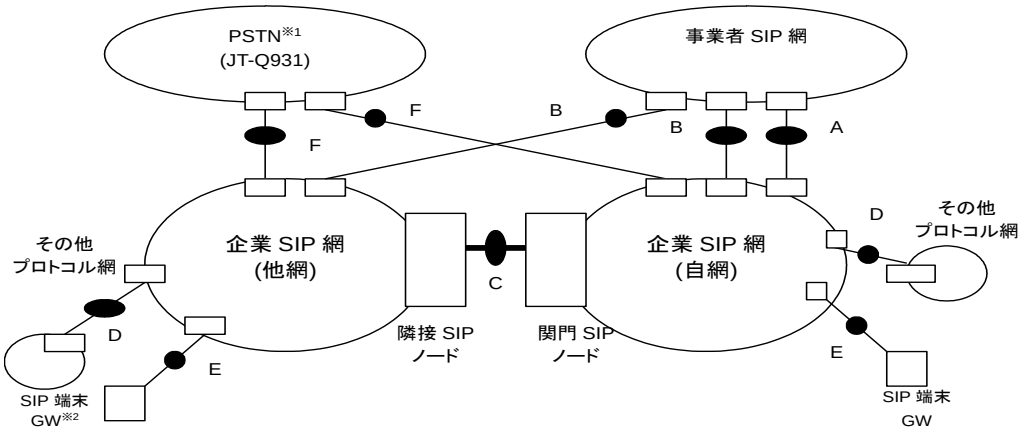
1.1. 本標準の適用範囲

本標準は、JJ-22.01 [1] に規定されるフレームワーク標準の網接続アーキテクチャにおいて、私設総合サービス網交換機 (PINX : Private Integrated services Network eXchange) および SIP (Session Initiation Protocol) 端末間 (インタフェース C、E) の SIPS スキームを用いたセキュアな SIP セッション確立のための推奨仕様を規定するものである。

また、インタフェース B を経由して接続される端末が、本標準の範囲を超えた能力を保持することを妨げるものではない。但し、その場合においても本標準に準拠する端末との接続性について考慮することが望ましい。

1.1.1. 基本接続形態

本標準は、図 1.1 で示す企業 SIP 網相互接続モデルに規定されるインタフェース C、E に適用可能な管理された企業 SIP 網との接続インタフェースの条件を示す。本インタフェースの規定を遵守できるインタフェースを有する企業 SIP 網に関して、本標準では“管理された企業 SIP 網”と呼ぶ。以下企業 SIP 網と表記する場合は、“管理された企業 SIP 網”であることを前提とする。



※1：PSTN・・・Public Switched Telephone Networks
※2：GW・・・GateWay

図 1.1 企業 SIP 網相互接続モデル

1.2. 本標準の目的と規定

本標準では、私設総合サービス網交換機 (PINX) 及び SIP 端末の実装に際して、

- 接続条件に関わる規定の解釈を一意とすることで、実装可能な標準とする。
- SIP 網を介した私設総合サービス網交換機 (PINX) と SIP 端末との接続において、共通的に適用することが可能な標準とする。
- 本規定の範囲を超えるまたは、厳密に本規定を遵守していない SIP UA (User Agent) との接続性にも最大限配慮した標準とする。

ことを目的に以下の規定を行う。

- SIPS スキームが定義されている [RFC5630] [2] に関する事項

1.3. 本標準の内容

本仕様の本文では、主として以下の事項について規定を行う。

- 私設総合サービス網交換機 (PINX) と SIP 端末は、B2BUA で接続されることを前提とし、SIP 網を介した私設総合サービス網交換機 (PINX) にセキュアに接続するための動作規定として、SIPS スキームおよび TLS (Transport Layer Security) 接続 [RFC5246] [4] を使用する場合のモデルについて規定する (2 章、3 章、4 章)。
- SIPS スキームを用いた場合のコールフローの例示をを行う (5 章)。
- SIP スキームを用いた場合の既知の問題について述べる (6 章、7 章)。

1.3.1. 本標準の策定の背景と位置づけについて

本標準策定時点において、日本国内では私設総合サービス網交換機 (PINX) と SIP 端末間の SIPS スキームおよび TLS 接続に関する規定が無かった。このため、本標準策定においては、本標準に準拠する私設総合サービス網交換機 (PINX) 、SIP 端末およびプロキシサーバが、SIPS スキームおよび TLS 接続を用いて、セキュアな SIP セッションの確立が可能となるような共通の技術標準を策定することを目的として策定を行った。

1.3.2. 本標準の位置づけ

本標準は、国内において SIPS スキームを利用し、セキュアな SIP セッションの確立を行うことを目的とした私設総合サービス網交換機 (PINX) 、SIP 端末、プロキシサーバにおける最低限の条件を満たす基準仕様として参照されることが期待される。

2. モデル

2.1. SIP で TLS を使用した場合のモデル

この節では SIP における TLS の使い方を簡単に説明する。

2.1.1. サーバ提供の証明書

このモデルでは、TLS ハンドシェイクのときに TLS サーバのみ証明書を提供する。このモデルは、プロキシサーバが TLS サーバ、ユーザエージェント (UA) が TLS クライアントになる場合に、UA とプロキシサーバの間の通信に適用できる。UA はプロキシサーバを TLS で認証するが、プロキシサーバは UA を TLS で認証しない。もしプロキシサーバが UA を認証する必要がある場合は、SIP の HTTP (Hyper Text Transfer Protocol) ダイジェスト認証で実現することができる。またこのモデルでは、TLS コネクションは常に UA によってセットアップされる必要がある (例えば、レジストレーションの時など) 。SIP は双方向のリクエスト (例えば、呼の着信など) を許可しており、UA は TLS コネクションを維持していることが期待される。また、その TLS コネクションは着信および発信のリクエスト両方で使用されることが期待される。

UA が常に TLS コネクションを開始し、維持するという方法は、NAT (Network Address Translation) やファイアウォールを越えるときに発生する問題も同時に解決することができる。既に存在するコネクションを利用して、レスポンスやリクエストを常に UA に到達させることが可能である。

[RFC5626] [5] は、相互運用可能な方法で TLS コネクションの開始とメンテナンスを行うメカニズムを提供する。

2.1.2. 相互認証

このモデルでは、TLS ハンドシェイクのときに TLS クライアントおよび TLS サーバの両方が証明書を提供する。このモデルは UA が証明書を所持している場合に、UA とプロキシサーバの間 (もしくはふたつの UA の間) の通信に適用できる。SIP リクエストを送信するときに適切な TLS コネクションがまだ存在しない場合、ユーザエージェントクライアント (UAC : User Agent Client) は新しい TLS コネクションを確立するための TLS クライアントの役割を担う。SIP リクエストを受信するために TLS コネクションが確立されるとき、ユーザエージェントサーバ (UAS : User Agent Server) は TLS サーバの役割を担う。UA やプロキシサーバはリクエストを送信する場合と受信する場合で、UAC および UAS のどちらにもなるため、相互認証が持つ対称の特質はとても利便性が高い。このモデルでは、TLS コネクションは随時にセットアップおよび破棄され、それ以降のリクエストのために TLS コネクションを維持することは期待されていない。

しかし、このモデルには重要な制限事項がいくつかある。

最初の明らかな制限事項は、お互いに TLS クライアントと TLS サーバのどちらにもなれるように、基礎となる TCP (Transmission Control Protocol) コネクションを双方向から確立可能な環境が必要となることである。これは多くの環境で成り立たない。例えば、NAT やファイアウォールは TCP コネクションの確立を片方向からのみ許可している場合が多い。これは一般的に SIP が運用されるような場所でも同様である。相互認証はこのような環境でも使用すること自体は可能だが、その場合、TLS コネクションは常に同じ側から開始されることが [RFC5626] [5]の 3.1.1 項で説明されている。 [RFC5626] [5] によると、このケースは相互認証の多くのアドバンテージを打ち消してしまう。

2 つ目の明らかな制限事項は、相互認証では両サイドで証明書の交換が必要なことである。この制限事項

は、特に SIP UA の側において、多くの環境で非実用的であると証明されている。なぜならば多数のユーザのための証明書インフラストラクチャを構築することは非常に難易度が高いためである。

これらの理由から、相互認証は主にサーバ間のコミュニケーション (例えば、SIP プロキシ間やプロキシとゲートウェイ、メディアサーバの間など) や双方向で証明書を用いることができる環境 (例えば、企業内で使用される高セキュリティデバイスなど) で用いられている。

2.1.3. SIPS の代わりに TLS with SIP を使用する場合

SIPS URI (Uniform Resource Identifier) は、リクエストが SIP のそれぞれのホップで TLS を使用して送信されることを暗示しているため、”best-effort TLS”には適していない。SIPS URI は”TLS-only”のリクエストの場合にのみ適している。これは [RFC3261] [3] の 26.2.2 項に示される。

> Addresses-of-Record として SIPS URI を配布するユーザは、セキュアではないトランスポートを経由したリクエストを、デバイスで拒否していることがある。

もしだれかが SIP で”best-effort TLS”を使用したい場合、SIP URI を使用し、TLS を使用してリクエストを送信する必要がある。

SIP over TLS の使用はとてもシンプルである。UA が TLS コネクションを開き、SIP メッセージのすべてのヘッダフィールド (From、To、Request-URI、Contact、Route など) において、SIPS URI の代わりに SIP URI を用いれば良い。TLS を用いる場合、Via ヘッダフィールドで TLS であることを示す。

[RFC3261] [3] の 26.3.2.1 項の状態:

> UA がオンラインになり、ローカル管理ドメインにレジストレーションするとき、UA はレジストラと TLS 接続を確立するべきである [SHOULD]。(中略) いったんレジストレーションがレジストラに受け入れられると、レジストラがこの管理ドメインのユーザに対するリクエストを経由するプロキシサーバとしても動作する場合、UA は TLS 接続を開いたままにするべきである [SHOULD] 。既設の TLS 接続は、ちょうどレジストレーションを完了した UA に対してやってくるリクエストを配送するために再利用される。

[RFC5626] [5] では、UA によってのみ TLS コネクションを開始することが出来る環境について、どのようにして TLS コネクションを確立し維持するかの方法を述べている。

同様に、プロキシサーバは TLS コネクションを開くことが出来る場合、たとえ SIPS URI ではなく SIP URI がルートに使用されている場合でも、TLS を用いてリクエストを転送することが出来る。プロキシサーバは、TLS トランスポートを用いたとしても、SIP URI を Record-Route ヘッダフィールドを挿入することができる。[RFC3261] [3] の 26.3.2.2 項では、ドメイン間でどのように TLS を用いることができるかについて説明している。

いくつかの UA やリダイレクトサーバ、プロキシサーバは、SIPS URI が使用されているかどうかとは別に、すべてのコネクションで TLS を強制するというローカルポリシーを持っていることがある。

2.1.4. “transport=tls” URI パラメータと TLS Via パラメータの使用方法

[RFC3261] [3] の 26.2.2 項では、SIPS や SIP URI において”transport=tls”URI パラメータを非推奨としている。:

> SIPS URI スキームにおいて、トランスポートは TLS に依存せず、それゆえ「sips:alice@atlanta.com;transport=tcp」と「sips:alice@atlanta.com;transport=sctp」は共に有効である (そうではあるが、UDP は SIPS で有効なトランスポートではないということに注意) 。「transport=tls」を使用することは、リクエストのひとつのホップにはっきりと限定されてしまうことを理由のひとつとして、結果的に反対された。これは [RFC2543] [6] からの変更点である。

