Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 8, 2017

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

July 7, 2016

Authenticated Identity Management in the Session Initiation Protocol (SIP)
draft-ietf-stir-rfc4474bis-10.txt

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.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 8, 2017.

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 SIP URI, commonly a canonical address-of-record (AoR) employed to reach a user (such as 'sip:alice@atlanta.example.com'), or a telephone number, which can be represented as either 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, the recipient of a SIP request has no way to verify that the From header field has been populated appropriately, in the absence of some sort of cryptographic authentication mechanism. This leaves SIP vulnerable to a category of abuses, including impersonation attacks that enable robocalling and related problems as described in [RFC7340]. Ideally, a cryptographic approach to identity can provide a much stronger and less spoofable assurance of identity than the Caller ID services that the telephone network provides today.

[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 user agents 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 user agents need to be able to securely communicate their users' identity 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, RFC 4474 was in several ways misaligned with deployment realities (see [I-D.rosenberg-sip-rfc4474-concerns]). Most significantly, RFC 4474 did not deal well with telephone numbers as identifiers, despite their enduring use in SIP deployments. RFC 4474 also provided a signature over material that intermediaries in existing deployments commonly altered. This specification therefore

revises RFC 4474 in light of recent reconsideration of the problem space to align with the threat model in [RFC7375], and aligns the signature format with PASSporT [I-D.ietf-stir-passport].

2. Terminology

In this document, the key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" are to be interpreted as described in RFC 2119 [RFC2119] and RFC 6919 [RFC6919].

3. Background

Per [RFC7340], problems such as robocalling, voicemail hacking, and swatting are enabled by an attacker's ability to impersonate someone else. The secure operation of most SIP applications and services depends on authorizing the source of communications as it is represented in a SIP request. Such authorization policies can be automated or be a part of human operation of SIP devices. An example of the former would be a voicemail service that compares the identity of the caller to a whitelist before determining whether it should allow the caller access to recorded messages. An example of the latter would be an Internet telephone application that displays the calling party number (and/or Caller-ID) of a caller, which a human may review to make a policy decision before answering a call. In both of these cases, attackers might attempt to circumvent these authorization policies through impersonation. Since the primary identifier of the sender of a SIP request, the From header field, can be populated arbitrarily by the controller of a user agent, impersonation is very simple today in many environments. The mechanism described in this document provides a strong identity system for detecting attempted impersonation in SIP requests.

This identity architecture for SIP depends on a logical "authentication service" which validates outgoing requests. An authentication service may be implemented either as part of a user agent or as a proxy server; typically, it is a component of a network intermediary like a proxy to which originating user agents send unsigned requests. Once the sender of the message has been authenticated, the authentication service then computes and adds cryptographic information (including a digital signature over some components of messages) to requests to communicate to other SIP entities that the sending user has been authenticated and its claim of a particular identity has been authorized. A "verification service" on the receiving end then validates this signature and enables policy decisions to be made based on the results of the verification.

Identities are issued to users by authorities. When a new user becomes associated with example.com, the administrator of the SIP service for that domain can issue them an identity in that namespace, such as alice@example.com. Alice may then send REGISTER requests to example.com that make her user agents eligible to receive requests for sip:alice@example.com. In some cases, Alice may be the owner of the domain herself, 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 baseline SIP, the credentials needed to prove a user is authorized to use a particular From header field must ultimately derive from the domain owner: either a user agent gives requests to the domain name owner in order for them to be signed by the domain owner's credentials, or the user agent must possess credentials that prove in some fashion that the domain owner has given the user agent the right to a name.

The situation is however more complicated for telephone numbers, however. Authority over telephone numbers does not correspond directly to Internet domains. While a user could register at a SIP domain with a username that corresponds to a telephone number, any connection between the administrator of that domain and the assignment of telephone numbers is not currently reflected on the Internet. Telephone numbers do not share the domain-scope property described above, as they are dialed without any domain component. This document thus assumes the existence of a separate means of establishing authority over telephone numbers, for cases where the telephone number is the identity of the user. As with SIP URIs, the necessary credentials to prove authority for a name might reside either in the endpoint or at some intermediary.

This document specifies a means of sharing a cryptographic assurance of end-user SIP identity in an interdomain or intradomain context. It relies on the authentication service constructing tokens based on the PASSporT [I-D.ietf-stir-passport] format, a JSON [RFC7159] object comprising values copied from certain header field values in the SIP request. The authentication service then computes a signature over those JSON object in a manner following PASSporT. That signature is then placed in a SIP Identity header. In order to assist in the validation of the Identity header, this specification also describes some metadata fields associated with the header 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 this 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 PASSporT objects to other protocols.

