Internet Engineering Task Force (IETF)
Request for Comments: 8224
Obsoletes: 4474
Category: Standards Track
ISSN: 2070-1721

J. Peterson
NeuStar
C. Jennings
Cisco
E. Rescorla
RTFM, Inc.
C. Wendt
Comcast

February 2018

Authenticated Identity Management in the Session Initiation Protocol (SIP)

Abstract

The baseline security mechanisms in the Session Initiation Protocol (SIP) are inadequate for cryptographically assuring the identity of the end users that originate SIP requests, especially in an interdomain context. This document defines a mechanism for securely identifying originators of SIP requests. It does so by defining a SIP header field for conveying a signature used for validating the identity and for conveying a reference to the credentials of the signer.

This document obsoletes RFC 4474.

Status of This Memo

This is an Internet Standards Track document.

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

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc8224.

Table of Contents

1. Introduction

This document provides enhancements to the existing mechanisms for authenticated identity management in the Session Initiation Protocol (SIP) [RFC3261]. An identity, for the purposes of this document, is defined as either

- a canonical address-of-record (AoR) SIP URI employed to reach a user (such as "sip:alice@atlanta.example.com") or

- a telephone number, which commonly appears either in a tel URI [RFC3966] or as the user portion of a SIP URI.

[RFC3261] specifies several places within a SIP request where users can express an identity for themselves, most prominently the user-populated From header field. However, in the absence of some sort of cryptographic authentication mechanism, the recipient of a SIP request has no way to verify that the From header field has been populated appropriately. This leaves SIP vulnerable to a category of abuses such as impersonation attacks that facilitate or enable robocalling, voicemail hacking, swatting, and related problems as described in [RFC7340]. Ideally, a cryptographic approach to identity can provide a much stronger assurance of identity than the Caller ID services that the telephone network provides today, and one less vulnerable to spoofing.

[RFC3261] encourages user agents (UAs) to implement a number of potential authentication mechanisms, including Digest authentication, Transport Layer Security (TLS), and S/MIME (implementations may support other security schemes as well). However, few SIP UAs today support the end-user certificates necessary to authenticate themselves (via S/MIME, for example), and for its part Digest authentication is limited by the fact that the originator and destination must share a prearranged secret. Practically speaking, originating UAs need to be able to securely communicate their users' identities to destinations with which they have no previous association.

As an initial attempt to address this gap, [RFC4474] specified a means of signing portions of SIP requests in order to provide an identity assurance. However, [RFC4474] was in several ways misaligned with deployment realities (see [SIP-RFC4474-CONCERNS]). Most significantly, [RFC4474] did not deal well with telephone numbers as identifiers, despite their enduring use in SIP deployments. [RFC4474] also provided a signature over material that intermediaries in existing deployments commonly altered. This specification therefore deprecates the syntax and behavior specified by [RFC4474], reconsidering the problem space in light of the threat model in [RFC7375] and aligning the signature format with PASSporT (Personal Assertion Token) [RFC8225]. Backwards-compatibility considerations are given in Section 10.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119].

In addition, this document uses three terms specific to the mechanism:

- Identity: An identifier for the user of a communications service; for the purposes of SIP, either a SIP URI or a telephone number. Identities are derived from an "identity field" in a SIP request such as the From header field.

- Authentication Service: A logical role played by a SIP entity that adds Identity headers to SIP requests.

- Verification Service (or "Verifier"): A logical role played by a SIP entity that validates Identity headers in a SIP request.

3. Architectural Overview

The identity architecture for SIP defined in this specification depends on a logical "authentication service" that validates outgoing requests. An authentication service may be implemented either as part of a UA or as a proxy server; typically, it is a component of a network intermediary like a proxy to which originating UAs send unsigned requests. Once the originator of the message has been authenticated, through prearranged means with the authentication service, the authentication service then creates and adds an Identity header field to the request. This requires computing cryptographic information -- including a digital signature over some components of messages -- that lets other SIP entities verify that the sending user has been authenticated and its claim of a particular identity has been authorized. These "verification services" validate the signature and enable policy decisions to be made based on the results of the validation.

Policy decisions made after validation depend heavily on the verification service's trust for the credentials that the authentication service uses to sign requests. As robocalling, voicemail hacking, and swatting usually involve impersonation of telephone numbers, credentials that will be trusted by relying parties to sign for telephone numbers are a key component of the architecture. Authority over telephone numbers is, however, not as easy to establish on the Internet as authority over traditional domain names. This document assumes the existence of credentials for establishing authority over telephone numbers for cases where the telephone number is the identity of the user, but does not mandate or specify a credential system; [RFC8226] describes a credential system compatible with this architecture.

Although addressing the vulnerabilities in the Secure Telephone Identity Revisited (STIR) problem statement [RFC7340] and threat model mostly requires dealing with telephone number as identities, SIP must also handle signing for SIP URIs as identities. This is typically easier to deal with, as these identities are issued by organizations that have authority over Internet domains. When a new user becomes associated with example.com, for example, the administrator of the SIP service for that domain can issue them an identity in that namespace, such as sip:alice@example.com. Alice may then send REGISTER requests to example.com that make her UAs eligible to receive requests for sip:alice@example.com. In other cases, Alice may herself be the owner of her own domain and may issue herself identities as she chooses. But ultimately, it is the controller of the SIP service at example.com that must be responsible for authorizing the use of names in the example.com domain. Therefore, for the purposes of SIP as defined in [RFC3261], the necessary

credentials needed to prove that a user is authorized to use a particular From header field must ultimately derive from the domain owner: either (1) a UA gives requests to the domain name owner in order for them to be signed by the domain owner's credentials or (2) the UA must possess credentials that prove that the domain owner has given the UA the right to a name.

In order to share a cryptographic assurance of end-user SIP identity in an interdomain or intradomain context, an authentication service constructs tokens based on the PASSporT format [RFC8225], which is special encoding of a JSON [RFC8259] object comprising values derived from certain header field values in the SIP request. The authentication service computes a signature over those JSON elements as PASSporT specifies. An encoding of the resulting PASSporT is then placed in the SIP Identity header field. In order to assist in the validation of the Identity header field, this specification also describes a parameter of the Identity header field that can be used by the recipient of a request to recover the credentials of the signer.

Note that the scope of this document is limited to providing an identity assurance for SIP requests; solving this problem for SIP responses is outside the scope of this work (see [RFC4916]). Future work might specify ways that a SIP implementation could gateway PASSporTs to other protocols.

4. Identity Header Field Syntax

The Identity and Identity-Info header fields that were previously defined in [RFC4474] are deprecated by this document. This revised specification collapses the grammar of Identity-Info into the Identity header field via the "info" parameter. Note that unlike the prior specification in [RFC4474], the Identity header field is now allowed to appear more than one time in a SIP request. The revised grammar for the Identity header field builds on the ABNF [RFC5234] in [RFC3261], Section 25. It is as follows:

```
Identity = "Identity" HCOLON signed-identity-digest SEMI
    ident-info *( SEMI ident-info-params )
signed-identity-digest = 1*(base64-char / ".")
ident-info = "info" EQUAL ident-info-uri
ident-info-uri = LAQUOT absoluteURI RAQUOT
ident-info-params = ident-info-alg / ident-type /
    ident-info-extension
ident-info-alg = "alg" EQUAL token
ident-type = "ppt" EQUAL token
ident-info-extension = generic-param

base64-char = ALPHA / DIGIT / "/" / "+"
```

In addition to the "info" parameter, and the "alg" parameter previously defined in [RFC4474], this specification defines the optional "ppt" parameter (PASSporT Type). The "absoluteURI" portion of ident-info-uri MUST contain a URI; see Section 7.3 for more on choosing how to advertise credentials through this parameter.

The signed-identity-digest contains a base64 encoding of a PASSporT [RFC8225], which secures the request with a signature that PASSporT generates over the JSON header and payload objects; some of those header and claim element values will mirror values of the SIP request.

4.1. PASSporT Construction

For SIP implementations to populate the PASSporT header JSON object with fields from a SIP request, the following elements MUST be placed as the values corresponding to the designated JSON keys:

- First, per the baseline PASSporT specification [RFC8225], the JSON "typ" key MUST have the value "passport".

- Second, the JSON key "alg" MUST mirror the value of the optional "alg" parameter in the SIP Identity header field. Note that if the "alg" parameter is absent from the Identity header, the default value is "ES256".

- Third, the JSON key "x5u" MUST have a value equivalent to the quoted URI in the "info" parameter, per the simple string comparison rules of [RFC3986], Section 6.2.1.

- Fourth, if a PASSporT extension is in use, then the optional JSON key "ppt" MUST be present and have a value equivalent to the quoted value of the "ppt" parameter of the Identity header field.

An example of the PASSporT header JSON object without any extension is:

```
{ "typ":"passport",
  "alg":"ES256",
  "x5u":"https://www.example.com/cert.cer" }
```

To populate the PASSporT payload JSON object from a SIP request, the following elements MUST be placed as values corresponding to the designated JSON keys:

- First, the JSON "orig" object MUST be populated. If the originating identity is a telephone number, then the array MUST be populated with a JSON object containing a "tn" element with a value set to the value of the quoted originating identity, a canonicalized telephone number (see Section 8.3). Otherwise, the object MUST be populated with a JSON object containing a "uri" element, set to the value of the AoR of the UA sending the message as taken from the addr-spec of the From header field, per the procedures in Section 8.5.

- Second, the JSON "dest" array MUST be populated. If the destination identity is a telephone number, then the array MUST be populated with a JSON object containing a "tn" element with a value set to the value of the quoted destination identity, a canonicalized telephone number (see Section 8.3). Otherwise, the

array MUST be populated with a JSON object containing a "uri" element, set to the value of the addr-spec component of the To header field, which is the AoR to which the request is being sent, per the procedures in Section 8.5. Multiple JSON objects are permitted in "dest" for future compatibility reasons.