“tls”のパラメータは、[RFC3261] [3] の 25 章の ABNF (Augmented Backus-Naur Form) からは削除されていない。これは、パーサがパラメータを正確に処理するために、ABNF の上位互換性を保つ必要があるためである。”transport=tls”パラメータはこれ以降の RFC で定義されることはないが、 [RFC2543] [6] と [RFC3261] [3] の間のいくつかのインターネットドラフトには存在する。

本仕様では”transport=tls”パラメータは利用しない。

“transport=tls”パラメータの復活や、シングルホップで TLS を使用することを示すための代わりとなるメカニズムは、本仕様の対象外である。

Via ヘッダフィールドは、下記の RFC で各プロトコルが定義されている。

[RFC3261] [3] : ”UDP”、”TCP”、”TLS”、”SCTP”

[RFC4168] [14] : ”TLS-SCTP”

2.2. ホップバイホップセキュリティの検出

SIPS Request-URI の存在は、それぞれのホップでリクエストがセキュアに送信されることを示すためには必須ではない。それでは UAS はどのようにして、リクエストの経路全体で SIPS が使用され、リクエストがセキュアであるかどうかを知ることができるか？事実上、UAS は確実なことを知ることはできない。しかし、[RFC3261] [3] の 26.4.4 項では、UAS でセキュリティを検証するいくつかのチェック項目を設けることを推奨している。加えて、History-Info ヘッダフィールド [RFC4244] [15] では、SIP と SIPS から再帰的に検査することで、検出することが可能である。プロキシサーバによって SIP から SIPS に変更されることは、リクエストの受信者にそのリクエストが各ホップでセキュアに配信されたという、事実とは異なった印象を与えてしまうため、問題がある。

強調しておくとすべてのチェックは、本仕様や [RFC3261] [3] に示されるルールおよび推奨に従わないプロキシサーバや B2BUA (Back-to-Back User Agent) が経路上にいる場合、裏をかくことが可能である。

プロキシサーバは、SIP や SIPS のリクエストのルーティングに対して、それぞれ独自のポリシーを持つ。例えば、ある環境のプロキシサーバは SIPS URI のみ扱うように設定することができる。また、別のプロキシサーバでは、規定を遵守していないリクエストを検出することや、セキュアではないリクエストを拒否する設定が可能である。また、別のプロキシサーバでは、Request-URI や Path、Record-Route、To、From、Contact ヘッダフィールドを検査し、Via ヘッダフィールドに SIPS を強制することが可能である。

[RFC3261] [3] の 26.4.4 項では、発信元 UAC が S/MIME (Secure Multipurpose Internet Mail Extensions) を用いることで、To ヘッダフィールドの元の形式を、end-to-end で確保することもできることが説明されている。[RFC3261] [3] の 26.4.4 項では特に述べられていないが、これは、[RFC3893] [12] が暗号化された重要なヘッダフィールド (To や From のような) と署名された S/MIME のボディの”トンネル”として使用される可能性があることを示している。UAS はこの仕組みを用いることにより、これらの重要なヘッダフィールドを検証することが可能である。このアプローチは確かにルールに準拠してはいるが、より好ましいアプローチは [RFC4474] [16] で定義される SIP ID メカニズムを用いる方法である。SIP ID では、送信者の AOR (Address of Record) (From ヘッダフィールドに含まれる) とオリジナルの宛先の AOR (To ヘッダフィールドに含まれる) を含んだ署名された ID ダイジェストを作成する。

2.3. RFC3261 [3] における SIPS の意味についての問題

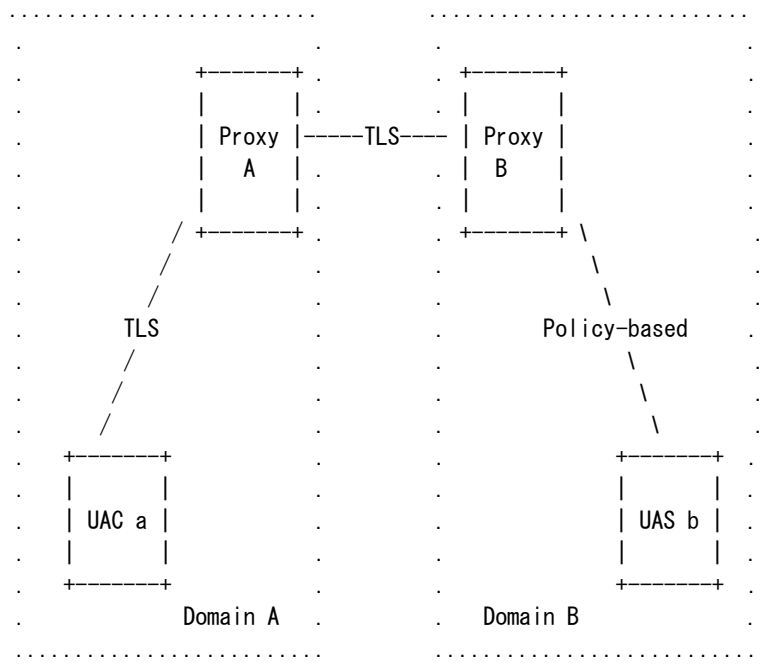
[RFC3261] [3] の 19.1 節では SIPS URI について下記のように説明している:

> SIPS URI はリソースがセキュアにコンタクトされることを指定する。これは特に、UAC とその URI を所持するドメインの間で TLS が使用されることを意味する。そこからユーザに到達するためには、ドメインポリシーに依存した特定のセキュリティメカニズムを用いた、セキュアなコミュニケーションが使用される

26.2.2 項では、Request-URI についてこれを反復している:

> リクエストの Request-URI として使用される場合、SIPS スキームは、Request-URI のドメイン部分に対して責任を負う SIP エンティティに到達するまで、リクエストが転送される各ホップが TLS で安全を確保されなければならないことを示す。当該のドメインに到達すると、ローカルセキュリティとルーティングポリシーにしたがって、かなりの確率で UAS への最後のホップのために、TLS を使用して扱われる。リクエストの発信元によって使用されるとき (リクエストの発信元がターゲットの Address-of-Record として SIPS URI を使用する場合など) 、SIPS は、ターゲットドメインへのリクエストパス全体がセキュアであることを要求する。

典型的な SIP の台形構造を用いて sips:b@B という URI の意味について説明する。実際のドメイン名の代わりに「exapmle.com」や「example.net」を使用し、論理名として「A」と「B」を使用する。



ラストホップに例外を持つ SIP の台形構造

[RFC3261] [3] によると、もし a@A が sips:b@B にリクエストを送ると下記が適用される:

・UA a@A とプロキシ A の間で TLS が要求される
・プロキシ A とプロキシ B の間で TLS が要求される
・ローカルポリシーに応じて、プロキシ B と UA b@B の間で TLS が要求される

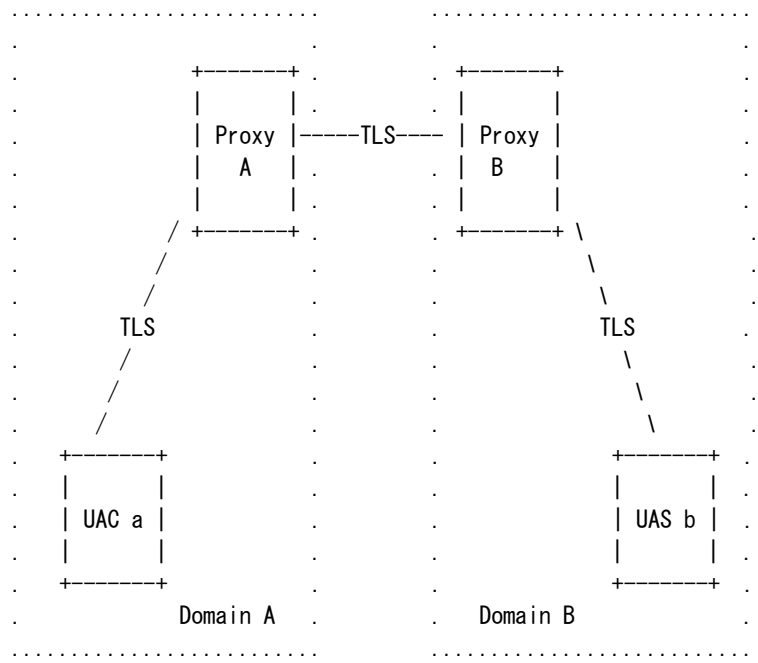
UA a@A とプロキシ A の間で TLS が必須なのにも関わらず、プロキシ B と UA b@B の間で TLS が必須とならないのは何故か疑問に思う人がいる。その主な理由は、 [RFC3261] [3] が [RFC5626] [5] よりも前に書かれているからである。当時は、プロキシ B のみ証明書を所持、UA b は証明書を所持しておらず、プロキシ B は UA b との間に TLS を確立できない可能性があることから、これが多くの実際的な構成であると思われていた。この場合、UA b が TLS サーバの役割となってしまうと、TLS コネクションのリクエストを受け入れることができない。これは、[RFC3261] [3] 準拠の UAS b は、UAC の役割として動作するためにリクエスト送信用の TLS をサポートしなければならないにも関わらず、リクエスト受信用の TLS をサポートする必要がないことに起因する。多くの人が誤って認識しているが、TLS をサポートしていない UAS にリクエストを送信するにときに SIPS URI を使うことを許可するのであれば、最後のホップの例外は作られなかった。最後のホップの例外は、SIPS URI が使用されたときに、TLS で送信せずにリクエストの受信を許可するための試みである。これはリクエストの送信には適用されない。この仕様にいくつかの欠陥があるという根拠は、[RFC5626] [5] で十分な解決策が提供されているからである。[RFC5626] [5] では、UA b が NAT やファイアウォールの後ろにあり、そもそもプロキシ B が TCP セッションを確立できない場合の問題についても解決している。

さらに、ダイアログ内で SIPS を使用することの問題を考察する。もし a@A が Request-URI に SIPS を用いて b@B にリクエストを送信した場合、[RFC3261] [3] の 8.1.1.8 項によると、「Contact ヘッダフィールドに SIPS URI を含まなければならない」。これは、b@B がそのダイアログ内で新しいリクエスト (例えば BYE や re-INVITE など) を送信する場合に、SIPS URI を使わなければならないことを意味する。もし Record-Route

エントリがない場合、もしくは最後の Record-Route エントリが SIPS URI で構成されている場合、そもそも b@B は SIPS を理解していることを予想され、また、TLS のサポートも要求されていることを暗示している。しかし、もし最後の Record-Route エントリが SIP URI だった場合、b は TLS を使用せずにリクエストを送信することができる。(しかし、b はまだメッセージをパースする時に SIPS スキームをハンドルできなければならない。) どちらのケースにしても、b@B から送信されるリクエストの Request-URI は SIPS URI となる。

3. オペレーションの概要

2.3節で述べた問題があるため、本仕様ではリクエストが最後のホップに転送されるときの例外には反対を唱える (詳細は5.3節を参照)。本仕様ではリモートターゲットまでのすべてのホップで、常にTLSが使用されることを保証する。