This specification allows either a user agent or a proxy server to provide the authentication service function and/or the verification service function. To maximize end-to-end security, it is obviously preferable for end-users to acquire their own credentials; if they do, their user agents 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 user agents (phones, PCs, laptops, PDAs, gaming devices) that may be employed by a single user. In such environments, 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. In these cases, implementation the authentication service at an intermediary may be more practical. This trade-off needs to be understood by implementers of this specification.

4. Overview of Operations

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

Imagine a case where Alice, who has the home proxy of example.com and the address-of-record sip:alice@example.com, wants to communicate with Bob at sip:bob@example.org. They have no prior relationship, and Bob implements best practices to prevent impersonation attacks.

Alice generates an INVITE and places her identity, in this case her address-of-record, in the From header field of the request. She then sends an INVITE over TLS to an authentication service proxy for the example.com domain.

The proxy authenticates Alice (possibly by sending a Digest authentication challenge), and validates that she is authorized to assert the identity that she populated in the From header field. This value could be Alice's AoR, but in other cases it could be some different value that the authentication service has authority over, such as a telephone number. The proxy authentication service then constructs a PASSporT object which contains a JSON representations of headers and claims which mirror certain parts of the SIP request, including the identity in the From header field. As a part of generating the PASSporT object, the authentication service signs a hash of those headers and claims with the appropriate credential for the identity (in this case, the certificate for example.com, which covers the identity sip:alice@example.com), and the signature is inserted by the proxy server into the Identity header field value of the request. Optionally, the JSON headers and claims themselves may also be included in the object, encoded in the "canon" parameter of the Identity header.

The proxy, as the holder of the private key for the example.com domain, is asserting that the originator of this request has been authenticated and that she is authorized to claim the identity that appears in the From header field. The proxy inserts an "info" parameter into the Identity header that tells Bob how to acquire keying material necessary to validate its credentials (a public key), in case he doesn't already have it.

When Bob's domain receives the request, it verifies the signature provided in the Identity header, and thus can validate that the authority over the identity in the From header field authenticated the user, and permitted the user to assert that From header field value. This same validation operation may be performed by Bob's user agent server (UAS). As the request has been validated, it is rendered to Bob. If the validation was unsuccessful, some other treatment would be applied by the receiving domain.

5. Signature Generation and Validation

5.1. Authentication Service Behavior

This document specifies a role for SIP entities called an authentication service. The authentication service role can be instantiated, for example, by an intermediary such as a proxy server or by a user agent. 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 6.1). 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 these entities 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 to SIP requests. The procedures below define the steps that must be taken when each an header is added. More than one may appear in a single request, and an authentication service may add an Identity header to a request that already contains one or more Identity headers. If the Identity header added follows extended signing procedures beyond the baseline given in Section 8, then it differentiates the header with a "ppt" parameter per the fourth step below.

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

Step 1:

First, the authentication service must determine whether it is authoritative for the identity of the sender of the request. In ordinary operations, the authentication service decides this by inspecting the URI value from the addr-spec component of From header field; this URI will be referred to here as the 'identity field'. If the identity field contains a SIP or SIP Secure (SIPS) URI, and the user portion is not a telephone number, the authentication service MUST extract the hostname portion of the identity field and compare it to the domain(s) for which it is responsible (following the procedures in RFC 3261 [RFC3261], Section 16.4). If the identity field uses the TEL URI scheme [RFC3966], or the identity field is a SIP or SIPS URI with a telephone number in the user portion, the authentication service determines whether or not it is responsible for this telephone number; see Section 7.1 for more information. An authentication service proceeding with a signature over a telephone number MUST then follow the canonicalization procedures described in Section 7.2. 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 if the authentication service does not have the authority to make the claim it asserts.

Step 2:

The authentication service MUST then determine whether or not the sender 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 sender of the message. Some possible ways in which this authentication might be performed include:

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 user agent, a user agent may authenticate its own user through any system-specific means, perhaps simply by virtue of having physical access to the user agent.

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 6.1 for more information.

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

Step 3:

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 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 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. This behavior is not mandatory because a user agent client (UAC) could only exploit the Date header in order to cause a request to fail verification; the Identity header is not intended to provide a source of non-repudiation or a perfect record of when messages are processed. Finally, the authentication service MUST verify that both the Date header and the current time fall within the validity period of its credential.