- Third, the JSON key "iat" MUST appear. The authentication service SHOULD set the value of "iat" to an encoding of the value of the SIP Date header field as a JSON NumericDate (as UNIX time, per [RFC7519], Section 2), though an authentication service MAY set the value of "iat" to its own current clock time. If the authentication service uses its own clock time, then the use of the full form of PASSporT is REQUIRED. In either case, the authentication service MUST NOT generate a PASSporT for a SIP request if the Date header is outside of its local policy for freshness (sixty seconds is RECOMMENDED).

- Fourth, if the request contains a Session Description Protocol (SDP) message body and if that SDP contains one or more "a=fingerprint" attributes, then the JSON key "mky" MUST appear with the algorithm(s) and value(s) of the fingerprint attributes (if they differ), following the format given in [RFC8225], Section 5.2.2.

For example:

```
{ "orig":{"tn":"12155551212"},
  "dest":{"tn":["12155551213"]},
  "iat":1443208345 }
```

For information on the security properties of these SIP message elements and why their inclusion mitigates replay attacks, see Section 12. Note that future extensions to PASSporT could introduce new claims and that further SIP procedures could be required to extract information from the SIP request to populate the values of those claims; see Section 9 of this document.

The "orig" and "dest" arrays may contain identifiers of heterogeneous type; for example, the "orig" array might contain a "tn" claim, while the "dest" contains a "uri" claim. Also note that in some cases, the "dest" array may be populated with more than one value. This could, for example, occur when multiple "dest" identities are specified in a meshed conference. Defining how a SIP implementation would align multiple destination identities in PASSporT with such systems is left as a subject for future specifications.

After these two JSON objects, the header and the payload, have been constructed and base64-encoded, they must each be hashed and signed per [RFC8225], Section 6. The header, payload, and signature components comprise a full PASSporT object. The resulting PASSporT may be carried in SIP in either (1) a full form, which includes the header and payload as well as the signature or (2) a compact form, which only carries the signature per [RFC8225], Section 7. The hashing and signing algorithm is specified by the "alg" parameter of the Identity header field and the mirrored "alg" parameter of PASSporT. All implementations of this specification MUST support the required signing algorithms of PASSporT. At present, there is one mandatory-to-support value for the "alg" parameter: "ES256", as defined in [RFC7519], which connotes an Elliptic Curve Digital Signature Algorithm (ECDSA) P-256 digital signature.

4.1.1. Example Full and Compact Forms of PASSporT in Identity

As Appendix F of the JSON Web Signature (JWS) specification [RFC7515] notes, there are cases where "it is useful to integrity-protect content that is not itself contained in a JWS." Since the fields that make up the majority of the PASSporT header and payload have values replicated in the SIP request, the SIP usage of PASSporT may exclude the base64-encoded version of the header and payload JSON objects from the Identity header field and instead present a detached signature: what PASSporT calls its compact form; see [RFC8225], Section 7.

When an authentication service constructs an Identity header, the contents of the signed-identity-digest field MUST contain either a full or compact PASSporT. Use of the compact form is RECOMMENDED in order to reduce message size, but note that extensions often require the full form (see Section 9).

For example, a full form of PASSporT in an Identity header might look as follows (backslashes shown for line folding only):

```
Identity: eyJhbGciOiJFUzI1NiIsInR5cCI6InBhc3Nwb3J0IiwieDV1I \
joiaHR0cHM6Ly9jZXJ0LmV4YW1wbGUub3JnL3Bhc3Nwb3J0LmNlciJ9.eyJ \
kZXN0Ijp7InVyaSI6WyJzaXA6YWxpY2VAZXhhbXBsZS5jb20iXX0sImlhdC \
I6IjE0NDMyMDgzNDUiLCJvcmlnIjp7InRuIjoiMTIxNTU1NTEyMTIifX0.r \
q3pjT1hoRwakEGjHCnWSwUnshd0-zJ6F1VOgFWSjHBr8Qjpjlk-cpFYpFYs \
ojNCpTzO3QfPOlckGaS6hEck7w;info=<https://biloxi.example.org \
/biloxi.cert>
```

The compact form of the same PASSporT object would appear in the Identity header as:

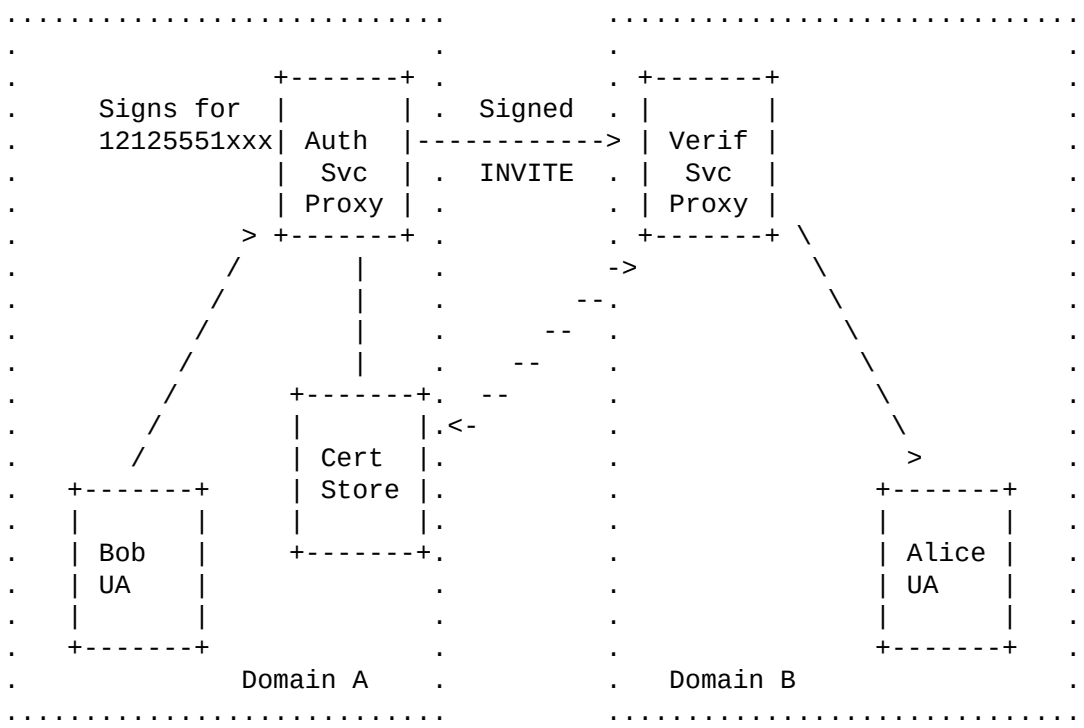
```
Identity: ..rq3pjT1hoRwakEGjHCnWSwUnshd0-zJ6F1VOgFWSjHBr8Qj \
pjlk-cpFYpFYsojNCpTzO3QfPOlckGaS6hEck7w;                   \
info=<https://biloxi.example.org/biloxi.cert>
```

5. Example of Operations

This section provides an informative (non-normative) high-level example of the operation of the mechanisms described in this document.

Imagine a case where Bob, who has the home proxy of example.com and the AoR sip:12155551212@example.com;user=phone, wants to communicate with Alice at sip:alice@example.com. They have no prior relationship, and Alice implements best practices to prevent impersonation attacks.

Bob's UA generates an INVITE and places his AoR in the From header field of the request. He then sends an INVITE to an authentication service proxy for his domain.



The proxy authenticates Bob and validates that he is authorized to assert the identity that he populated in the From header field. The proxy authentication service then constructs a PASSporT that contains a JSON representation of values that mirror certain parts of the SIP request, including the identity in the From header field value. As a part of generating the PASSporT, the authentication service signs a hash of that JSON header and payload with the private key associated with the appropriate credential for the identity (in this example, a certificate with authority to sign for numbers in a range from 12155551000 to 12155551999), and the signature is inserted by the proxy server into the Identity header field value of the request as a compact form of PASSporT. Alternatively, the JSON header and payload themselves might also have been included in the object when using the full form of PASSporT.

The proxy authentication service, as the holder of a private key with authority over Bob’s telephone number, is asserting that the originator of this request has been authenticated and that he is authorized to claim the identity that appears in the From header field. The proxy inserts an "info" parameter into the Identity header field that tells Alice how to acquire keying material necessary to validate its credentials (a public key), in case she doesn’t already have it.

When Alice’s domain receives the request, a proxy verification service validates the signature provided in the Identity header field and then determines that the authentication service credentials demonstrate authority over the identity in the From header field. This same validation operation might be performed by a verification service in Alice’s UA server (UAS). Ultimately, this valid request is rendered to Alice. If the validation were unsuccessful, some other treatment could be applied by the receiving domain or Alice’s UA.