ラストホップの例外を除いたSIPの台形構造

SIPSスキームは推移的な信頼を暗示している。プロキシサーバの不正行為を妨げる方法が存在しないことは明らかである ([RFC3261] [3] の6.4.4項を参照) 。SIPSはリソースがセキュアにコンタクトされることを依頼するために有用であるが、一方で、実際にリソースがセキュアにコンタクトされたことを示すためには有用ではない。それゆえ、受信したリスエストがRequest-URI (もしくはToヘッダフィールド) にSIPS URIを含んでいたとしても、実際にそのリクエストが各ホップでセキュアに送信されていることが保証されていると推測するのは適切ではない。一部の人たちは、ユーザにそのセッションがセキュアであることを表示するという意味で (例えば、鍵アイコンなど) 、SIPSスキームはHTTPSスキームと同等であると信じたくなる誘惑にかられる。これは明らかに事実とは異なり、それゆえSIPS URIの意図は過大に評価されている。今のところend-to-endでSIPのセキュリティを示すメカニズムは存在しない。他のメカニズムでは、セキュリティレベルをより具体的に示す方法を提供できる。例えば、SIP ID [RFC4474] [16] では、認証IDメカニズムとドメイン間の整合性保護メカニズムを提供している。

Request-URIにSIPS URIが使用される時、実際にリクエストが各ホップでTLSを用いて転送されたという絶対の保障がないのであれば、何故SIPS URIがインターネットのようなグローバルでオープンな環境において便利なのか?と一部の人たちは疑問に思う。SIPをTLSトランスポートで使うだけの場合と何故違うのか？その違いは、SIPS URIをRequest-URIに使用するということは、もしあなたがネットワークの各ホップでTLSの使用を指示しており、かつ、リクエストを拒否することができない場合に、あなたはそのリクエストをTLSなしで配送するよりも、リクエストを失敗させたほうが良いということ意味する点である。SIPS Request-URIの代わりにSIP Request URIでTLSを使用するということは、「ベストエフォート」なサービスであるということを暗示する。このようなリクエストは配送可能ではあるが、必ずしも各ホップでTLSを使用して配送さ

れる必要はない。

もう一つのよくある疑問は、何故TLSの使用を強制する方法として、Proxy-RequireおよびRequireオプションタグを使用しないのか？ということである。その答えは、確かにこれらは機能的にはRequest-URIにSIPSを用いたときと同等であるが、しかし、SIPS URIはレジストレーションのContactヘッダフィールドやダイアログ生成リクエストのContact、Route、Record-Route、Path、From、To、Refer-To、Referred-By、その他の多くのヘッダフィールドで使用可能なためである。また、SIPS URIは人間が使用可能なフォーマット (例えば、名刺やユーザインタフェース) としても提供できる。SIPS URIはSIPS URIを含むことが許可されている他のプロトコルやドキュメントフォーマット (例えば、HTMLなど) でさえも使用することができる。

このドキュメントでSIPSは、SIPS URIで指定されたSIPリソースが、UACとリモートUASの間の各ホップで、TLSを用いてセキュアにコンタクトされることを明示する (プロキシサーバがRequest-URIのターゲットドメインの責任を負うだけであることとは対照的である) 。SIPS URIがUASリソースを識別する時、例えばPSTN網のようなSIP以外のネットワークで起きることは、このドキュメントが定める仕様の対象外である。

3.1. ルーティング

スキームを除いて全く同じである SIP と SIPS URI (例えば、sip:alice@example.com と sips:alice@example.com) は同じリソースを参照する。この要件は [RFC3261] [3] の 19.1 節の「SIP URI で記述されるリソースにおいて、もしそのリソースとの通信がセキュアに行われることを希望する場合、スキームを変更するだけで SIPS URI に‘アップグレード’することができる。」という文章に暗示されている。ただしこれは、SIPS URI が必ず到達可能であるということを意味しているわけではない。特に、もしプロキシサーバがクライアントや他のプロキシサーバとセキュアなコネクションを確立できない場合には到達できないことがある。またこれは、プロキシサーバがリクエストを転送する時に、任意で SIP URI から SIPS URI に‘アップグレード’することを提案しているわけではない (4.3 節を参照) 。むしろこれは、SIP でリソースのアドレス指定が可能なときは、SIPS でもリソースのアドレス指定が可能であることを意味している。

例えば、SIPS の Contact ヘッダフィールドでレジストレーションされた UA のケースを考える。5.3 節および 5.4 節のメッセージ F13 に示されるように、もし UAC がリクエストを SIP の Request-URI でアドレス指定した場合、プロキシサーバもそのリクエストを SIP の Request-URI でアドレス指定し UAS に転送する。プロキシサーバはレジストレーションで使用された SIPS の Request-URI ではなく、SIP の Request-URI を使用して、リクエストを UA に転送する。プロキシサーバは、レジストレーションの時に使用された Contact ヘッダフィールドから SIPS スキームを除いた残りの URI 部分に SIP スキームを付与し、新しい URI として Request-URI に使用することで、これを実現する。もしプロキシサーバがこれを行わず、SIPS を Request-URI に使用してしまった場合、レスポンス (例えば、INVITE に対する 200 OK) は SIPS の Contact ヘッダフィールドを含まなければならなくなってしまう。SIPS の Contact ヘッダフィールドは、他の UA がダイアログ内リクエスト (ACK を含む) を送信する場合にも、SIPS の Contact ヘッダフィールドを使用することを強制されることになってしまうだろう (もし UA が SIPS をサポートしておらず実行不可能な場合においても) 。

本仕様では、プロキシサーバがリクエストを転送する時に、SIPS の Request-URI で記述されたリソースのスキームを変更し SIP URI に “ダウングレード” することや、セキュアではないリンクを通してリクエストを送信してはならない、という義務がある。もしリクエストを“ダウングレード”しなければ拒否される必要がある場合、そのリクエストは 480 (Temporarily Unavailable) のレスポンスで拒否されるだろう (もしかすると Warning ヘッダフィールドにコード 380 “SIPS Not Allowed” が付与する可能性もある) 。また同様に本

仕様では、プロキシがリクエストを転送する時に、SIP の Request-URI で記述されたリソースのスキームを変更し SIPS URI に“アップグレード”してはならないという義務がある (さもないと、プロキシサーバ以降のホップのみ“アップグレード”されており、すべてのホップが SIPS URI というわけではないにも関わらず、UAS にセキュアにルーティングされたという誤解を与えてしまう) 。もしリクエストが“アップグレード”されているという誤解を与えてしまって、リクエストが拒否される必要がある場合、そのリクエストは 480 (Temporarily Unavailable) のレスポンスで拒否されるだろう (もしかすると Warning ヘッダフィールドにコード 381 “SIPS Required”が付与する可能性もある) 。詳細は 4.3 節を参照。

例えば、sip:bob@example.com と sips:bob@example.com という AOR は”example.com”というドメインの“Bob”という同一のユーザを参照する。一つ目の URI が SIP の場合、二つ目の URI が SIPS の場合である。ルーティングの観点では、リクエストが sip:bob@example.com 宛てでも sips:bob@example.com 宛てでも同じルートを経由する。それゆえ、Bob がレジストレーションする時、彼が AOR に SIP を使った場合でも、SIPS を使った場合でも、同じユーザを参照するため問題にはならない。この考え方は一見、[RFC3261] [3] の 19.1.4 項の SIP と SIPS URI は決して同じにはならないという記述と矛盾しているように見える。具体的には、これは REGISTER リクエストにおいて、Contact ヘッダフィールドの URI でレジストレーション情報の紐付けを比較する目的では、同じにならないということを述べている。要はこの文章はレジストレーションに紐づく Contact ヘッダフィールドに適用されるということである。AOR を含む Contact ヘッダフィールドの関連が、ユーザが SIPS URI に到達可能かどうかを決定する。

この例について考えると、もし Bob (AOR は bob@example.com) が SIPS の Contact ヘッダフィールド (例えば、sips:bob@ example.com) でレジストレーションすると、位置登録・位置特定サービスは Bob が sips:bob@example.com と sip:bob@bobphone.example.com に到達可能であることを知る。

もしリクエストが sips:bob@example.com という AOR で送られると、Bob のプロキシサーバは sips:bob@exapmle.com という Request-URI を用いてリクエストを Bob にルーティングする。もしリクエストが sip:bob@example.com という AOR で送られると、Bob のプロキシサーバは sip:bob@example.com という Request-URI を用いてリクエストを Bob にルーティングする。

もし Bob が彼に対するすべてのリクエストが常に TLS を用いて転送されることを保障したい場合、Bob はレジストレーションの時に [RFC5626] [5] を使用することができる。

しかしながら、もし Bob が Contact ヘッダフィールドに SIPS を用いる代わりに SIP (sip:bob@example.com) を用いた場合、Bob の SIPS Contact ヘッダフィールドは存在しないため、sips:bob@example.com という AOR 宛てのリクエストは Bob にはルーティングされないだろうし、また、SIPS から SIP への”ダウングレード”も許可されていない。

コールフローは 5 章を参照。

4. 規定要件

この章は、この仕様によって定義されたすべての規範的な要件を説明する。

4.1. 一般的なユーザーエージェントの動作

4.1.1. UAC の動作

SIPS URI が提示されると、UAC はそれを SIP URI に変更してはならない [MUST NOT] 。
たとえば、ディレクトリエントリに SIPS AOR が含まれている場合、UAC は SIP Request-URI を使用してその AOR に要求を送信することを期待していない。

同様に、ユーザが SIPS URI を持つビジネスカードを読む場合、SIP URI を読み取ることはできない。

SIPS コンタクトヘッダフィールドを含む 3XX レスポンスの場合、リダイレクションの結果としてリクエストを送信するとき、UAC はそれを SIP Request-URI に置き換えることはない (例えば、SIPS スキームを SIP スキームに置き換える) 。

[RFC3261] [3] の 8.1.1.8 項で義務付けられているように、リクエストでは “Request-URI または Top Route ヘッダフィールド値に SIPS URI が含まれている場合、Contact ヘッダフィールドには SIPS URI も含める必要がある” [MUST] 。

(一時的に使用不可能) レスポンスを受信した場合、UAC は SIPS スキームを SIP スキームに自動的に置き換えてリクエストを再試行してはならない [MUST NOT]、 [RFC3261] [3] の 8.1.3.5 項、セキュリティ上の脆弱性が存在します。

UAC が SIP URI を使用して呼を再試行する場合、UAC は、SIPS Request-URI の代わりに SIP Request-URI を使用してセッションを再開始することを認可するように、ユーザから確認を得るべきである [SHOULD] 。

ルートセットが空でない場合 (例えば、レジストラによってサービスルート [RFC3608] [9] が返された場合) 、SIPS Request-URI を使用する場合、すべての SIPS URI からなる Route ヘッダフィールドを使用するのは UAC の責任である。

具体的には、ルートセットに SIP URI が含まれている場合、UAC はリクエストを送信する前にスキームを “sip” から “sips” 変更するだけで、SIP URI を SIPS URI に変更する必要がある [MUST]。

これにより、すべての SIP URI を使用して 1 つのサービスルートを設定または検出し、SIP および SIPS URI の両方に送信要求を許可できる。

UAC が SIP Request-URI を使用している場合、ルートセットが空でなく、一番上の Route ヘッダーフィールドエントリが lr パラメータを持つ SIPS URI である場合、UAC は (SIP Request-URI を使用して) TLS 上でリクエストを送信しなければならない [MUST] 。

ルートが空でなく、Route ヘッダーフィールドエントリが lr パラメータのない SIPS URI である場合、UAC は、ルートセットの最上位エントリに対応する SIPS Request-URI を使用して TLS 上でリクエストを送信しなければならない。

[RFC3261] [3] で既に定義されているものを強調するために、UA は “transport = tls” パラメータを使用してはならない [MUST NOT] 。