See Section 12 for information on how the Date header field assists verifiers.

Step 4:

Subsequently, the authentication service MUST form a PASSporT object and add a corresponding an Identity header to the request containing this signature. For baseline PASSporT objects headers (without an Identity header "ppt" parameter), this follows the procedures in Section 8; 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 has been added to the request, the authentication service MUST also add a "info" parameter to the Identity header. The "info" parameter contains a URI from which the authentication service's credential can be acquired; see Section 6.3 for more on credential acquisition.

Step 5:

In the circumstances described below, an authentication service will add a "canon" parameter to the Identity header. The syntax of "canon" is given in Section 8; essentially, it contains a base64 encoding of the JSON header and claims in the PASSporT object. The presence of "canon" is OPTIONAL baseline PASSporT objects in SIP as a because the information carried in the baseline PASSporT object's headers and claims is usually redundant with information already carried elsewhere in the SIP request. Omitting "canon" can significantly reduce SIP message size, especially when the PASSporT object contains media keys.

When however an authentication service creates a PASSporT that uses extension claims beyond the baseline PASSporT object, including "canon" is REQUIRED in order for the verification service to be capable of validating the signature. See Section 9.

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 indicating a verification failure. If the authentication service did not originally send the Identity header with the "canon" parameter, it SHOULD retry a request once after receiving a 438 response, this time including the "canon". The information in "canon" is useful on the verification side for debugging errors, and there are some known causes of verification failures (such as the Date header changing in transit, see Section 12.1 for more information) that can be resolved by the inclusion of "canon".

Finally, the authentication service MUST forward the message normally.

5.2. Verifier Behavior

This document specifies a logical role for SIP entities called a verification service, or verifier. When a verifier receives a SIP message containing one or more Identity headers, it inspects the signature(s) to verify the identity of the sender 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 headers. A verification service performs the procedures above on each Identity header that appears in a request. If the verifier does not support an Identity header present in a request due to the presence of an unsupported "ppt" parameter, or if no Identity header is present, and the presence of an Identity header 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 failure responses, see Section 13.3.

In order to verify an Identity header in a message, an entity acting as a verifier MUST perform the following steps, in the order here specified. Note that when an Identity header contains the optional "canon" parameter, the verifier MUST follow the additional procedures in Section 5.2.1.

Step 1:

The verifier MUST inspect any optional "ppt" parameter appearing the Identity request. If no "ppt" parameter is present, then the verifier proceeds normally below. If a "ppt" parameter value is present, and the verifier does not support it, it MUST ignore the Identity header. If a supported "ppt" parameter value is present, the verifier follows the procedures below, including the variations described in Step 5.

Step 2:

In order to determine whether the signature for the identity field should be over the entire identity field URI or just a canonicalized telephone number, the verification service MUST follow the canonicalization process described in Section 7.2. That section also describes the procedures the verification service MUST follow to determine if the signer is authoritative for a telephone number. For domains, the verifier MUST follow the process described in Section 7.3 to determine if the signer is authoritative for the identity field.

Step 3:

The verifier must first ensure that it possesses the proper keying material to validate the signature in the Identity header field, which usually involves dereferencing a URI in the "info" parameter of the Identity header. See Section 6.2 for more information on these procedures. If the verifier does not support the credential described in the "info" parameter, then it should consider the credential for this header unsupported. If a SIP request contains no Identity headers with a supported credential, then the verifier MUST return a 437 "Unsupported Credential" response.

Step 4:

The verifier MUST furthermore ensure that the value of the Date header of the request meets local policy for freshness (usually,

within sixty seconds) and that it falls within the validity period of the credential used to sign the Identity header. For more on the attacks this prevents, see Section 12.1. If the "canon" parameter is present, the verifier should follow the Date-related behavior in Section 5.2.1.

Step 5:

The verifier MUST validate the signature in the Identity header field over the PASSporT object. For baseline PASSporT objects (with no Identity header "ppt" parameter) the verifier MUST follow the procedures for generating the signature over a PASSporT object described in Section 8. If a "ppt" parameter is present (and per Step 1, is understood), the verifier follows the procedures for that "ppt" (see Section 9). If a verifier determines that the that the signature in the Identity does not correspond to the reconstructed signed-identity-digest, then the Identity header should be considered invalid.

The presence of multiple Identity headers within a message raises the prospect that a verification services could receive a message containing some valid and some invalid Identity headers. If the verifier determines all Identity headers within a message are invalid, then a 438 'Invalid Identity Header' response MUST be returned.