5.1. Example Identity Header Construction

For the following SIP request:

```
INVITE sip:alice@example.com SIP/2.0
Via: SIP/2.0/TLS pc33.atlanta.example.com;branch=z9hG4bKnashds8
To: Alice <sip:alice@example.com>
From: Bob <sip:12155551212@example.com;user=phone>;tag=1928301774>
Call-ID: a84b4c76e66710
CSeq: 314159 INVITE
Max-Forwards: 70
Date: Fri, 25 Sep 2015 19:12:25 GMT
Contact: <sip:12155551212@gateway.example.com>
Content-Type: application/sdp
Content-Length: ...

v=0
o=UserA 2890844526 2890844526 IN IP4 pc33.atlanta.example.com
s=Session SDP
c=IN IP4 pc33.atlanta.example.com
t=0 0
m=audio 49172 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

An authentication service will create a corresponding PASSporT object. The properly serialized PASSporT header and payload JSON objects would look as follows. For the header, the values chosen by the authentication service at "example.com" might read:

```
{"alg":"ES256","typ":"passport","x5u":"https://cert.example.org/
   passport.cer"}
```

The serialized payload will derive values from the SIP request (the From, To, and Date header field values) as follows:

```
{"dest":{"uri":["sip:alice@example.com"]},"iat":1443208345,
  "orig":{"tn":"12155551212"}}
```

The authentication service would then generate the signature over the object, following the procedures in [RFC8225], Section 6. That signature would look as follows:

```
rq3pjT1hoRwakEGjHCnWSwUnshd0-zJ6F1VOgFWSjHBr8Qjpjlk-cpFYpFYs \
 ojNCpTzO3QfPOlckGaS6hEck7w
```

An authentication service signing this request and using the compact form of PASSporT would thus generate and add to the request an Identity header field of the following form:

```
Identity: ..rq3pjT1hoRwakEGjHCnWSwUnshd0-zJ6F1VOgFWSjHBr8Qjpj \
 lk-cpFYpFYsojNCpTzO3QfPOlckGaS6hEck7w; \
 info=<https://cert.example.org/passport.cer>
```

6. Signature Generation and Validation

SIP entities that instantiate the authentication service and verification service roles will, respectively, generate and validate the Identity header and the signature it contains.

6.1. Authentication Service Behavior

Any entity that instantiates the authentication service role MUST possess the private key of one or more credentials that can be used to sign for a domain or a telephone number (see Section 7.1). The authentication service role can be instantiated, for example, by an intermediary such as a proxy server or by a UA. Intermediaries that instantiate this role MUST be capable of authenticating one or more SIP users who can register for that identity. Commonly, this role will be instantiated by a proxy server, since proxy servers are more likely to have a static hostname, hold corresponding credentials, and have access to SIP registrar capabilities that allow them to authenticate users. It is also possible that the authentication service role might be instantiated by an entity that acts as a redirect server, but that is left as a topic for future work.

An authentication service adds the Identity header field to SIP requests. The procedures below define the steps that must be taken when each Identity header field is added. More than one Identity header field may appear in a single request, and an authentication service may add an Identity header field to a request that already contains one or more Identity header fields.

Entities instantiating the authentication service role perform the following steps, in order, to generate an Identity header field for a SIP request:

Step 1: Check Authority for the Identity

First, the authentication service must determine whether it is authoritative for the identity of the originator of the request. The authentication service extracts the identity from the URI value from the "identity field"; in ordinary operations, that is the addr-spec component of the From header field. In order to determine whether

the signature for the identity field should be over the entire identity field URI or just a telephone number, the authentication service MUST follow the process described in Section 8.1. The information in that section will lead to either the telephone number canonicalization procedures in Section 8.3 for telephone numbers or the URI normalization procedures described in Section 8.5 for domain names. Whichever the result, if the authentication service is not authoritative for the identity in question, it SHOULD process and forward the request normally unless the local policy is to block such requests. The authentication service MUST NOT add an Identity header field if the authentication service does not have the authority to make the claim it asserts.

Step 2: Authenticate the Originator

The authentication service MUST then determine whether or not the originator of the request is authorized to claim the identity given in the identity field. In order to do so, the authentication service MUST authenticate the originator of the message. Some possible ways in which this authentication might be performed include the following:

- If the authentication service is instantiated by a SIP intermediary (proxy server), it may authenticate the request with the authentication scheme used for registration in its domain (e.g., Digest authentication).

- If the authentication service is instantiated by a SIP UA, a UA may authenticate its own user through any system-specific means, perhaps simply by virtue of having physical access to the UA.

Authorization of the use of a particular username or telephone number in the user part of the From header field is a matter of local policy for the authentication service; see Section 7.1 for more information.

Note that this check is performed only on the addr-spec in the identity field (e.g., the URI of the originator, like "sip:alice@atlanta.example.com"); it does not cover the display-name portion of the From header field (e.g., "Alice Atlanta"). For more information, see Section 12.6.

Step 3: Verify Date is Present and Valid

An authentication service MUST add a Date header field to SIP requests that do not have one. The authentication service MUST ensure that any preexisting Date header field in the request is accurate. Local policy can dictate precisely how accurate the Date must be; a RECOMMENDED maximum discrepancy of sixty seconds will

ensure that the request is unlikely to upset any verifiers. If the Date header field value contains a time different by more than one minute from the current time noted by the authentication service, the authentication service SHOULD reject the request. Finally, the authentication service MUST verify that both the Date header field and the current time fall within the validity period of its credential.

See Section 12.1 for information on how the Date header field assists verifiers.

Step 4: Populate and Add the Identity Header

Subsequently, the authentication service MUST form a PASSporT object and add a corresponding Identity header field to the request containing either the full or compact form of PASSporT. For the baseline PASSporT header (headers containing no "ppt" parameter), this follows the procedures in Section 4; if the authentication service is using an alternative "ppt" format, it MUST add an appropriate "ppt" parameter and follow the procedures associated with that extension (see Section 9). After the Identity header field has been added to the request, the authentication service MUST also add an "info" parameter to the Identity header field. The "info" parameter contains a URI from which the authentication service's credential can be acquired; see Section 7.3 for more on credential acquisition.

An authentication service MAY use the full form of the PASSporT in the Identity header field. The presence of the full form is OPTIONAL because the information carried in the baseline PASSporT headers and claims is usually redundant with information already carried elsewhere in the SIP request. Using the compact form can significantly reduce SIP message size, especially when the PASSporT payload contains media keys. The syntax of the compact form is given in [RFC8225], Section 7; essentially, it contains only the signature component of the PASSporT.

Note that per the behavior specified in [RFC8225], use of the full form is mandatory when optional extensions are included. See Section 9.

6.1.1. Handling Repairable Errors

Also, in some cases, a request signed by an authentication service will be rejected by the verification service on the receiving side, and the authentication service will receive a SIP 4xx status code in the backwards direction, such as a 438 ("Invalid Identity Header") response indicating a verification failure. If the authentication

service did not originally send the full form of the PASSporT object in the Identity header field, it SHOULD retry the request with the full form after receiving a 438 response; however, implementations SHOULD NOT retry the request more than once. Authentication services implemented at proxy servers would retry such a request as a sequential fork, by reprocessing the destination as a new target and handling it serially as described in Section 16.6 of [RFC3261].

The information in the full form is useful on the verification side for debugging errors, and there are some known causes of verification failures (such as the Date header field value changing in transit; see Section 12.1 for more information) that can be resolved by the inclusion of the full form of PASSporT.

Finally, the authentication service forwards the message normally.

6.2. Verifier Behavior

This document specifies a logical role for SIP entities; this role is called a verification service, or verifier. When a verifier receives a SIP message containing one or more Identity header fields, it inspects the signature(s) to verify the identity of the originator of the message. The results of a verification are provided as input to an authorization process that is outside the scope of this document.

A SIP request may contain zero, one, or more Identity header fields. A verification service performs the steps below on each Identity header field that appears in a request. If a verification service cannot use any Identity header in a request, due to the absence of Identity headers or unsupported "ppt" parameters, and the presence of an Identity header field is required by local policy (for example, based on a per-sending-domain policy or a per-sending-user policy), then a 428 "Use Identity Header" response MUST be sent in the backwards direction. For more on this and other verifier responses, see Section 6.2.2.

In order to verify an Identity header field in a message, an entity acting as a verifier MUST perform the following steps, in the order specified below. Note that when an Identity header field contains a full-form PASSporT object, the verifier MUST follow the additional procedures in Section 6.2.4.

Step 1: Check for an Unsupported "ppt"

The verifier MUST inspect any optional "ppt" parameter appearing in the Identity header. If no "ppt" parameter is present, then the verifier proceeds normally with Steps 2 through 5. If a "ppt" parameter value is present and the verifier does not support it,

it MUST ignore the Identity header field. If a supported "ppt" parameter value is present, the verifier proceeds with Step 2 and will ultimately follow the "ppt" variations described in Step 5.

Step 2: Determine the Originator's Identity

In order to determine whether the signature for the identity field should be over the entire identity field URI or just a telephone number, the verification service MUST follow the process described in Section 8.1. The information in that section will lead to either the telephone number canonicalization procedures in Section 8.3 for telephone numbers or the URI normalization procedures described in Section 8.5 for domain names.

Step 3: Identify Credential for Validation

The verifier must ensure that it has access to the proper keying material to validate the signature in the Identity header field; this usually involves dereferencing a URI in the "info" parameter of the Identity header field. See Section 7.2 for more information on these procedures. If the verifier does not support the credential described in the "info" parameter, then it treats the credential for this header field as unsupported.

Step 4: Check the Freshness of Date

The verifier furthermore ensures that the value of the Date header field of the request meets local policy for freshness (sixty seconds is RECOMMENDED) and that it falls within the validity period of the credential used to sign the Identity header field. For more on the attacks this prevents, see Section 12.1. If the full form of the PASSporT is present, the verifier SHOULD compare the "iat" value in the PASSporT to the Date header field value in the request. If the two are different, and the "iat" value differs from the Date header field value but remains within verification service policy for freshness, the verification service SHOULD perform the computation required by Step 5, using the "iat" value instead of the Date header field value.

Step 5: Validate the Signature

The verifier MUST validate the signature in the Identity header field over the PASSporT object. For baseline PASSporT objects (with no Identity header field "ppt" parameter), the verifier MUST follow the procedures for generating the signature over a PASSporT object as described in Section 4. If a "ppt" parameter is present (and, per Step 1, is supported), the verifier follows the procedures for that "ppt" (see Section 9). If a verifier determines that the signature

in the Identity header field does not correspond to the reconstructed signed-identity-digest, then the Identity header field should be considered invalid.

6.2.1. Authorization of Requests

The verification of an Identity header field does not entail any particular treatment of the request. The handling of the message after the verification process depends on how the verification service is implemented and on local policy. This specification does not propose any authorization policy for UAs or proxy servers to follow based on the presence of a valid Identity header field, the presence of an invalid Identity header field, the absence of an Identity header field, or the presence of a stale Date header field value. However, it is anticipated that local policies could involve making different forwarding decisions in intermediary implementations, or changing how the user is alerted or how identity is rendered in UA implementations.

The presence of multiple Identity header fields within a message raises the prospect that a verification service could receive a message containing both valid and invalid Identity header fields. As a guideline, this specification recommends that only if a verifier determines that all Identity header fields within a message are invalid should the request be considered to have an invalid identity. If at least one Identity header field value is valid and from a trusted source, then relying parties can use that header for authorization decisions regardless of whether other untrusted or invalid Identity headers appear in a request.

6.2.2. Failure Response Codes Sent by a Verification Service

[RFC4474] originally defined four response codes for failure conditions specific to the Identity header field and its original mechanism. These status codes are retained in this specification, with some slight modifications. Also, this specification details responding with a 403 "Forbidden" response when a stale Date header field value is received; see below.

A 428 response will be sent (per Section 6.2) when an Identity header field is required but no Identity header field without a "ppt" parameter or with a supported "ppt" value has been received. In the case where one or more Identity header fields with unsupported "ppt" values have been received, then a verification service may send a 428 with a human-readable reason phrase like "Use Supported PASSporT Format". Note, however, that this specification gives no guidance on

how a verification service might decide to require an Identity header field for a particular SIP request. Such authorization policies are outside the scope of this specification.

The 436 "Bad Identity Info" response code indicates an inability to acquire the credentials needed by the verification service for validating the signature in an Identity header field. Again, given the potential presence of multiple Identity header fields, this response code should only be sent when the verification service is unable to dereference the URIs and/or acquire the credentials associated with all Identity header fields in the request. This failure code could be repairable if the authentication service resends the request with an "info" parameter pointing to a credential that the verification service can access.

The 437 "Unsupported Credential" response (previously "Unsupported Certificate"; see Section 13.2) is sent when a verification service can acquire, or already holds, the credential represented by the "info" parameter of at least one Identity header field in the request but does not support said credential(s), for reasons such as failing to trust the issuing certification authority (CA) or failing to support the algorithm with which the credential was signed.

The 438 "Invalid Identity Header" response indicates that of the set of Identity header fields in a request, no header field with a valid and supported PASSporT object has been received. Like the 428 response, this is sent by a verification service when its local policy dictates that a broken signature in an Identity header field is grounds for rejecting a request. Note that in some cases, an Identity header field may be broken for other reasons than that an originator is attempting to spoof an identity: for example, when a transit network alters the Date header field of the request. Sending a full-form PASSporT can repair some of these conditions (see Section 6.2.4), so the recommended way to attempt to repair this failure is to retry the request with the full form of PASSporT if it had originally been sent with the compact form. The alternative reason phrase "Invalid PASSporT" can be used when an extended full-form PASSporT lacks required headers or claims, or when an extended full-form PASSporT signaled with the "ppt" parameter lacks required claims for that extension. Sending a string along these lines will help humans debugging the sending system.

Finally, a 403 response may be sent when the verification service receives a request with a Date header field value that is older than the local policy for freshness permits. The same response may be used when the "iat" in the full form of a PASSporT has a value older than the local policy for freshness permits. The reason phrase "Stale Date" can be sent to help humans debug the failure.

Future specifications may explore ways, including Reason codes or Warning headers, to communicate further information that could be used to disambiguate the source of errors in cases with multiple Identity headers in a single request or to provide similar detailed feedback for debugging purposes.

6.2.3. Handling Retried Requests

If a verification service sends a failure response in the backwards direction, the authentication service may retry the request as described in Section 6.1.1. If the authentication service is instantiated at a proxy server, then it will retry the request as a sequential fork. Verification services implemented at a proxy server will recognize this request as a spiral rather than a loop due to the proxy behavior fix documented in [RFC5393], Section 4.2. However, if the verification service is implemented in an endpoint, the endpoint will need to override the default UAS behavior (in particular, the SHOULD in [RFC3261], Section 8.2.2.2) to accept this request as a spiral rather than a loop.

6.2.4. Handling the Full Form of PASSporT