4.1.1.1. 登録

UAC は SIPS または SIP AOR のいずれかの Contact ヘッダフィールドで登録します。

UA が SIPS URI で到達可能であることを望む場合、UA は SIPS Contact ヘッダフィールドで登録しなければならない。

SIP または SIPS Request-URI のいずれかを使用してその UA の AOR に宛てられた要求は、その UA にルーティングされる。

これには、SIP と SIPS の両方をサポートする UA が含まれる。

この仕様では、UA がプロキシをプロビジョニングして SIPS Request-URI を使用して要求を転送するだけの SIP ベースのメカニズムは提供していない。

このような選択を提供するために Web インターフェイスなどの非 SIP メカニズムを使用できる。

このような選択を提供するための SIP メカニズムは、この仕様の範囲外である。

UA が SIPS URI で到達したくない場合は、SIP Contact ヘッダフィールドに登録する必要がある。

SIPS Contact ヘッダフィールドに登録することは、SIPS コンタクトと、それに対応する SIP コンタクトの両方を AOR にバインドすることを意味するため、UA は、REGISTER 要求に同じ Contact ヘッダフィールドの SIPS と SIP バージョンの両方を含めてはならない [MUST NOT] 。 この場合、UA は SIPS バージョンのみを使用しなければならない [MUST] 。同様に、SIP Contact ヘッダフィールドが不必要なので、UA は SIP Contact ヘッダフィールドと SIPS Contact ヘッダフィールドの両方を別々のレジストレーションとして登録するべきではない [SHOULD NOT] 。

そうであれば、第 2 の登録は第 1 の登録を置換する (例えば、UA は最初に SIP Contact ヘッダフィールドで登録することができる、それは SIPS をサポートしないことを示す。後に SIPS をサポートする SIPS Contact ヘッダフィールドで登録する) 。

同様に、UA が最初に SIPS Contact ヘッダフィールドで登録し、後で SIP Contact ヘッダフィールドに登録する場合、その SIP Contact ヘッダフィールドは SIPS Contact ヘッダフィールドを置き換える。

登録時に使用された TLS 接続以外でリクエストが配信されないようにしたい場合は、UA が [RFC5626] [5] を使用することができる。

REGISTER 要求のすべての Contact ヘッダフィールドが SIPS である場合、UAC は REGISTER 要求の From および To ヘッダフィールドで SIPS AOR を使用しなければならない [MUST] 。

Contact ヘッダフィールドの少なくとも 1 つが SIPS でない場合 (例えば、sip、mailto、tel、http、https) 、UAC は REGISTER 要求の From フィールドと To ヘッダフィールドで SIP AOR を使用しなければならない [MUST] 。

[RFC3261] [3] ですでに定義されていることを強調するために、UAC は “transport = tls” パラメータを使用してはいけない [MUST NOT] 。

4.1.1.2. ダイアログの SIPS

ダイアログを開始するリクエストの Request-URI が SIP URI である場合、UAC は Contact ヘッダフィールドで何を使用するかについて注意する必要がある (Record-Route がこのホップに使用されない場合) 。

Contact ヘッダフィールドが SIPS URI であった場合、UAS は各ホップでセキュアなトランスポートで送信される中間ダイアログ要求のみを受け入れることを意味する。

この場合 Request-URI は SIP URI であるため、UA がその URI に要求を送信すると、SIPS URI に要求を送信できない可能性がある。

トップ Route ヘッダフィールドに SIPS URI が含まれていない場合、リクエストがセキュアなトランスポートで送信された場合 (たとえば、最初のホップは [RFC5626] [5] のようにプロキシへの TLS 接続を再使用している可能性がある) でも、UAC は Contact ヘッダフィールド内の SIP URI を使用する必要がある [MUST] 。

ダイアログ内でターゲットリフレッシュが発生した場合 (例えば、re-INVITE 要求、UPDATE 要求) 、元のリクエストで SIPS Request-URI を使用した場合、UAC は SIPS URI を持つ Contact ヘッダフィールドを含む必要がある [MUST] 。

4.1.1.3. 派出されたダイアログとトランザクション

セッション、ダイアログ、およびトランザクションは、既存のものから“派生”することができる。

派生ダイアログの良い例は、REFER メソッド [RFC3515] [8] を使用した結果として確立されたものである。

一般的な原則として、派生したダイアログやトランザクションは、関係するエンティティの明示的な承認なしに SIPS を SIP に効果的にダウングレードすることはできない。

たとえば、REFER 要求を使用して呼の転送を実行すると、既存のダイアログが終了し、別の URI が Refer-To URI に基づいて作成される。

その初期ダイアログが SIPS を使用して確立された場合、REFER 要求の受信者によって与えられた明示的な許可がない限り、UAC は SIP を使用して新しいものを確立してはならない [MUST NOT] 。

これは、ユーザに警告が表示される可能性がある。

このような警告は、例えば、セキュアなディレクトリサービスアプリケーションの場合、SIPS をサポートしない UA に要求がルーティングされる可能性があることをユーザに警告するのに役立つ。

REFER 要求は、ダイアログが作成されないリソースを参照するためにも使用できる。

実際、REFER 要求は、元のタイプとは異なるタイプ (すなわち、SIP または SIPS ではない) のリソースを指すために使用することができる。 これに関するセキュリティ上の考慮事項については、[RFC3515] [8] の 5.2 節を参照。

派生したダイアログやトランザクションの他の例は、第三者呼び出し制御 [RFC3725] [10] 、Replace ヘッダフィールド [RFC3891] [11] 、Join ヘッダフィールド [RFC3911] [13] の使用が含まれる。

ここでも、一般的な原則は、これらのメカニズムが適切な認可なしで SIPS の SIP への効果的なダウングレードをもたらすべきではないということである [SHOULD NOT] 。

4.1.1.4. GRUU

グローバルルーティング可能なユーザエージェント URI (GRUU : Globally Routable User Agent URI) [RFC5627] [17] がインスタンス ID / AOR ペアに割り当てられると、SIP GRUU と SIPS GRUU の両方が割り当てられる。

登録によって GRUU が取得されると、REGISTER 要求の Contact ヘッダフィールドに SIP URI が含まれている場合、GRUU の SIP バージョンが返される。

REGISTER リクエストの Contact ヘッダフィールドに SIPS URI が含まれている場合は、GRUU の SIPS バージョンが返される。

GRUU で誤ったスキーム (レジストラのエラー) が受信された場合、UAC はそれを適切なスキームが使用されたものとして扱うべきである [SHOULD]。 (すなわち、GRUU を使用する前に、スキームを適切なスキームで置き換えるべきである [SHOULD]。)

4.1.2. UAS の動作

SIPS URI が提示されると、UAS はそれを SIP URI に変更してはならない[MUST NOT]。

[RFC3261] [3] の 12.1.1 項で規定されているように、:

> ダイアログを開始したリクエストが Request-URI または先頭の Record-Route ヘッダフィールド値に SIPS URI を含んでいた場合、Record-Route ヘッダフィールドが無く、Contact ヘッダフィールドに SIPS URI がある場合は、レスポンスの Contact ヘッダフィールドは SIPS URI でなければならない [MUST]。

UAS が SIPS URI ではなく SIP URI だけで到達することを望まない場合、UAS は 480 (一時的に利用不可) レスポンスで応答しなければならない [MUST]。

UAS は warn-code 380 “SIPS Not Allowed” の Warning ヘッダを含めるべきである [SHOULD]。 [RFC3261] [3] の 8.2.2.1 項は、SIPS URI スキームをまったくサポートしない UAS が “416 (サポートされていない URI スキーム) レスポンスで要求を拒否すべきである [SHOULD] ”と述べている。

UAS が SIP URI に接続するのではなく、SIPS URI で接続することを希望する場合、UAS は、480 (一時的に利用できない) レスポンスで SIP Request-URI へのリクエストを拒否しなければならない [MUST]。

UAS は、warn-code 381 “SIPS Required”を含む Warning ヘッダを含めるべきである [SHOULD]。

UAS が、登録のために使用したものに対応しない URI スキームに宛てられた着信要求を受け入れることは、ローカルポリシーの問題である。

たとえば、“常に SIPS”というポリシーを持つ UA は、TLS 上の SIPS Request-URI を使用してレジストラに対処し、SIPS Contact ヘッダフィールドで登録する。そして UAS は、warn-code 381 “SIPS Required” の Warning ヘッダを持つ 480 (一時的に使用不可) レスポンスで SIP スキームを使用したリクエストを拒否する。

“ベストエフォート型 SIPS”のポリシーを持つ UA は、TLS 上の SIPS Request-URI を使用してレジストラに対処し、SIPS Contact ヘッダフィールドで登録し、UAS は SIP または SIPS Request-URI のいずれかに宛てられたリクエストを受け入れる。

“No SIPS”というポリシーを持つ UA は、SIP Request-URI を使用してレジストラに対処し、TLS を使用できてもできなくても、SIP AOR および SIP Contact ヘッダフィールドで登録し、SIP Request-URI に宛てられた要求を UAS が受け入れる。

URI が矛盾して使用されている (例えば、Request-URI が SIPS URI であるが Contact ヘッダフィールドが SIP URI である) 場合、UAS は 400 (Bad Request) レスポンスでリクエストを拒否しなければならない [MUST]。

ダイアログ内でターゲットリフレッシュが発生した場合 (例えば、re-INVITE 要求、UPDATE 要求)、元のリクエストが SIPS Request-URI を使用した場合には、UAS は SIPS URI を持つ Contact ヘッダフィールドを含む必要がある [MUST]。

[RFC3261] [3] で既に定義されているものを強調するために、UAS は “transport = tls” パラメータを使用してはならない [MUST NOT] 。

4.2. レジストラの動作

UAC は、Contact ヘッダフィールドを SIPS または SIP AOR のいずれかに登録する。

ルーティングの観点からは、どのリソースが同じリソースを識別するために登録に使用されるかは関係ない。

レジストラは、同一の AOR で SIP スキームを有するものと SIPS スキームを有するものがあることを考慮しなければならない [MUST] 。

レジストラは、すべての適切な URI が SIPS スキームである場合に限り、SIPS Contact ヘッダフィールドへのバインディングを受け入れなければならない。 そうしないと、セキュアなリソース (SIPS) とセキュリティ保護されていないリソース (SIP) の間違ったバインドが発生する可能性がある。

これには、Request-URI と Contact とすべての Path ヘッダフィールドが含まれるが、From ヘッダフィールドと To ヘッダフィールドは含まれない。

URI が適切な SIPS スキームでない場合、レジストラは 400 (Bad Request) で REGISTER を拒否しなければならない [MUST] 。

レジストラはサービスルート [RFC3608] [9] を返すことができ、特定のホップで TLS が必須かどうかについて、いくつかの制約を課すことができる。

たとえば、レジストラによって返された Path ヘッダフィールドの一番上のエントリが SIPS URI である場合、レジストラは、Request-URI が SIP であっても、TLS が最初のホップに使用されることを UAC に伝えている。

UA が SIPS Contact ヘッダフィールドで登録した場合、サービスルート [RFC3608] [9] を返すレジストラは、レジストラが、そのクライアントによって送信されたリクエストで SIP と SIPS の両方を使用されることを許可するならば、SIP URI からなるサービスルートを返さなければならない [MUST] 。

UA が SIPS Contact ヘッダフィールドで登録した場合、レジストラがその UA によって送信されたリクエストに SIPS URI のみを使用できるようにする場合、サービスルートを返すレジストラは SIPS URI で構成されるサービスルートを返さなければならない [MUST] 。

4.2.1. GRUU

GRUU [RFC5627] [17] が登録によってインスタンス ID と AOR の組に割り当てられるとき、レジストラは SIP GRUU と SIPS GRUU の両方を割り当てなければならない [MUST] 。

REGISTER リクエストの Contact ヘッダフィールドに SIP URI が含まれている場合、レジストラは GRUU の SIP バージョンを返さなければならない [MUST] 。

REGISTER リクエストの Contact ヘッダフィールドに SIPS URI が含まれている場合、レジストラは GRUU の SIPS バージョンを返さなければならない [MUST] 。

4.3. プロキシの動作

プロキシは、最後のホップに要求を転送または再ターゲットするときに [RFC3261] [3] の最後のホップの例外を使用してはならない [MUST NOT] 。

具体的には、プロキシが SIPS Request-URI でリクエストを受信した場合、プロキシはそのリクエストを SIPS Request-URI に転送またはリターゲットする必要がある [MUST] 。

ターゲット UAS が SIPS Contact ヘッダフィールドの代わりに SIP Contact ヘッダフィールドを使用して以前に登録していた場合、プロキシはその要求を Contact ヘッダフィールドに示された URI に転送してはならない [MUST NOT] 。

プロキシがその理由で要求を拒否する必要がある場合、プロキシは 480 (一時的に利用不可) 応答でそれを拒絶しなければならない [MUST] 。

この場合、プロキシには warn-code 380 “SIPS Not Allowed” の Warning ヘッダを含めるべきである [SHOULD] 。

TLS 接続を設定したり、既存のものを再利用可能な場合には、プロキシは TLS 上で SIP URI を使用してリクエストを転送すべきである [SHOULD] 。

たとえば、[RFC 5626] [5] では、既存の TLS 接続を再利用することができる。

いくつかのプロキシは、TLS 以外の何らかのリクエストを送信することを禁止するポリシーを持つことができる。

プロキシが SIP Request-URI でリクエストを受信した場合、プロキシはそのリクエストを SIPS Request-URI に転送してはならない [MUST NOT] 。

ターゲット UAS が SIPS Contact ヘッダフィールドを使用して以前に登録していて、プロキシがリクエストを転送することを決定した場合、プロキシはその SIPS スキームを SIP スキームに置き換え、残りの URI をそのまま残して、転送された要求の Request-URI として使用しなければならない [MUST] 。

プロキシは UAS に要求を転送するために TLS を使用しなければならない [MUST] 。

一部のプロキシは、UAS が SIPS Contact ヘッダフィールドを使用して登録していた場合、非 SIPS Request-URI を使用するすべての要求で転送しないというポリシーを持つことができる。

プロキシがそのようなポリシーを持っているか、TLS 接続を確立できない場合、プロキシは warn‐code 381 “SIPS Required” という Warning ヘッダを持つ 480 (一時的に利用できない) により拒否するだろう [MAY] 。

URI が矛盾して使用されている (例えば、Request-URI が SIPS URI であるが Contact ヘッダフィールドが SIP URI である) ためにプロキシが要求を拒絶する必要がある場合、プロキシはレスポンスコード 400 (Bad Request) を使用すべきである [SHOULD] 。

TLS 接続を確立するための証明書を提供できない (すなわち、サーバ側認証が使用される) UAC をサポートするために、[RFC5626] [5] で定義されたアウトバウンドプロキシ手順をプロキシが使用することが推奨される。

プロキシが SIPS Request-URI を使用してリクエストを送信し、SIP Contact ヘッダフィールドを持つ 3XX レスポンス、または 416 レスポンス、または warn-code 380 “SIPS Not Allowed” の Warning ヘッダを含む 480 (一時的に使用不可能な) レスポンスを受信したとき、プロキシは応答で再帰してはならない [MUST NOT] 。

この場合、プロキシは、UAC が適切なアクションをとることを可能にするために、再帰の代わりに最適な応答を転送すべきである [SHOULD] 。

プロキシが SIP Request-URI を使用してリクエストを送信し、SIPS Contact ヘッダフィールドを持つ 3XX レスポンス、または warn-code 381 "SIPS Required"の Warning ヘッダを持つ 480 (一時的に使用できない) レスポンスを受信したとき、プロキシはレスポンスを再帰してはならない [MUST NOT] 。

この場合、プロキシは、UAC が適切なアクションをとることを可能にするために、再帰の代わりに最適な応答を転送すべきである [SHOULD] 。

[RFC3261] [3] で既に定義されているものを強調するために、プロキシは “transport = tls” パラメータを使用してはならない [MUST NOT] 。

4.4. リダイレクトサーバの動作

プロキシを使用する代わりに TLS を使用するリダイレクトサーバを使用する場合、いくつかの考慮すべき制限がある。プロキシと UAS ([RFC5626] [5] など) との間に事前に確立された接続がないため、インバウンド接続が許可されるシナリオにのみ適切となる。

たとえば、TLS 相互認証が使用されるサーバとサーバ間の環境 (リダイレクトサーバまたはプロキシサーバ) や NAT トラバーサルの問題がない場所で使用できる。

リダイレクトサーバは、証明書を持たないエンティティにはリダイレクトできない。

リダイレクトサーバは、サーバと UAS 間に NAT がある場合は使用できないことがある。

リダイレクトサーバが SIP Request-URI でリクエストを受信すると、リダイレクトサーバは 3XX レスポンスを SIP または SIPS Contact ヘッダフィールドのどちらかにリダイレクトすることができる [MAY] 。

ターゲット UAS が以前に SIPS Contact ヘッダフィールドを使用して登録していた場合で、リダイレクトサーバが TLS を使用可能な環境にある場合 (前の段落で説明したように) には、SIPS Contact ヘッダフィールドを返すべきである [SHOULD] 。

ターゲット UAS が以前に SIP Contact ヘッダフィールドを使用して登録していた場合、リダイレクトサーバはリクエストをリダイレクトする場合、3XX 応答で SIP Contact ヘッダフィールドを返さなければならない [MUST] 。

リダイレクトサーバが SIPS Request-URI を持つリクエストを受信すると、リダイレクトサーバは 3XX レスポンスを SIP または SIPS Contact ヘッダフィールドにリダイレクトすることができる [MAY] 。

ターゲット UAS が以前に SIPS Contact ヘッダフィールドを使用して登録していた場合で、TLS が使用可能な環境にある時には、リダイレクトサーバは SIPS Contact ヘッダフィールドを返すべきである [SHOULD] 。

ターゲット UAS が以前に SIP Contact ヘッダフィールドを使用して登録していた場合で、リダイレクトを選択した時には、リダイレクトサーバは 3XX 応答で SIP Contact ヘッダフィールドを返さなければならない [MUST] 。 さもなければ、UAS は warn-code 380 の “SIPS Not Allowed” の Warning ヘッダを持つ 480 (一時的に利用できない) 応答で要求を拒否してもよい [MAY] 。

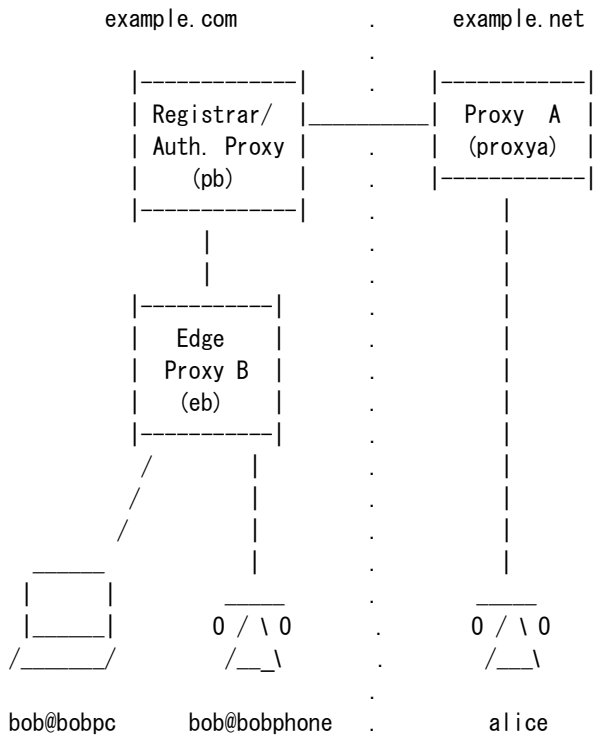
リダイレクトサーバが情報を持たない (例えば、異なるドメインの AOR) UAS にリダイレクトする場合、Contact ヘッダフィールドはどのスキームであってもよい。

一貫して使用されないためにリダイレクトサーバが要求を拒否する必要がある場合、リダイレクトサーバはレスポンスコード 400 (Bad Request) を使用する必要がある [SHOULD] 。

[RFC3261] [3] で既に定義されていることを強調するために、リダイレクトサーバは “transport = tls” パラメータを使用してはならない [MUST NOT] 。

5. 呼の流れ

以下の図は、この章の例として使われるトポロジーを説明します。



トポロジー

以下の例では、Bob が二つのクライアントを持っていて、一つは彼のコンピュータ上で動作する SIP PC クライアントで、もう一つは SIP 電話機である。PC クライアントは SIPS をサポートしておらず、そのため、SIP コンタクトヘッダーフィールドを伴った登録のみできる。しかしながら、SIP 電話機は SIPS と TLS をサポートしている。そのため、SIPS コンタクトヘッダーフィールドを伴った登録ができる。Bob のデバイスの両方がエッジプロキシーB を経由していくと、その結果、eb.example.com を示すルートヘッダーフィールドを含むことになる。エッジプロキシーB は自身に対応しているルートヘッダーフィールドを削除して、パスヘッダーフィールドに自身を追加する。登録プロセスの呼の流れは 5.1 節に説明される。

登録後は、bob@example.com の Bob の AOR に以下の 2 つの接続が関連付けられる。

・sips:bob@bobphone.example.com

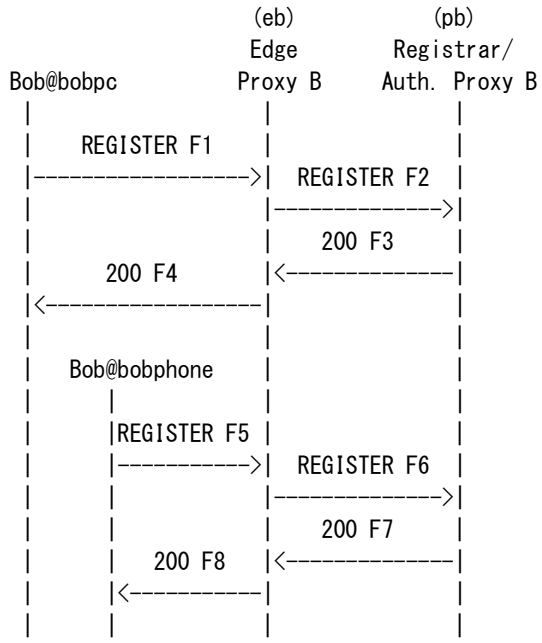
・sip:bob@bobpc.example.com

その時、Alice が彼女自身のプロキシーA 経由で Bob に電話をかける。プロキシーA は Bob のドメイン：example.com にたどり着く。この例では、そのドメインは Bob の登録/認証プロキシーB としている。プロキシーA は自分自身に関連するルートヘッダーフィールドを削除し、レコードルートヘッダーフィールドに自分自身を追加して、登録/認証プロキシーB に要求を転送する。

以下の章では登録と 2 つの例を説明する。最初の例 5.2 節) では Alice が Bob の SIPS AOR を呼びます。第 2 の例 (5.3 節) では Alice が TCP 転送を使用して Bob の SIP AOR を呼びます。第 3 の例 (5.4 節) では Alice が TLS 転送を使用して、Bob の SIP AOR を呼びます。