If the full form of PASSporT is present in an Identity header, this permits the use of optional extensions as described in [RFC8225], Section 8.3. Furthermore, the verification service can extract from the "orig" and "dest" elements of the PASSporT full form the canonical telephone numbers created by the authentication service, as well as an "iat" claim corresponding to the Date header field that the authentication service used. These values may be used to debug canonicalization problems or to avoid unnecessary signature breakage caused by intermediaries that alter certain SIP header field values in transit.

However, the verification service MUST NOT treat the value in the "orig" of a full-form PASSporT as the originating identity of the call: the originating identity of the call is always derived from the SIP signaling, and it is that value, per the procedures above in Section 6.2 Step 2, that is used to recompute the signature at the verification service. That value, rather than the value inside the PASSporT object, is rendered to an end user in ordinary SIP operations, and if a verification service were to simply trust that

the value in the "orig" corresponded to the call that it received without comparing it to the call signaling, this would enable various cut-and-paste attacks. As an optimization, when the full form is present, the verification service MAY delay performing that cryptographic operation and first compute its own canonicalization of an originating telephone number to compare it to the values in the "orig" element of PASSporT. This would allow the verification service to ascertain whether or not the two ends agree on the canonical number form; if they do not, then surely the signature validation would fail.

7. Credentials

This section gives general guidance on the use of credential systems by authentication and verification services, as well as requirements that must be met by credential systems that conform with this architecture. It does not mandate any specific credential system.

Furthermore, this specification allows either a UA or a proxy server to provide the authentication service function and/or the verification service function. For the purposes of end-to-end security, it is obviously preferable for end systems to acquire their own credentials; in this case, UAs can act as authentication services. However, for some deployments, end-user credentials may be neither practical nor affordable, given the potentially large number of SIP UAs (phones, PCs, laptops, PDAs, gaming devices) that may be employed by a single user. Synchronizing keying material across multiple devices may be prohibitively complex and require quite a good deal of additional endpoint behavior. Managing several credentials for the various devices could also be burdensome. Thus, for reasons of credential management alone, implementing the authentication service at an intermediary may be more practical. This trade-off needs to be understood by implementers of this specification.

7.1. Credential Use by the Authentication Service

In order to act as an authentication service, a SIP entity must possess the private keying material of one or more credentials that cover domain names or telephone numbers. These credentials may represent authority over one domain (such as example.com) or a set of domains enumerated by the credential. Similarly, a credential may represent authority over a single telephone number or a range of telephone numbers. The way that the scope of a credential’s authority is expressed is specific to the credential mechanism.

Authorization of the use of a particular username or telephone number in the From header field value is a matter of local policy for the authentication service, one that depends greatly on the manner in which authentication is performed. For non-telephone number user parts, one policy might be as follows: the username given in the "username" parameter of the Proxy-Authorization header field must correspond exactly to the username in the From header field of the SIP message. However, there are many cases in which this is too limiting or inappropriate; a realm might use "username" parameters in the Proxy-Authorization header field that do not correspond to the user portion of From header fields, or a user might manage multiple accounts in the same administrative domain. In this latter case, a domain might maintain a mapping between the values in the "username" parameter of the Proxy-Authorization header field and a set of one or more SIP URIs that might legitimately be asserted for that "username". For example, the username can correspond to the "private identity" as defined by the Third Generation Partnership Project (3GPP) [TS-3GPP.23.228], in which case the From header field can contain any one of the public identities associated with this private identity. In this instance, another policy might be as follows: the URI in the From header field must correspond exactly to one of the mapped URIs associated with the "username" given in the Proxy-Authorization header field. This is a suitable approach for telephone numbers in particular.

This specification could also be used with credentials that cover a single name or URI, such as alice@example.com or sip:alice@example.com. This would require a modification to authentication service behavior to operate on a whole URI rather than a domain name. Because this is not believed to be a pressing use case, this is deferred to future work, but implementers should note this as a possible future direction.

Exceptions to such authentication service policies arise for cases like anonymity; if the AoR asserted in the From header field uses a form like "sip:anonymous@example.com" (see [RFC3323]), then the "example.com" proxy might authenticate only that the user is a valid user in the domain and insert the signature over the From header field as usual.

7.2. Credential Use by the Verification Service

In order to act as a verification service, a SIP entity must have a way to acquire credentials for authorities over particular domain names, telephone numbers, and/or number ranges. Dereferencing the URI found in the "info" parameter of the Identity header field (as described in Section 7.3) MUST be supported by all verification service implementations to create a baseline means of credential

acquisition. Provided that the credential used to sign a message is not previously known to the verifier, SIP entities SHOULD discover this credential by dereferencing the "info" parameter, unless they have some implementation-specific way of acquiring the needed keying material, such as an offline store of periodically updated credentials. The 436 "Bad Identity Info" response exists for cases where the verification service cannot dereference the URI in the "info" parameter.

This specification does not propose any particular policy for a verification service to determine whether or not the holder of a credential is the appropriate party to sign for a given SIP identity. Guidance on this is deferred to credential mechanism specifications.

Verification service implementations supporting this specification may wish to have some means of retaining credentials (in accordance with normal practices for credential lifetimes and revocation) in order to prevent themselves from needlessly downloading the same credential every time a request from the same identity is received. Credentials cached in this manner may be indexed in accordance with local policy: for example, by their scope of authority or by the URI given in the "info" parameter value. Further consideration of how to cache credentials is deferred to the credential mechanism specifications.

7.3. "info" Parameter URIs

An "info" parameter MUST contain a URI that dereferences to a resource that contains the public key components of the credential used by the authentication service to sign a request. It is essential that a URI in the "info" parameter be dereferencable by any entity that could plausibly receive the request. For common cases, this means that the URI SHOULD be dereferencable by any entity on the public Internet. In constrained deployment environments, a service private to the environment MAY be used instead.

Beyond providing a means of accessing credentials for an identity, the "info" parameter further serves as a means of differentiating which particular credential was used to sign a request, when there are potentially multiple authorities eligible to sign. For example, imagine a case where a domain implements the authentication service role for a range of telephone numbers and a UA belonging to Alice has acquired a credential for a single telephone number within that range. Either would be eligible to sign a SIP request for the number in question. Verification services, however, need a means to differentiate which one performed the signature. The "info" parameter performs that function.

7.4. Credential System Requirements

This document makes no recommendation for the use of any specific credential system. Today, there are two primary credential systems in place for proving ownership of domain names: certificates (e.g., X.509 v3; see [RFC5280]) and the domain name system itself (e.g., DNS-Based Authentication of Named Entities (DANE); see [RFC6698]). It is envisioned that either could be used in the SIP identity context: an "info" parameter could, for example, give an HTTP URL of the Content-Type "application/pkix-cert" pointing to a certificate (following the conventions of [RFC2585]). The "info" parameter might use the DNS URL scheme (see [RFC4501]) to designate keys in the DNS.

While no comparable public credentials exist for telephone numbers, either approach could be applied to telephone numbers. A credential system based on certificates is given in [RFC8226], but this specification can work with other credential systems; for example, using the DNS was proposed in [CIDER].

In order for a credential system to work with this mechanism, its specification must detail:

- which URI schemes the credential will use in the "info" parameter, and any special procedures required to dereference the URIs,
- how the verifier can learn the scope of the credential,
- any special procedures required to extract keying material from the resources designated by the URI,
- any algorithms required to validate the credentials (e.g., for certificates, any algorithms used by certificate authorities to sign certificates themselves), and
- how the associated credentials will support the mandatory signing algorithm(s) required by PASSporT [RFC8225].

SIP entities cannot reliably predict where SIP requests will terminate. When choosing a credential scheme for deployments of this specification, it is therefore essential that the trust anchor(s) for credentials be widely trusted or that deployments restrict the use of this mechanism to environments where the reliance on particular trust anchors is assured by business arrangements or similar constraints.

Note that credential systems must address key lifecycle management concerns: were a domain to change the credential available at the Identity header field "info" parameter URI before a verifier evaluates a request signed by an authentication service, this would

cause obvious verifier failures. When a rollover occurs, authentication services SHOULD thus provide new "info" URIs for each new credential and SHOULD continue to make older key acquisition URIs available for a duration longer than the plausible lifetime of a SIP transaction (a minute would most likely suffice).

8. Identity Types

The STIR problem statement [RFC7340] focuses primarily on cases where the called and calling parties identified in the To and From header field values use telephone numbers, as this remains the dominant use case in the deployment of SIP. However, the Identity header mechanism also works with SIP URIs without telephone numbers (of the form "sip:user@host") and, potentially, other identifiers when SIP interworks with other protocols.

Authentication services confirm the identity of the originator of a call, which is typically found in the From header field value. The guidance in this specification also applies to extracting the URI containing the originator's identity from the P-Asserted-Identity header field value instead of the From header field value. In some trusted environments, the P-Asserted-Identity header field is used in lieu of the From header field to convey the AoR or telephone number of the originator of a request; where it does, local policy might therefore dictate that the canonical identity derives from the P-Asserted-Identity header field rather than the From header field.

Ultimately, in any case where local policy canonicalizes the identity into a form different from how it appears in the From header field, the use of the full form of PASSporT by authentication services is RECOMMENDED, but because the "orig" claim of PASSporT itself could then divulge information about users or networks, implementers should be mindful of the guidelines in Section 11.

8.1. Differentiating Telephone Numbers from URIs

In order to determine whether or not the user portion of a SIP URI is a telephone number, authentication services and verification services MUST perform the following procedure on any SIP URI they inspect that contains a numeric user part. Note that the same procedures are followed for creating the canonical form of a URI found in the From header field as the procedures used for a URI found in the To header field or the P-Asserted-Identity header field.

First, implementations will ascertain if the user portion of the URI constitutes a telephone number. Telephone numbers most commonly appear in SIP header field values in the username portion of a SIP URI (e.g., "sip:+17005551008@chicago.example.com;user=phone"). The

user part of SIP URIs with the "user=phone" parameter conforms to the syntax of the tel URI scheme [RFC3966]. It is also possible for a tel URI to appear in SIP header fields outside the context of a SIP or Session Initiation Protocol Secure (SIPS) URI (e.g., "tel:+17005551008"). Thus, in standards-compliant environments, numbers will be explicitly labeled by the use of tel URIs or the "user=phone" parameter.

Alternatively, implementations in environments that do not conform to those standards MAY follow local policies for identifying telephone numbers. For example, implementations could infer that the user part is a telephone number due to the presence of the "+" indicator at the start of the user portion. Absent even that indication, if there are numbers present in the user portion, implementations might conceivably also detect that the user portion of the URI contains a telephone number by determining whether or not those numbers would be dialable or routable in the local environment -- bearing in mind that the telephone number may be a valid E.164 number [E.164], a nationally specific number, or even a private branch exchange number. Implementations could also rely on external hints: for example, a verification service implementation could infer from the type of credential that signed a request that the signature must be over a telephone number.

Regardless of how the implementation detects telephone numbers, once a telephone number has been detected, implementations SHOULD follow the procedures in Section 8.3. If the URI field does not contain a telephone number or if the result of the canonicalization of the From header field value does not form a valid E.164 telephone number, the authentication service and/or verification service SHOULD treat the entire URI as a SIP URI and apply the procedures in Section 8.5. These URI normalization procedures are invoked to canonicalize the URI before it is included in a PASSporT object in, for example, a "uri" claim. See Section 8.5 for that behavior.

8.2. Authority for Telephone Numbers

In order for telephone numbers to be used with the mechanism described in this document, authentication services must receive credentials from an authority for telephone numbers or telephone number ranges, and verification services must trust the authority employed by the authentication service that signs a request. Per Section 7.4, enrollment procedures and credential management are outside the scope of this document; approaches to credential management for telephone numbers are discussed in [RFC8226].

8.3. Telephone Number Canonicalization Procedures

Once an implementation has identified a telephone number, it must construct a number string. That requires performing the following steps:

- Implementations MUST drop any "+"s, internal dashes, parentheses, or other non-numeric characters, except for the "#" or "*" keys used in some special service numbers (typically, these will appear only in the To header field value). This MUST result in an ASCII string limited to "#", "*", and digits without whitespace or visual separators.

- Next, an implementation must assess if the number string is a valid, globally routable number with a leading country code.

  If not, implementations SHOULD convert the number into E.164 format, adding a country code if necessary; this may involve transforming the number from a dial string (see [RFC3966]), removing any national or international dialing prefixes or performing similar procedures. It is only in the case that an implementation cannot determine how to convert the number to a globally routable format that this step may be skipped. This will be the case, for example, for nationally specific service numbers (e.g., 911, 112); however, calls to those numbers are routed in a very strict fashion, which ordinarily prevents them from reaching entities that don't understand the numbers.

- Some domains may need to take unique steps to convert their numbers into a global format, and such transformations during canonicalization can also be made in accordance with specific policies used within a local domain. For example, one domain may only use local number formatting and need to convert all To/From header field user portions to E.164 by prepending country-code and region-code digits; another domain might have prefixed usernames with trunk-routing codes, in which case the canonicalization will need to remove the prefix. This specification cannot anticipate all of the potential transformations that might be useful.

- The resulting canonical number string will be used as input to the hash calculation during signing and verifying processes.

The ABNF of this number string is:

```
tn-spec =  1*tn-char
tn-char = "#" / "*" / DIGIT
```

The resulting number string is used in the construction of the telephone number field(s) in a PASSporT object.

8.4. Authority for Domain Names

To use a SIP URI as an identity in this mechanism requires authentication and verification systems to support standard mechanisms for proving authority over a domain name: that is, the domain name in the host portion of the SIP URI.

A verifier MUST evaluate the correspondence between the user's identity and the signing credential by following the procedures defined in [RFC5922], Section 7.2. While [RFC5922] deals with the use of TLS and is specific to certificates, the procedures described are applicable to verifying identity if one substitutes the "hostname of the server" for the domain portion of the user's identity in the From header field of a SIP request with an Identity header field.

This process is complicated by two deployment realities. In the first place, credentials have varying ways of describing their subjects and may indeed have multiple subjects, especially in "virtual hosting" cases where multiple domains are managed by a single application (see [RFC5922], Section 7.8). Secondly, some SIP services may delegate SIP functions to a subordinate domain and utilize the procedures in [RFC3263] that allow requests for, say, "example.com" to be routed to "sip.example.com". As a result, a user with the AoR "sip:alice@example.com" may process requests through a host like "sip.example.com", and it may be that latter host that acts as an authentication service.

To address the second of these problems, a domain that deploys an authentication service on a subordinate host might supply that host with the private keying material associated with a credential whose subject is a domain name that corresponds to the domain portion of the AoRs that the domain distributes to users. Note that this corresponds to the comparable case of routing inbound SIP requests to a domain. When the NAPTR and SRV procedures of [RFC3263] are used to direct requests to a domain name other than the domain in the original Request-URI (e.g., for "sip:alice@example.com", the corresponding SRV records point to the service "sip1.example.org"), the client expects that the certificate passed back in any TLS exchange with that host will correspond exactly with the domain of the original Request-URI, not the domain name of the host.

Consequently, in order to make inbound routing to such SIP services work, a domain administrator must similarly be willing to share the domain's private key with the service. This design decision was made to compensate for the insecurity of the DNS, and it makes certain potential approaches to DNS-based "virtual hosting" unsecurable for SIP in environments where domain administrators are unwilling to share keys with hosting services.

8.5. URI Normalization

Just as telephone numbers may undergo a number of syntactic transformations during transit, the same can happen to SIP and SIPS URIs without telephone numbers as they traverse certain intermediaries. Therefore, when generating a PASSporT object based on a SIP request, any SIP and SIPS URIs must be transformed into a canonical form that captures the AoR represented by the URI before they are provisioned in PASSporT claims such as "uri". The URI normalization procedures required are as follows.

Following the ABNF of [RFC3261], the SIP or SIPS URI in question MUST discard all elements after the "hostport" of the URI, including all uri-parameters and escaped headers, from its syntax. Of the userinfo component of the SIP URI, only the user element will be retained: any password (and any leading ":" before the password) MUST be removed, and since this userinfo necessarily does not contain a telephone-subscriber component, no further parameters can appear in the user portion.

The hostport portion of the SIP or SIPS URI MUST similarly be stripped of any trailing port along with the ":" that proceeds the port, leaving only the host.

The ABNF of this canonical URI form (following the syntax defined in [RFC3261]) is:

```
canon-uri =  ( "sip" / "sips" ) ":" user "@" host
```

Finally, the URI will be subject to the syntax-based URI normalization procedures of [RFC3986], Section 6.2.2. Implementations MUST perform case normalization (rendering the scheme, user, and host all lowercase) and percent-encoding normalization (decoding any percent-encoded octet that corresponds to an unreserved character, per [RFC3986], Section 2.3). However, note that normalization procedures face known challenges in some internationalized environments (see [IRI-COMPARISON]) and that perfect normalization of URIs may not be possible in those environments.

For future PASSporT applications, it may be desirable to provide an identifier without an attached protocol scheme. Future specifications that define PASSporT claims for SIP as a using protocol could use these basic procedures but could eliminate the scheme component. A more exact definition is left to future specifications.

9. Extensibility

As future requirements may warrant increasing the scope of the Identity mechanism, this specification specifies an optional "ppt" parameter of the Identity header field, which mirrors the "ppt" header in PASSporT. The "ppt" parameter value MUST consist of a token containing an extension specification, which denotes an extended set of one or more signed claims per the type extensibility mechanism specified in [RFC8225], Section 8. Note that per the guidance in that section, "ppt" is used only to enforce a mandatory extension: optional claims may be added to any PASSporT object without requiring the use of "ppt", but the compact form of PASSporT MUST NOT be used when optional claims are present in the PASSporT payload.

The potential for extensions is one of the primary motivations for allowing the presence of multiple Identity header fields in the same SIP request. It is envisioned that future extensions might allow for alternate information to be signed or explicitly allow different parties to provide the signatures than the authorities envisioned by baseline STIR. A request might, for example, have one Identity added by an authentication service at the originating administrative domain and then another Identity header field added by some further intermediary using a PASSporT extension. While this specification does not define any such specific purpose for multiple Identity header fields, implementations MUST support receiving multiple header fields for reasons of future compatibility.

An authentication service cannot assume that verifiers will understand any given extension. Verifiers that do support an extension may then trigger appropriate application-level behavior in the presence of an extension; authors of extensions should provide appropriate extension-specific guidance to application developers on this point.

10. Backwards Compatibility with RFC 4474

This specification introduces several significant changes from the version of the Identity header field defined by [RFC4474]. However, due to the problems enumerated in [SIP-RFC4474-CONCERNS], it is not believed that the original Identity header field has seen any deployment, or even implementation in deployed products.

As such, this mechanism contains no provisions for signatures generated with this specification to work with implementations compliant with [RFC4474], nor does it contain any related backwards-compatibility provisions. Hypothetically, were an implementation compliant with [RFC4474] to receive messages containing this revised version of the Identity header field, it would likely fail the request with a 436 response code due to the absence of an Identity-Info header field (Section 4). Implementations of this specification, for debugging purposes, might interpret a 436 with a reason phrase of "Bad Identity Info" (previously "Bad Identity-Info"; see Section 13.2) as an indication that the request has failed because it reached a (hypothetical) verification service that is compliant with [RFC4474].

11. Privacy Considerations

The purpose of this mechanism is to provide a reliable identification of the originator of a SIP request, specifically a cryptographic assurance that an authority asserts the originator can claim the URI the identity stipulated in the request. This URI may contain or imply a variety of personally identifying information, including the name of a human being, their place of work or service provider, and, possibly, further details. The intrinsic privacy risks associated with that URI are, however, no different from those of baseline SIP. Per the guidance in [RFC6973], implementers should make users aware of the privacy trade-off of providing secure identity.

The identity mechanism presented in this document is compatible with the standard SIP practices for privacy described in [RFC3323]. A SIP proxy server can act as both a privacy service as described in [RFC3323] and an authentication service. Since a UA can provide any From header field value that the authentication service is willing to authorize, there is no reason why private SIP URIs that contain legitimate domains (e.g., sip:anonymous@example.com) cannot be signed by an authentication service. The construction of the Identity header field is the same for private URIs as it is for any other sort of URIs. Similar practices could be used to support opportunistic signing of SIP requests for UA-integrated authentication services with self-signed certificates, though that is outside the scope of this specification and is left as a matter for future investigation.

Note, however, that even when using anonymous SIP URIs, an authentication service must possess a certificate corresponding to the host portion of the addr-spec of the From header field value of the request; accordingly, using domains like "anonymous.invalid" will not be usable by privacy services that simultaneously act as authentication services. The assurance offered by the usage of anonymous URIs with a valid domain portion is "this is a known user in my domain that I have authenticated, but I am keeping its identity private."

It is worth noting two features of this more anonymous form of identity. One can eliminate any identifying information in a domain through the use of the domain "anonymous.invalid", but we must then acknowledge that it is difficult for a domain to be both anonymous and authenticated. The use of the domain "anonymous.invalid" entails that no corresponding authority for the domain can exist, and as a consequence, authentication service functions for that domain are meaningless. The second feature is more germane to the threats this document mitigates [RFC7375]. None of the relevant attacks, all of which rely on the attacker taking on the identity of a victim or hiding their identity using someone else's identity, are enabled by an anonymous identity. As such, the inability to assert an authority over an anonymous domain is irrelevant to our threat model.

[RFC3325] defines the "id" priv-value token, which is specific to the P-Asserted-Identity header field. The sort of assertion provided by the P-Asserted-Identity header field is very different from the Identity header field presented in this document. It contains additional information about the originator of a message that may go beyond what appears in the From header field; P-Asserted-Identity holds a definitive identity for the originator that is somehow known to a closed network of intermediaries. Presumably, that network will use this identity for billing or security purposes. The danger of this network-specific information leaking outside of the closed network motivated the "id" priv-value token. The "id" priv-value token has no implications for the Identity header field, and privacy services MUST NOT remove the Identity header field when a priv-value of "id" appears in a Privacy header field.

The full form of the PASSporT object provides the complete JSON objects used to generate the signed-identity-digest of the Identity header field value, including the canonicalized form of the telephone number of the originator of a call if the signature is over a telephone number. In some contexts, local policy may require a canonicalization that differs substantially from the original From header field. Depending on those policies, potentially the full form of PASSporT might divulge information about the originating network or user that might not appear elsewhere in the SIP request. Were it

to be used to reflect the contents of the P-Asserted-Identity header field, for example, then the object would need to be converted to the compact form when the P-Asserted-Identity header is removed to avoid any such leakage outside of a trust domain. Since, in those contexts, the canonical form of the originator's identity could not be reassembled by a verifier and thus the Identity signature validation process would fail, using P-Asserted-Identity with the full form of PASSporT in this fashion is NOT RECOMMENDED outside of environments where SIP requests will never leave the trust domain. As a side note, history shows that closed networks never stay closed and one should design their implementation assuming connectivity to the broader Internet.

Finally, note that unlike [RFC3325], the mechanism described in this specification adds no information to SIP requests that has privacy implications -- apart from disclosing that an authentication service is willing to sign for an originator.

12. Security Considerations

This document describes a mechanism that provides a signature over the Date header field of SIP requests, parts of the To and From header fields, and (when present) any media keying material in the message body. In general, the considerations related to the security of these header fields are the same as those given in [RFC3261] for including header fields in tunneled "message/sip" MIME bodies (see Section 23 of [RFC3261] in particular). This section details the individual security properties obtained by including each of these header fields within the signature; collectively, this set of header fields provides the necessary properties to prevent impersonation. It addresses the solution-specific attacks against in-band solutions enumerated in [RFC7375], Section 4.1.

12.1. Protected Request Fields

The From header field value (in ordinary operations) indicates the identity of the originator of the message; for the purposes of this document, either the SIP AoR URI or an embedded telephone number provides the identity of a SIP user. Note that in some deployments the identity of the originator may reside in P-Asserted-Identity instead. The originator's identity is the key piece of information that this mechanism secures; the remainder of the signed parts of a SIP request are present to provide reference integrity and to prevent certain types of cut-and-paste attacks.

The Date header field value protects against cut-and-paste attacks, as described in [RFC3261], Section 23.4.2. That specification recommends that implementations notify the user of a potential

security issue if the signed Date header field value is stale by an hour or more. To prevent cut-and-paste of recently observed messages, this specification instead RECOMMENDS a shorter interval of sixty seconds. Implementations of this specification MUST NOT deem valid a request with an outdated Date header field. Note that per the behavior described in [RFC3893], Section 10, servers can keep state of recently received requests, and thus if an Identity header field is replayed by an attacker within the Date interval, verifiers can detect that it is spoofed because a message with an identical Date from the same source had recently been received.

It has been observed in the wild that some networks change the Date header field value of SIP requests in transit; to accommodate that type of scenario, alternative behavior might be necessary. Verification services that observe a signature validation failure MAY therefore reconstruct the Date header field component of the signature from the "iat" carried in the full form of PASSporT: provided that time recorded by "iat" falls within the local policy for freshness that would ordinarily apply to the Date header, the verification service MAY treat the signature as valid, provided it keeps adequate state to detect recent replays. Note that this will require the inclusion of the full form of the PASSporT object by authentication services in networks where such failures are observed.

The To header field value provides the identity of the SIP user that this request originally targeted. Covering the identity in the To header field with the Identity signature serves two purposes. First, it prevents cut-and-paste attacks in which an Identity header field from a legitimate request for one user is cut-and-pasted into a request for a different user. Second, it preserves the starting URI scheme of the request; this helps prevent downgrade attacks against the use of SIPS. The To identity offers additional protection against cut-and-paste attacks beyond the Date header field. For example, without a signature over the To identity, an attacker who receives a call from a target could immediately cut-and-paste the Identity and From header field value from that INVITE into a new request to the target’s voicemail service within the Date interval, and the voicemail service would have no way of knowing that the Identity header field it received had been originally signed for a call intended for a different number. However, note the caveats below in Section 12.1.1.

When signing a request that contains a fingerprint of keying material in SDP for DTLS-SRTP [RFC5763], this mechanism always provides a signature over that fingerprint. This signature prevents certain classes of impersonation attacks in which an attacker forwards or cut-and-pastes a legitimate request. Although the target of the attack may accept the request, the attacker will be unable to

exchange media with the target, as they will not possess a key corresponding to the fingerprint. For example, there are some baiting attacks, launched with the REFER method or through social engineering, where the attacker receives a request from the target and reoriginates it to a third party. These might not be prevented by only a signature over the From, To, and Date, but they could be prevented by securing a fingerprint for DTLS-SRTP. While this is a different form of impersonation than is commonly used for robocalling, ultimately there is little purpose in establishing the identity of the user that originated a SIP request if this assurance is not coupled with a comparable assurance over the contents of the subsequent media communication. This signature also reduces the potential for active eavesdropping attacks against the SIP media. In environments where DTLS-SRTP is unsupported, however, no field is signed and no protections are provided.

12.1.1. Protection of the To Header and Retargeting

Armed with the original value of the To header field, the recipient of a request may be tempted to compare it to their own identity in order to determine whether or not the identity information in this call might have been replayed. However, any request may be legitimately retargeted as well, and as a result legitimate requests may reach a SIP endpoint whose user is not identified by the URI designated in the To header field value. It is therefore difficult for any verifier to decide whether or not some prior retargeting was "legitimate". Retargeting can also cause confusion when identity information is provided for requests sent in the backwards direction in a dialog, as the dialog identifiers may not match credentials held by the ultimate target of the dialog. For further information on the problems of response identity, see [SIP-RETARGET].

Any means for authentication services or verifiers to anticipate retargeting is outside the scope of this document and is likely to have the same applicability to response identity as it does to requests in the backwards direction within a dialog. Consequently, no special guidance is given for implementers here regarding the "connected party" problem (see [RFC4916]); authentication service behavior is unchanged if retargeting has occurred for a dialog-forming request. Ultimately, the authentication service provides an Identity header field for requests in the dialog only when the user is authorized to assert the identity given in the From header field, and if they are not, an Identity header field is not provided. And per the threat model of [RFC7375], resolving problems with "connected" identity has little bearing on detecting robocalling or related impersonation attacks.

12.2. Unprotected Request Fields

[RFC4474] originally provided protections for Contact, Call-ID, and CSeq. This document removes protection for these fields. The absence of these header field values creates some opportunities for determined attackers to impersonate based on cut-and-paste attacks; however, the absence of these header field values does not seem impactful to the primary focus of this document, which is the prevention of the simple unauthorized claiming of an identity for the purposes of robocalling, voicemail hacking, or swatting.