5.1. Bob が彼のコンタクトを登録する

この図では Bob の機器が登録するプロセスを説明します。彼の PC クライアント (Bob@bobpc) は SIP スキーム (手順) で登録する。そして、彼の SIP 電話 (Bob@phone) は SIPS スキームで登録する。



Bob Registers His Contacts

メッセージ詳細

F1 REGISTER Bob's PC Client -> Edge Proxy B

```
REGISTER sip:pb.example.com SIP/2.0
Via: SIP/2.0/TCP bobpc.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, outbound
Route: <sip:eb.example.com;lr>
Contact: <sip:bob@bobpc.example.com>
  ;+sip.instance="<urn:uuid:0C67446E-F1A1-11D9-94D3-000A95A0E128>"
  ;reg-id=1
Content-Length: 0
```

F2 REGISTER Edge Proxy B -> Registrar/Authoritative Proxy B

```
REGISTER sip:pb.example.com SIP/2.0
Via: SIP/2.0/TCP eb.example.com:5060;branch=z9hG4bK87asdks7
```

Via: SIP/2.0/TCP bobpc.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, outbound

Path: <sip:laksdyjanseg237+fsdf+uy623hytIJ8@eb.example.com;lr;ob>

Contact: <sip:bob@bobpc.example.com>

;+sip.instance="<urn:uuid:0C67446E-F1A1-11D9-94D3-000A95A0E128>"

;reg-id=1

Content-Length: 0

F3 200 (REGISTER) Registrar/Authoritative Proxy B -> Edge Proxy B

SIP/2.0 200 OK

Via: SIP/2.0/TCP eb.example.com:5060;branch=z9hG4bK87asdks7

Via: SIP/2.0/TCP bobpc.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

Require: outbound

Supported: path, outbound

Path: <sip:laksdyjanseg237+fsdf+uy623hytIJ8@eb.example.com;lr;ob>

Contact: <sip:bob@bobphone.example.com>

;+sip.instance="<urn:uuid:0C67446E-F1A1-11D9-94D3-000A95A0E128>"

;reg-id=1

;expires=3600

Date: Mon, 12 Jun 2006 16:43:12 GMT

Content-Length: 0

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

SIP/2.0 200 OK

Via: SIP/2.0/TCP bobpc.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

Require: outbound

Supported: path, outbound

Path: <sip:laksdyjanseg237+fsdf+uy623hytIJ8@eb.example.com;lr;ob>

Contact: <sip:bob@bobphone.example.com>
  ;+sip.instance="<urn:uuid:0C67446E-F1A1-11D9-94D3-000A95A0E128>"
  ;reg-id=1
  ;expires=3600
Date: Thu, 09 Aug 2007 16:43:12 GMT
Content-Length: 0

F5 REGISTER Bob's Phone -> Edge Proxy B

REGISTER sips:pb.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, outbound
Route: <sips:eb.example.com;lr>
Contact: <sips:bob@bobphone.example.com>
  ;+sip.instance="<urn:uuid:6F85D4E3-E8AA-46AA-B768-BF39D5912143>"
  ;reg-id=1
Content-Length: 0

F6 REGISTER Edge Proxy B -> Registrar/Authoritative Proxy B

REGISTER sips:pb.example.com SIP/2.0
Via: SIP/2.0/TLS eb.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, outbound
Path: <sips:psodkfsj+34+kklsL+uJH-Xm816k09Kk@eb.example.com;lr;ob>
Contact: <sips:bob@bobphone.example.com>
  ;+sip.instance="<urn:uuid:6F85D4E3-E8AA-46AA-B768-BF39D5912143>"
  ;reg-id=1
Content-Length: 0

F7 200 (REGISTER) Registrar/Authoritative Proxy B -> Edge Proxy B

SIP/2.0 200 OK

Via: SIP/2.0/TLS eb.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
Require: outbound
Supported: path, outbound
Path: <sips:psodkfsj+34+kklsL+uJH-Xm816k09Kk@eb.example.com;lr;ob>
Contact: <sips:bob@bobphone.example.com>
  ;+sip.instance="<urn:uuid:6F85D4E3-E8AA-46AA-B768-BF39D5912143>"
  ;reg-id=1
  ;expires=3600
Date: Thu, 09 Aug 2007 16:43:50 GMT
Content-Length: 0

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

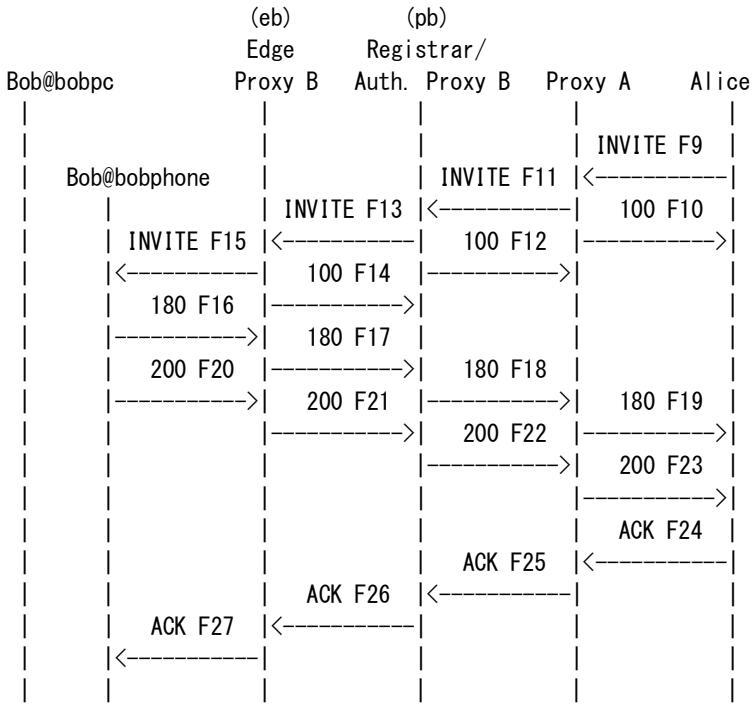
SIP/2.0 200 OK
Via: SIP/2.0/TLS bobphone.example.com:5061;branch=z9hG4bK9555
To: Bob <sips:bob@example.com>;tag=5150
From: Bob <sips:bob@example.com>;tag=90210
Call-ID: faif9a@qwefnwdclk
CSeq: 12 REGISTER
Require: outbound
Supported: path, outbound
Path: <sips:psodkfsj+34+kklsL+uJH-Xm816k09Kk@eb.example.com;lr;ob>
Contact: <sips:bob@bobphone.example.com>
  ;+sip.instance="<urn:uuid:6F85D4E3-E8AA-46AA-B768-BF39D5912143>"
  ;reg-id=1
  ;expires=3600
Date: Thu, 09 Aug 2007 16:43:50 GMT
Content-Length: 0

5.2. Alice が Bob の SIPS AOR を呼ぶ

Bob の登録は 5.1 節に従って既に行われている。

この最初の例では、Alice が Bob の SIPS AOR (sips:bob@example.com) を呼ぶ。登録/認証プロキシーB は登録データベースから関連付けを解決し、2 つのコンタクトヘッダーフィールドの関連を見つける。Alice は SIPS Request-URI (sips:bob@example.com) で Bob を呼んでいるので、登録/認証プロキシーB は呼が bobphone (SIPS コンタクトヘッダーフィールドを使用して登録された) にのみ、転送される必要があることを決定する。そして、呼の要求はエッジプロキシーB を経由して、sips:bob@bobphone.example.com にのみ送られる。登録/認証プロキシーB とエッジプロキシーB の両方がレコードルートヘッダーフィールドにそれら自身を挿

入する。Bob は sips:bob@bobphone.example.com で応答する。



Alice Calls Bob's SIPS AOR

メッセージ詳細

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

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 -> Registrar/Authoritative 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: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) Registrar/Authoritative 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 Registrar/Authoritative Proxy B -> Edge Proxy B

INVITE sips:bob@bobphone.example.com SIP/2.0
Via: SIP/2.0/TLS pb.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
Route:
 <sips:psodkfsj+34+kklsL+uJH-Xm816k09Kk@edge.example.com;lr;ob>

Record-Route: <sips:pb.example.com;lr>, <sips: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) Edge Proxy B -> Registrar/Authoritative Proxy B

SIP/2.0 100 Trying

Via: SIP/2.0/TLS pb.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 INVITE Edge Proxy B -> Bob's phone

INVITE sips:bob@bobphone.example.com SIP/2.0

Via: SIP/2.0/TLS eb.example.com:5061;branch=z9hG4bKbiba

Via: SIP/2.0/TLS pb.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: 67

To: Bob <sips:bob@example.com>

From: Alice <sips:alice@example.net>;tag=8675309

Call-ID: lzksjf8723k@sodk6587

CSeq: 1 INVITE

Record-Route:

<sips:psodkfsj+34+kklsL+uJH-Xm816k09Kk@eb.example.com;lr;ob>,

<sips:pb.example.com;lr>, <sips:proxya.example.net;lr>

Contact: <sips:alice@alice-1.example.net>

Content-Type: application/sdp

Content-Length: {as per SDP}

{SDP not shown}

F16 180 (INVITE) Bob's Phone -> Edge Proxy B

SIP/2.0 180 Ringing

Via: SIP/2.0/TLS eb.example.com:5061;branch=z9hG4bKbiba

Via: SIP/2.0/TLS pb.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:psodkfsj+34+kklsL+uJH-Xm816k09Kk@eb.example.com;lr;ob>,

<sips:pb.example.com;lr>, <sips:proxya.example.net;lr>

Contact: <sips:bob@bobphone.example.com>

Content-Length: 0

F17 180 (INVITE) Edge Proxy B -> Registrar/Authoritative Proxy B

SIP/2.0 180 Ringing

Via: SIP/2.0/TLS pb.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:psodkfsj+34+kklsL+uJH-Xm816k09Kk@eb.example.com;lr;ob>,

<sips:pb.example.com;lr>, <sips:proxya.example.net;lr>

Contact: <sips:bob@bobphone.example.com>

Content-Length: 0

F18 180 Registrar/Authoritative Proxy B -> Proxy A

SIP/2.0 180 Ringing

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:psodkfsj+34+kklsL+uJH-Xm816k09Kk@eb.example.com;lr;ob>,

<sips:pb.example.com;lr>, <sips:proxya.example.net;lr>

Contact: <sips:bob@bobphone.example.com>

Content-Length: 0

F19 180 (INVITE) Proxy A -> Alice

SIP/2.0 180 Ringing
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:psodkfsj+34+kklsL+uJH-Xm816k09Kk@eb.example.com;lr;ob>,
 <sips:pb.example.com;lr>, <sips:proxya.example.net;lr>
Contact: <sips:bob@bobphone.example.com>
Content-Length: 0

F20 200 (INVITE) Bob's Phone -> Edge Proxy B

SIP/2.0 200 OK
Via: SIP/2.0/TLS eb.example.com:5061;branch=z9hG4bKbiba
Via: SIP/2.0/TLS pb.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:psodkfsj+34+kklsL+uJH-Xm816k09Kk@eb.example.com;lr;ob>,
 <sips:pb.example.com;lr>, <sips:proxya.example.net;lr>
Contact: <sips:bob@bobphone.example.com>
Content-Length: 0

F21 200 (INVITE) Edge Proxy B -> Registrar/Authoritative Proxy B

SIP/2.0 200 OK
Via: SIP/2.0/TLS pb.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:psodkfsj+34+kklsL+uJH-Xm816k09Kk@eb.example.com;lr;ob>,

<sips:pb.example.com;lr>, <sips:proxya.example.net;lr>
Contact: <sips:bob@bobphone.example.com>
Content-Length: 0

F22 200 Registrar/Authoritative 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:psodkfsj+34+kklsL+uJH-Xm816k09Kk@eb.example.com;lr;ob>,
<sips:pb.example.com;lr>, <sips:proxya.example.net;lr>
Contact: <sips:bob@bobphone.example.com>
Content-Length: 0

F23 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:psodkfsj+34+kklsL+uJH-Xm816k09Kk@eb.example.com;lr;ob>,
<sips:pb.example.com;lr>, <sips:proxya.example.net;lr>
Contact: <sips:bob@bobphone.example.com>
Content-Length: 0

F24 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:proxya.example.net;lr>, <sips:pb.example.com;lr>,

<sips:psodkfsj+34+kklsL+uJH-Xm816k09Kk@pb.example.com;lr;ob>
Content-Length: 0

F25 ACK Proxy A -> Registrar/Authoritative 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:pb.example.com;lr>,
<sips:psodkfsj+34+kklsL+uJH-Xm816k09Kk@pb.example.com;lr;ob>
Content-Length: 0

F26 ACK Registrar/Authoritative Proxy B -> Edge Proxy B

ACK sips:bob@bobphone.example.com SIP/2.0
Via: SIP/2.0/TLS pb.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: 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:pb.example.com;lr>,
<sips:psodkfsj+34+kklsL+uJH-Xm816k09Kk@pb.example.com;lr;ob>
Content-Length: 0

F27 ACK Proxy B -> Bob's Phone

ACK sips:bob@bobphone.example.com SIP/2.0
Via: SIP/2.0/TLS eb.example.com:5061;branch=z9hG4bKkmfdgk
Via: SIP/2.0/TLS pb.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

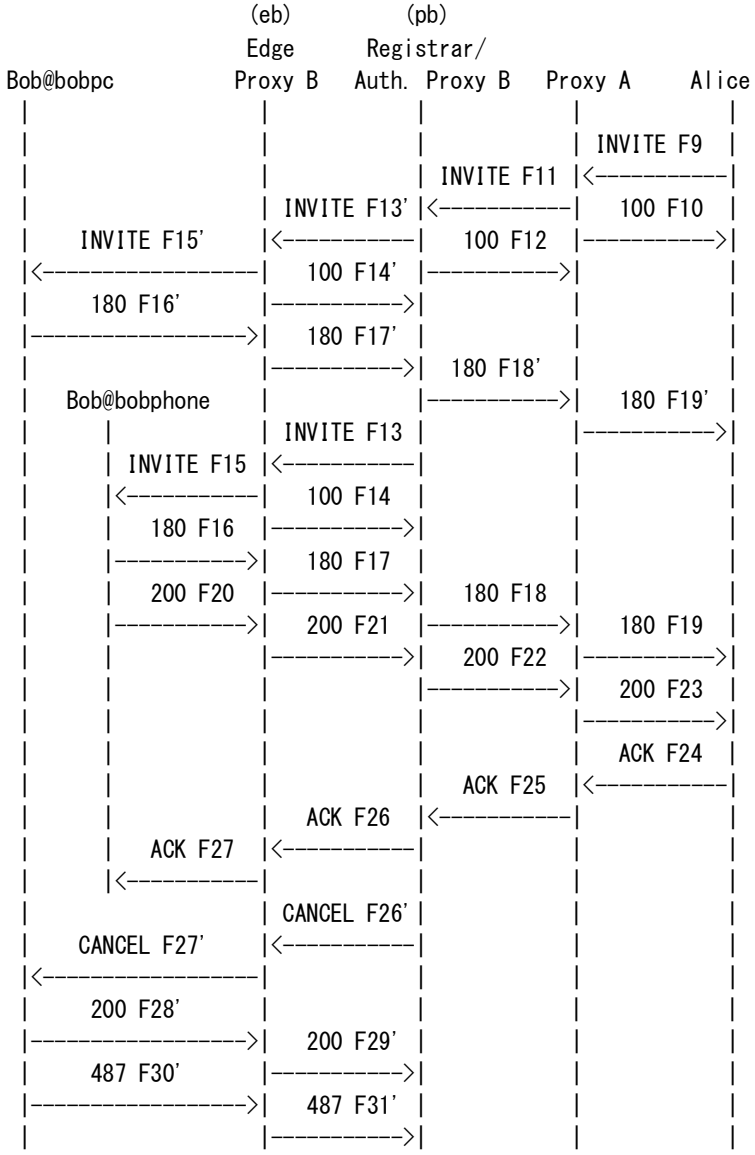
CSeq: 1 ACK

Content-Length: 0

5.3. Alice が TCP を使用して Bob の SIP AOR を呼ぶ

Bob の登録は既に 5.1 節で行われている。

第 2 の例では Alice が代わりに Bob の SIP AOR を呼びます。また、トランスポートとして TCP を使用する。登録/認証プロキシーB は登録データベースを参照し、2 つのコンタクトヘッダーフィールドの関連を見つけます。Alice は SIP Request-URI (sip:bob@example.com) で呼んでいるので、登録/認証プロキシーB は、呼が bobpc (SIP コンタクトヘッダーフィールドで登録された) と bobphone (SIPS コンタクトヘッダーフィールドで登録された) の両方に転送される必要があると決定する。そして、その要求を、エッジプロキシーB を経由して、sip:bob@bobpc.example.com と sip:bob@bobphone.example.com に分割する。注意すべきは、登録のために使われるコンタクトヘッダーフィールドの Request-URI の SIPS スキームが SIP スキームとして扱われていることである。登録/認証プロキシーB とエッジプロキシーB の両方はレコードルートヘッダーに自分自身を挿入する。Bob の電話のポリシーは SIP と SIPS (例えば"best effort") の呼を受け付ける。そのため、彼の PC クライアントと SIP 電話機を同時に呼び出した。Bob は SIP 電話機で応答した。そして、PC クライアントに分割された呼はキャンセルされた。



Alice Calls Bob's SIP AOR

メッセージ詳細

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
To: Bob <sip:bob@example.com>
From: Alice <sip:alice@example.net>;tag=8675309
Call-ID: lzksjf8723k@sodk6587
CSeq: 1 INVITE
Content-Length: 0

F11 INVITE Proxy A -> Registrar/Authoritative 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: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) Registrar/Authoritative 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 Registrar/Authoritative Proxy B -> Edge Proxy B

INVITE sip:bob@bobpc.example.com SIP/2.0
Via: SIP/2.0/TCP pb.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

Route: <sip:laksdyjanseg237+fsdf+uy623hytIJ8@eb.example.com;lr;ob>

Record-Route: <sip:pb.example.com;lr>, <sip: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) Edge Proxy B -> Registrar/Authoritative Proxy B

SIP/2.0 100 Trying

Via: SIP/2.0/TCP pb.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' INVITE Edge Proxy B -> Bob's PC Client

INVITE sip:bob@bobpc.example.com SIP/2.0

Via: SIP/2.0/TCP eb.example.com:5060;branch=z9hG4bKbiba

Via: SIP/2.0/TCP pb.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: 67

To: Bob <sip:bob@example.com>

From: Alice <sip:alice@example.net>;tag=8675309

Call-ID: lzksjf8723k@sodk6587

CSeq: 1 INVITE

Record-Route:

<sip:laksdyjanseg237+fsdf+uy623hytIJ8@eb.example.com;lr;ob>,

<sip:pb.example.com;lr>, <sip:proxya.example.net;lr>

Contact: <sip:alice@alice-1.example.net>

Content-Type: application/sdp

Content-Length: {as per SDP}
{SDP not shown}

F16' 180 (INVITE) Bob's PC Client -> Edge Proxy B

SIP/2.0 180 Ringing
Via: SIP/2.0/TCP eb.example.com:5060;branch=z9hG4bKbiba
Via: SIP/2.0/TCP pb.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:laksdyjanseg237+fsdf+uy623hytIJ8@eb.example.com;lr;ob>,
<sip:pb.example.com;lr>, <sip:proxya.example.net;lr>
Contact: <sip:bob@bobpc.example.com>
Content-Length: 0

F17' 180 (INVITE) Edge Proxy B -> Registrar/Authoritative Proxy B

SIP/2.0 180 Ringing
Via: SIP/2.0/TCP pb.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:laksdyjanseg237+fsdf+uy623hytIJ8@eb.example.com;lr;ob>,
<sip:pb.example.com;lr>, <sip:proxya.example.net;lr>
Contact: <sip:bob@bobpc.example.com>
Content-Length: 0

F18' 180 (INVITE) Registrar/Authoritative Proxy B -> Proxy A

SIP/2.0 180 Ringing
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:laksdyjanseg237+fsdf+uy623hytIJ8@eb.example.com;lr;ob>,

<sip:pb.example.com;lr>, <sip:proxya.example.net;lr>

Contact: <sip:bob@bobpc.example.com>

Content-Length: 0

F19' 180 (INVITE) Proxy A -> Alice

SIP/2.0 180 Ringing

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:laksdyjanseg237+fsdf+uy623hytIJ8@eb.example.com;lr;ob>,

<sip:pb.example.com;lr>, <sip:proxya.example.net;lr>

Contact: <sip:bob@bobpc.example.com>

Content-Length: 0

F13 INVITE Registrar/Authoritative Proxy B -> Edge Proxy B

INVITE sip:bob@bobphone.example.com SIP/2.0

Via: SIP/2.0/TCP pb.example.com:5060;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

Route: <sip:psodkfsj+34+kklsL+uJH-Xm816k09Kk@eb.example.com;lr;ob>

Record-Route: <sip:pb.example.com;lr>, <sip: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) Edge Proxy B -> Registrar/Authoritative Proxy B

SIP/2.0 100 Trying

Via: SIP/2.0/TCP pb.example.com:5060;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 INVITE Edge Proxy B -> Bob's Phone

INVITE sip:bob@bobphone.example.com SIP/2.0

Via: SIP/2.0/TLS eb.example.com:5061;branch=z9hG4bKtroubaba

Via: SIP/2.0/TCP pb.example.com:5060;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:

<sip:psodkfsj+34+kklsL+uJH-Xm816k09Kk@eb.example.com;lr;ob>,

<sip:pb.example.com;lr>, <sip:proxya.example.net;lr>

Contact: <sip:alice@alice-1.example.net>

Content-Type: application/sdp

Content-Length: {as per SDP}

{SDP not shown}

F16 180 (INVITE) Bob's Phone -> Edge Proxy B

SIP/2.0 180 Ringing

Via: SIP/2.0/TLS eb.example.com:5061;branch=z9hG4bKtroubaba

Via: SIP/2.0/TCP pb.example.com:5060;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:

<sip:psodkfsj+34+kklsL+uJH-Xm816k09Kk@eb.example.com;lr;ob>,

<sip:pb.example.com;lr>, <sip:proxya.example.net;lr>

Contact: <sip:bob@bobphone.example.com>
Content-Length: 0

F17 180 (INVITE) Edge Proxy B -> Registrar/Authoritative Proxy B

SIP/2.0 180 Ringing
Via: SIP/2.0/TCP pb.example.com:5060;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:
 <sip:psodkfsj+34+kklsL+uJH-Xm816k09Kk@eb.example.com;lr;ob>,
 <sip:pb.example.com;lr>, <sip:proxya.example.net;lr>
Contact: <sip:bob@bobphone.example.com>
Content-Length: 0