It might seem attractive to provide a signature over some of the information present in the Via header field value(s). For example, without a signature over the sent-by field of the topmost Via header field, an attacker could remove that Via header field and insert its own in a cut-and-paste attack, which would cause all responses to the request to be routed to a host of the attacker's choosing. However, a signature over the topmost Via header field does not prevent attacks of this nature, since the attacker could leave the topmost Via intact and merely insert a new Via header field directly after it, which would cause responses to be routed to the attacker's host "on their way" to the valid host; the end result would be exactly the same. Although it is possible that an intermediary-based authentication service could guarantee that no Via hops are inserted between the sending UA and the authentication service, it could not prevent an attacker from adding a Via hop after the authentication service and thereby preempting responses. It is necessary for the proper operation of SIP for subsequent intermediaries to be capable of inserting such Via header fields, and thus it cannot be prevented. As such, though it is desirable, securing Via is not possible through the sort of identity mechanism described in this document; the best known practice for securing Via is the use of SIPS.

12.3. Malicious Removal of Identity Headers

In the end analysis, the Identity header field cannot protect itself. Any attacker could remove the header field from a SIP request and modify the request arbitrarily afterwards. However, this mechanism is not intended to protect requests from men-in-the-middle who interfere with SIP messages; it is intended only to provide a way that the originators of SIP requests can prove that they are who they claim to be. At best, by stripping identity information from a request, a man-in-the-middle could make it impossible to distinguish any illegitimate messages he would like to send from those messages sent by an authorized user. However, it requires a considerably greater amount of energy to mount such an attack than it does to mount trivial impersonations by just copying someone else's

From header field. This mechanism provides a way that an authorized user can provide a definitive assurance of his identity that an unauthorized user, an impersonator, cannot.

12.4. Securing the Connection to the Authentication Service

In the absence of UA-based authentication services, the assurance provided by this mechanism is strongest when a UA forms a direct connection, preferably one secured by TLS, to an intermediary-based authentication service. The reasons for this are twofold:

- If a user does not receive a certificate from the authentication service over the TLS connection that corresponds to the expected domain (especially when the user receives a challenge via a mechanism such as Digest), then it is possible that a rogue server is attempting to pose as an authentication service for a domain that it does not control, possibly in an attempt to collect shared secrets for that domain. A similar practice could be used for telephone numbers, though the application of certificates for telephone numbers to TLS is left as a matter for future study.

- Without TLS, the various header field values and the body of the request will not have integrity protection when the request arrives at an authentication service. Accordingly, a prior legitimate or illegitimate intermediary could modify the message arbitrarily.

Of these two concerns, the first is most material to the intended scope of this mechanism. This mechanism is intended to prevent impersonation attacks, not man-in-the-middle attacks; integrity over parts of the header and body is provided by this mechanism only to prevent replay attacks. However, it is possible that applications relying on the presence of the Identity header field could leverage this integrity protection for services other than replay protection.

Accordingly, direct TLS connections SHOULD be used between the UA client (UAC) and the authentication service whenever possible. The opportunistic nature of this mechanism, however, makes it very difficult to constrain UAC behavior, and moreover there will be some deployment architectures where a direct connection is simply infeasible and the UAC cannot act as an authentication service itself. Accordingly, when a direct connection and TLS are not possible, a UAC should use the SIPS mechanism, Digest "auth-int" for body integrity, or both when it can. The ultimate decision to add an Identity header field to a request lies with the authentication service, of course; domain policy must identify those cases where the UAC's security association with the authentication service is too weak.

12.5. Authorization and Transitional Strategies

Ultimately, the worth of an assurance provided by an Identity header field is limited by the security practices of the authentication service that issues the assurance. Relying on an Identity header field generated by a remote administrative domain assumes that the issuing domain uses recommended administrative practices to authenticate its users. However, it is possible that some authentication services will implement policies that effectively make users unaccountable (e.g., ones that accept unauthenticated registrations from arbitrary users). The value of an Identity header field from such authentication services is questionable. While there is no magic way for a verifier to distinguish "good" from "bad" signers by inspecting a SIP request, it is expected that further work in authorization practices could be built on top of this identity solution; without such an identity solution, many promising approaches to authorization policy are impossible. That much said, it is RECOMMENDED that authentication services based on proxy servers employ strong authentication practices.

One cannot expect the Identity header field to be supported by every SIP entity overnight. This leaves the verifier in a difficult position; when it receives a request from a given SIP user, how can it know whether or not the originator’s domain supports Identity? In the absence of ubiquitous support for Identity, some transitional strategies are necessary.

- A verifier could remember when it receives a request from a domain or telephone number that uses Identity and, in the future, view messages received from that source without an Identity header field with skepticism.

- A verifier could consult some sort of directory that indicates whether a given caller should have a signed identity. There are a number of potential ways in which this could be implemented. This is left as a subject for future work.

In the long term, some sort of identity mechanism, either the one documented in this specification or a successor, must become mandatory-to-use for SIP; that is the only way to guarantee that this protection can always be expected by verifiers.

Finally, it is worth noting that the presence or absence of the Identity header fields cannot be the sole factor in making an authorization decision. Permissions might be granted to a message on the basis of the specific verified Identity or really on any other aspect of a SIP request. Authorization policies are outside the

scope of this specification, but this specification advises any future authorization work not to assume that messages with valid Identity header fields are always good.

12.6. Display-Names and Identity

As a matter of interface design, SIP UAs might render the display-name portion of the From header field of a caller as the identity of the caller; there is a significant precedent in email user interfaces for this practice. Securing the display-name component of the From header field value is outside the scope of this document but may be the subject of future work, such as through the "ppt" name mechanism.

In the absence of signing the display-name, authentication services might check and validate it, and compare it to a list of acceptable display-names that may be used by the originator; if the display-name does not meet policy constraints, the authentication service could return a 403 response code. In this case, the reason phrase should indicate the nature of the problem: for example, "Inappropriate Display Name". However, the display-name is not always present, and in many environments the requisite operational procedures for display-name validation may not exist, so no normative guidance is given here.

13. IANA Considerations

IANA has completed a number of actions described in this document. Primarily, the previous references to [RFC4474] in the "Session Initiation Protocol (SIP) Parameters" registry have been updated to point to this document, unless specified otherwise below.

13.1. SIP Header Fields

The Identity-Info header in the SIP "Header Fields" registry has been marked as deprecated by this document.

Also, the Identity-Info header reserved the compact form "n" at its time of registration. That compact form has been removed from the registry. The Identity header, however, retains the compact form "y" reserved by [RFC4474].

13.2. SIP Response Codes

The 436 "Bad Identity-Info" default reason phrase has been changed to "Bad Identity Info" in the SIP "Response Codes" registry.

The 437 "Unsupported Certificate" default reason phrase has been changed to "Unsupported Credential".

13.3. Identity-Info Parameters

IANA manages a registry for Identity-Info parameters. Per this specification, IANA has changed the name of this registry to "Identity Parameters".

This specification defines one new value for the registry: "info" as defined in Section 7.3.

13.4. Identity-Info Algorithm Parameter Values

IANA managed an "Identity-Info Algorithm Parameter Values" registry; per this specification, IANA has deprecated and closed this registry. Since the algorithms for signing PASSporTs are defined in [RFC8225] rather than in this specification, there is no longer a need for an algorithm parameter registry for the Identity header field.

14. Changes from RFC 4474

The following are salient changes from the original RFC 4474:

- The credential mechanism has been generalized; credential enrollment, acquisition, and trust are now outside the scope of this document.

- This document reduces the scope of the Identity signature to remove CSeq, Call-ID, Contact, and the message body; signing of key fingerprints in SDP is now included.

- The Identity-Info header field has been deprecated, and its components have been relocated into parameters of the Identity header field (which obsoletes the previous version of the header field).

- The Identity header field can now appear multiple times in one request.

- The previous signed-identity-digest format has been replaced with PASSporT (signing algorithms are now defined in a separate specification).

- Status code descriptions have been revised.

15. References

15.1. Normative References

[E.164] International Telecommunication Union, "The international public telecommunication numbering plan", ITU-T Recommendation E.164, November 2010, <https://www.itu.int/rec/T-REC-E.164/en>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, DOI 10.17487/RFC3261, June 2002, <https://www.rfc-editor.org/info/rfc3261>.

[RFC3263] Rosenberg, J. and H. Schulzrinne, "Session Initiation Protocol (SIP): Locating SIP Servers", RFC 3263, DOI 10.17487/RFC3263, June 2002, <https://www.rfc-editor.org/info/rfc3263>.

[RFC3966] Schulzrinne, H., "The tel URI for Telephone Numbers", RFC 3966, DOI 10.17487/RFC3966, December 2004, <https://www.rfc-editor.org/info/rfc3966>.

[RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, DOI 10.17487/RFC3986, January 2005, <https://www.rfc-editor.org/info/rfc3986>.

[RFC5234] Crocker, D., Ed., and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, DOI 10.17487/RFC5234, January 2008, <https://www.rfc-editor.org/info/rfc5234>.

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

[RFC5922] Gurbani, V., Lawrence, S., and A. Jeffrey, "Domain Certificates in the Session Initiation Protocol (SIP)", RFC 5922, DOI 10.17487/RFC5922, June 2010, <https://www.rfc-editor.org/info/rfc5922>.

[RFC8225] Wendt, C. and J. Peterson, "PASSporT: Personal Assertion Token", RFC 8225, DOI 10.17487/RFC8225, February 2018, <https://www.rfc-editor.org/info/rfc8225>.

15.2. Informative References

[CIDER] Kaplan, H., "A proposal for Caller Identity in a DNS-based Entrusted Registry (CIDER)", Work in Progress, draft-kaplan-stir-cider-00, July 2013.

[IRI-COMPARISON] Masinter, L. and M. Duerst, "Comparison, Equivalence and Canonicalization of Internationalized Resource Identifiers", Work in Progress, draft-ietf-iri-comparison-02, October 2012.

[RFC2585] Housley, R. and P. Hoffman, "Internet X.509 Public Key Infrastructure Operational Protocols: FTP and HTTP", RFC 2585, DOI 10.17487/RFC2585, May 1999, <https://www.rfc-editor.org/info/rfc2585>.

[RFC3323] Peterson, J., "A Privacy Mechanism for the Session Initiation Protocol (SIP)", RFC 3323, DOI 10.17487/RFC3323, November 2002, <https://www.rfc-editor.org/info/rfc3323>.

[RFC3325] Jennings, C., Peterson, J., and M. Watson, "Private Extensions to the Session Initiation Protocol (SIP) for Asserted Identity within Trusted Networks", RFC 3325, DOI 10.17487/RFC3325, November 2002, <https://www.rfc-editor.org/info/rfc3325>.

[RFC3893] Peterson, J., "Session Initiation Protocol (SIP) Authenticated Identity Body (AIB) Format", RFC 3893, DOI 10.17487/RFC3893, September 2004, <https://www.rfc-editor.org/info/rfc3893>.

[RFC4474] Peterson, J. and C. Jennings, "Enhancements for Authenticated Identity Management in the Session Initiation Protocol (SIP)", RFC 4474, DOI 10.17487/RFC4474, August 2006, <https://www.rfc-editor.org/info/rfc4474>.

[RFC4501] Josefsson, S., "Domain Name System Uniform Resource Identifiers", RFC 4501, DOI 10.17487/RFC4501, May 2006, <https://www.rfc-editor.org/info/rfc4501>.

[RFC4916] Elwell, J., "Connected Identity in the Session Initiation Protocol (SIP)", RFC 4916, DOI 10.17487/RFC4916, June 2007, <https://www.rfc-editor.org/info/rfc4916>.

[RFC5393] Sparks, R., Ed., Lawrence, S., Hawrylyshen, A., and B. Campen, "Addressing an Amplification Vulnerability in Session Initiation Protocol (SIP) Forking Proxies", RFC 5393, DOI 10.17487/RFC5393, December 2008, <https://www.rfc-editor.org/info/rfc5393>.

[RFC5763] Fischl, J., Tschofenig, H., and E. Rescorla, "Framework for Establishing a Secure Real-time Transport Protocol (SRTP) Security Context Using Datagram Transport Layer Security (DTLS)", RFC 5763, DOI 10.17487/RFC5763, May 2010, <https://www.rfc-editor.org/info/rfc5763>.

[RFC6698] Hoffman, P. and J. Schlyter, "The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA", RFC 6698, DOI 10.17487/RFC6698, August 2012, <https://www.rfc-editor.org/info/rfc6698>.

[RFC6973] Cooper, A., Tschofenig, H., Aboba, B., Peterson, J., Morris, J., Hansen, M., and R. Smith, "Privacy Considerations for Internet Protocols", RFC 6973, DOI 10.17487/RFC6973, July 2013, <https://www.rfc-editor.org/info/rfc6973>.

[RFC8259] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", STD 90, RFC 8259, DOI 10.17487/RFC8259, December 2017, <https://www.rfc-editor.org/info/rfc8259>.

[RFC7340] Peterson, J., Schulzrinne, H., and H. Tschofenig, "Secure Telephone Identity Problem Statement and Requirements", RFC 7340, DOI 10.17487/RFC7340, September 2014, <https://www.rfc-editor.org/info/rfc7340>.

[RFC7375] Peterson, J., "Secure Telephone Identity Threat Model", RFC 7375, DOI 10.17487/RFC7375, October 2014, <https://www.rfc-editor.org/info/rfc7375>.

[RFC7515] Jones, M., Bradley, J., and N. Sakimura, "JSON Web Signature (JWS)", RFC 7515, DOI 10.17487/RFC7515, May 2015, <https://www.rfc-editor.org/info/rfc7515>.

[RFC7519] Jones, M., Bradley, J., and N. Sakimura, "JSON Web Token (JWT)", RFC 7519, DOI 10.17487/RFC7519, May 2015, <https://www.rfc-editor.org/info/rfc7519>.

[RFC8226] Peterson, J. and S. Turner, "Secure Telephone Identity Credentials: Certificates", RFC 8226, DOI 10.17487/RFC8226, February 2018, <https://www.rfc-editor.org/info/rfc8226>.

[SIP-RETARGET] Peterson, J., "Retargeting and Security in SIP: A Framework and Requirements", Work in Progress, draft-peterson-sipping-retarget-00, February 2005.

[SIP-RFC4474-CONCERNS] Rosenberg, J., "Concerns around the Applicability of RFC 4474", Work in Progress, draft-rosenberg-sip-rfc4474-concerns-00, February 2008.

[TS-3GPP.23.228] 3GPP, "IP Multimedia Subsystem (IMS); Stage 2", 3GPP TS 23.228 7.7.0, March 2007, <http://www.3gpp.org/ftp/Specs/html-info/23228.htm>.

Acknowledgments

The authors would like to thank Adam Roach, Jim Schaad, Ning Zhang, Syed Ali, Olle Jacobson, Dave Frankel, Robert Sparks, Dave Crocker, Stephen Kent, Brian Rosen, Alex Bobotek, Paul Kyzivat, Jonathan Lennox, Richard Shockey, Martin Dolly, Andrew Allen, Hadriel Kaplan, Sanjay Mishra, Anton Baskov, Pierce Gorman, David Schwartz, Eric Burger, Alan Ford, Christer Holmberg, Philippe Fouquart, Michael Hamer, Henning Schulzrinne, and Richard Barnes for their comments.

Authors' Addresses

Jon Peterson
Neustar, Inc.
1800 Sutter St. Suite 570
Concord, CA 94520
United States of America

Email: jon.peterson@neustar.biz

Cullen Jennings
Cisco
400 3rd Avenue SW, Suite 350
Calgary, AB T2P 4H2
Canada

Email: fluffy@cisco.com

Eric Rescorla
RTFM, Inc.
2064 Edgewood Drive
Palo Alto, CA 94303
United States of America

Email: ekr@rtfm.com

Chris Wendt
Comcast
One Comcast Center
Philadelphia, PA 19103
United States of America

Email: chris-ietf@chriswendt.net