F18 180 (INVITE) Registrar/Authoritative Proxy B -> Proxy A

SIP/2.0 180 Ringing
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:
 <sip:psodkfsj+34+kklsL+uJH-Xm816k09Kk@eb.example.com;lr;ob>,
 <sip:pb.example.com;lr>, <sip:proxya.example.net;lr>
Contact: <sip:bob@bobphone.example.com>
Content-Length: 0

F19 180 (INVITE) Proxy A -> Alice

SIP/2.0 180 Ringing
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:

<sip:psodkfsj+34+kklsL+uJH-Xm816k09Kk@eb.example.com;lr;ob>,
<sip:pb.example.com;lr>, <sip:proxya.example.net;lr>
Contact: <sip:bob@bobphone.example.com>
Content-Length: 0

F20 200 (INVITE) Bob's Phone -> Edge Proxy B

SIP/2.0 200 OK
Via: SIP/2.0/TLS eb.example.com:5061;branch=z9hG4bKtroubaba
Via: SIP/2.0/TCP pb.example.com:5060;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:
<sip:psodkfsj+34+kklsL+uJH-Xm816k09Kk@eb.example.com;lr;ob>,
<sip:pb.example.com;lr>, <sip:proxya.example.net;lr>
Contact: <sip:bob@bobphone.example.com>
Content-Length: 0

F21 200 (INVITE) Edge Proxy B -> Registrar/Authoritative Proxy B

SIP/2.0 200 OK
Via: SIP/2.0/TCP pb.example.com:5060;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:
<sip:psodkfsj+34+kklsL+uJH-Xm816k09Kk@eb.example.com;lr;ob>,
<sip:pb.example.com;lr>, <sip:proxya.example.net;lr>
Contact: <sip:bob@bobphone.example.com>
Content-Length: 0

F22 200 (INVITE) Registrar/Authoritative 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:
 <sip:psodkfsj+34+kklsL+uJH-Xm816k09Kk@eb.example.com;lr;ob>,
 <sip:pb.example.com;lr>, <sip:proxya.example.net;lr>
Contact: <sip:bob@bobphone.example.com>
Content-Length: 0

F23 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:
 <sip:psodkfsj+34+kklsL+uJH-Xm816k09Kk@eb.example.com;lr;ob>,
 <sip:pb.example.com;lr>, <sip:proxya.example.net;lr>
Contact: <sip:bob@bobphone.example.com>
Content-Length: 0

F24 ACK Alice -> Proxy A

ACK sip:bob@bobphone.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>;tag=5551212
From: Alice <sip:alice@example.net>;tag=8675309
Call-ID: lzksjf8723k@sodk6587
CSeq: 1 ACK
Route: <sip:proxya.example.net;lr>, <sip:pb.example.com;lr>,
 <sip:psodkfsj+34+kklsL+uJH-Xm816k09Kk@edge.example.com;lr;ob>
Content-Length: 0

F25 ACK Proxy A -> Registrar/Authoritative Proxy B

ACK sip: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:pb.example.com;lr>,
<sip:psodkfsj+34+kklsL+uJH-Xm816k09Kk@eb.example.com;lr;ob>
Content-Length: 0

F26 ACK Registrar/Authoritative Proxy B -> Edge Proxy B

ACK sip:bob@bobphone.example.com SIP/2.0
Via: SIP/2.0/TCP pb.example.com:5060;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: 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:psodkfsj+34+kklsL+uJH-Xm816k09Kk@eb.example.com;lr;ob>
Content-Length: 0

F27 ACK Proxy B -> Bob's Phone

ACK sip:bob@bobphone.example.com SIP/2.0
Via: SIP/2.0/TLS eb.example.com:5061;branch=z9hG4bKtroubaba
Via: SIP/2.0/TCP pb.example.com:5060;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-Length: 0

F26' CANCEL Registrar/Authoritative Proxy B -> Edge Proxy B

CANCEL sip:bob@bobpc.example.com SIP/2.0
Via: SIP/2.0/TCP pb.example.com:5060;branch=z9hG4bKbalouba.2
Max-Forwards: 70
To: Bob <sip:bob@example.com>
From: Alice <sip:alice@example.net>;tag=8675309

Call-ID: lzksjf8723k@sodk6587

CSeq: 1 CANCEL

Route: <sip:psodkfsj+34+kklsL+uJH-Xm816k09Kk@eb.example.com;lr;ob>

Content-Length: 0

F27' CANCEL Edge Proxy B -> Bob's PC Client

CANCEL sip:bob@bobpc.example.com SIP/2.0

Via: SIP/2.0/TCP eb.example.com:5060;branch=z9hG4bKtroubaba

Via: SIP/2.0/TCP pb.example.com:5060;branch=z9hG4bKbalouba.2

Max-Forwards: 69

To: Bob <sip:bob@example.com>

From: Alice <sip:alice@example.net>;tag=8675309

Call-ID: lzksjf8723k@sodk6587

CSeq: 1 CANCEL

Content-Length: 0

F28' 200 (CANCEL) Bob's PC Client -> Edge Proxy B

SIP/2.0 200 OK

Via: SIP/2.0/TCP eb.example.com:5060;branch=z9hG4bKtroubaba

Via: SIP/2.0/TCP pb.example.com:5060;branch=z9hG4bKbalouba.2

To: Bob <sip:bob@example.com>

From: Alice <sip:alice@example.net>;tag=8675309

Call-ID: lzksjf8723k@sodk6587

CSeq: 1 CANCEL

Content-Length: 0

F29' 200 (CANCEL) Edge Proxy B -> Registrar/Authoritative Proxy B

SIP/2.0 200 OK

Via: SIP/2.0/TCP pb.example.com:5060;branch=z9hG4bKbalouba.2

To: Bob <sip:bob@example.com>

From: Alice <sip:alice@example.net>;tag=8675309

Call-ID: lzksjf8723k@sodk6587

CSeq: 1 CANCEL

Content-Length: 0

F30' 487 (INVITE) Bob's PC Client -> Edge Proxy B

SIP/2.0 487 Request Terminated

Via: SIP/2.0/TCP eb.example.com:5060;branch=z9hG4bKtroubaba

Via: SIP/2.0/TCP pb.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

F31' 487 (INVITE) Edge Proxy B -> Registrar/Authoritative Proxy B

SIP/2.0 487 Request Terminated

Via: SIP/2.0/TCP pb.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

5.4. Alice が TLS を使用して Bob の SIP AOR で呼ぶ

Bob の登録は既に 5.1 節で行われている。

第 3 の例は、Alice がプロキシーに接続するためにトランスポートとして TLS を使って第 2 の例を示している。もし、Alice の UA が TLS をサポートしており、プロキシー ([RFC5626] [5] を使用した場合) への 1 つの接続を望むのであれば、その形態は普通であるだろう。下の例において、プロキシーA はプロキシーB と外部接続の通信のためにトランスポートとして TLS を使用している。しかし、それは必ずしも必要なケースではない。

Request-URI に SIP URI が使われ、要求を送信するためにトランスポートとして TLS が使われたとき、Via ヘッダーフィールドは TLS を示す。ルートヘッダーフィールド (もしあるならば) は一般に、SIP URI を使う (しかし、それは SIPS URI でもあるかもしれない)。 コンタクトヘッダーフィールドおよび To、From ヘッダーフィールドは、普通、SIP URI を示す 。

呼の流れは正確には 2 番目の例に従ってあるべきある (5.3 節) 。 唯一の違いがあるとすると、Via ヘッダーフィールドが TLS Via パラメータを使用することだろう。URIs には SIP URIs を残すが、SIPS URIs は残さないだろう。

6. 更なる問題

SIP [RFC3261] [3] 自身は、SIPS を使用するのにいくつかの複雑さがある。例えば、Record-Route が未使用のときである。SIPS URI が dialog-initiating リクエストにおける Contact ヘッダーフィールドに使用され、Record-Route が使用されていないとき、その SIPS URI はもう一方のエンドポイントまで使用できないかもしれない。もう一方のエンドポイントが SIPS そして/または TLS をサポートしないと、SIPS URI は使用できないだろう。last-hop の例外は、これが発生することが可能な時の例である。この場合、Record-Route を使用すること、すなわちプロキシを通してリクエストを送るのは、SIPS を働かせるのを手伝うことができる。

別の例は、Contact ヘッダーフィールドが SIPS URI であり、どのような Record-Route も未使用で、遠端が SIPS と TLS をサポートする場合、遠端で証明書を有効にすることができないなら SIP 発信側エンドポイントとの TLS 接続を確立することがまだ不可能かもしれない。発信側エンドポイントが以下で説明されるようなサーバサイド認証を使用し、または、発信側エンドポイントが有効にすることができる証明書を使用しないことは、通常にあり得るかもしれない。

TLS 自身は、どのように SIPS を使用可能にするか、重要な影響を与える。通常、サーバサイド認証 (サーバ側が証明書を備え、クライアント側は備えない) は、TLS サーバ側として機能しながら、TLS クライアント側として動作する SIP エンドユーザデバイス (例えば、電話やパソコン) とその SIP サーバ(プロキシ、レジストラ)の間で使用される。TLS 相互認証 (クライアント側とサーバ側の両方がそれぞれの証明書を備えるところ) は、通常、SIP サーバ (プロキシ、レジストラ) 間や、PSTN ゲートウェイやメディアサーバのような静的に構成されたデバイスで使用される。相互認証モデルでは、2 つのエンティティが TLS 接続を確立できるように、両側が静的配置か再帰できることによって、お互いの証明書を有効なルート証明書として有効にすることが必要となる。サーバサイド認証では、クライアント側が証明書を備えないとき、クライアント側だけがサーバ側の証明書を有効にすることができる。

このすべての結果は、SIPS URI が TLS 接続の証明に使用するときはいつも、サーバ側からの証明書を有効にするために、接続 (クライアント) を確立するエンティティにとって可能であると予想される。サーバサイド認証に関して、[RFC5626] [5] は推奨のアプローチである。相互認証のために、ネットワークアーキテクチャは、接続がお互いの証明書にアクセスする手段を持っているエンティティの間で作られていることを保証する必要がある。以前に確立した TLS 接続を再利用できるのを確実にする際に、Record-Route [RFC3261] [3] と Path [RFC3327] [7] はとても役立つ。また、他のメカニズムはある特定の状況では使用されるかもしれない。例えば、広く認識されるルート証明書の使用は、より容易に TLS 接続の構築を許可する。

7. セキュリティ問題

このドキュメントの大部分が SIPS URI の使用法に適用されるため、セキュリティ問題であると考えることができる。[RFC3261] [3] の last-hop の例外は、SIP で重大な潜在的脆弱性を含んでいる。したがって、それはこの仕様では推奨しない。[RFC3261] [3] の 26.4.4 項は SIPS URI 体系のセキュリティ問題について説明する。また、Appendix A ([RFC5630] [2] の appendix 参照) によって変更されるように、これらのセキュリティ問題はここに適用される。

8. IANA 問題

この仕様は、2 つの新しい warning コードである 380 "SIPS Not Allowed" と 381 "SIPS Required".を示す。warning コードは以下の通り定義され、http://www.iana.org から利用可能な SIP パラメータレジストリのサブレジストリを warning コード (warn-codes) に含む。

380 SIPS Not Allowed：SIPS スキームが許されていないので、UAS やプロキシが要求を処理できない。
(例えば、現在、登録された SIPS コンタクトが全くないので)

381 SIPS Required : SIPS スキームが必要であるので、UAS かプロキシが要求を処理できない。

Reference: [RFC 5630] [2]

warning コードサブレジストリでの注意は以下の通り。

warning コードは SIP 応答メッセージで情報補足をステータスコードに提供する。