The verification of an Identity header does not entail any particular treatment of the request. The handling of the message after the verification process depends on how the implementation service is implemented and on local policy. This specification does not propose any authorization policy for user agents or proxy servers to follow based on the presence of a valid Identity header, the presence of an invalid Identity header, or the absence of an Identity header, but 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 user agent implementations.

5.2.1. Handling 'canon' parameters

If the optional "canon" parameter of the Identity header is present, it contains a base64 encoding of the header and claim component of the PASSporT object constructed by the authentication service, and this it conveys any canonical telephone number formats created by the authentication service (see Section 7.2), as well as an "iat" claim corresponding to the Date header that the authentication service used. The "canon" is provided purely as an optimization and debugging mechanism for the verification service.

When "canon" is present, the verification service MAY compute its own canonicalization of the numbers and compare them to the values in the "canon" parameter before performing any cryptographic functions in order to ascertain whether or not the two ends agree on the canonical number form. Also, when "canon" is present, during Step 4 the verification service SHOULD compare the "iat" value in the "canon" to its Date header field value. If the two are different, and the "iat" value is later but 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 value. As some deployments in the field have been observed to change the Date header in transit, this procedure will prevent some unnecessary verification failures.

6. Credentials

6.1. Credential Use by the Authentication Service

In order to act as an authentication service, a SIP entity must have access to the private keying material of one or more credentials that cover domain names or telephone numbers. These credentials may represent authority over an entire domain (such as example.com) or potentially 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 is expressed is specific to the credential mechanism.

Authorization of the use of a particular username or telephone number in the identity field 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 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 Proxy-Authorization that do not correspond to the user-portion of SIP From headers, 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 Proxy-Authorization 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 in Third Generation Partnership Project (3GPP), 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. 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.

6.2. Credential Use by the Verification Service

In order to act as a verification service, a SIP entity must have a way to acquire and retain credentials for authorities over particular domain names and/or telephone numbers or number ranges. Dereferencing the URI found in the "info" parameter of the Identity header (as described in the next section) 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 more other implementation-specific way of acquiring the needed keying material, such as an offline store of periodically-updated credentials. If the URI in the "info" parameter cannot be dereferenced, then a 436 'Bad Identity-Info' response MUST be returned.

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 the credential mechanism specifications, which must meet the requirements in Section 6.4.

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, or the URI given in the "info" parameter value. Further consideration of how to cache credentials is deferred to the credential mechanism specifications.

6.3. Handling 'info' parameter URIs

An "info" parameter MUST contain a URI which 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 must be dereferencable by any entity on the public Internet. In constrained deployment environments, a service private to the environment might 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 and a user agent 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.

6.4. 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., 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 may 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 [I-D.ietf-stir-certificates], but this specification can work with other credential systems; for example, using the DNS was proposed in [I-D.kaplan-stir-cider].

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

- which URIs 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)

It is furthermore required that all credential specifications describe how the associated credentials will support the mandatory signing algorithm(s) required by PASSporT [I-D.ietf-stir-passport].

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-Info 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 Identity-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).

7. Identity Types

This specification 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, this specification also works with "greenfield" identifiers (of the form "sip:user@host"), and potentially other identifiers when SIP interworks with another protocol.

The guidance in this section 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 environments, the P-Asserted-Identity header field is used in lieu of the From header field to convey the address-of-record or telephone number of the sender of a request; while it is not envisioned that many of those networks would or should make use of the Identity mechanism described in this specification, where they do, local policy might therefore dictate that the canonical identity derive from the P-Asserted-Identity header field rather than the From.

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

It may not be trivial to tell if a given URI contains a telephone number. 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 which contains a numeric user part. Note that the same procedures are followed for creating the canonical form of URIs found in the From header field as they are in the To header field or the P-Asserted-Identity header field.

First, implementations must look for obvious indications that 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 that URI conforms to the syntax of the TEL URI scheme (RFC 3966 [RFC3966]). It is also possible for a TEL URI to appear in the SIP To or From header field outside the context of a SIP or SIPS URI (e.g., 'tel:+17005551008'). Thus, in some environments, numbers will be explicitly labeled by the use of TEL URIs or the 'user=phone' parameter, or implicitly by the presence of the '+' indicator at the start of the user-portion. Absent these indications, if there are numbers present in the user-portion, implementations may 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, a nationally-specific number, or even a private branch exchange number. Once a telephone number has been detected, implementations should follow the procedures in Section 7.2.

If the URI field does not contain a telephone number, URI normalization procedures are invoked to canonicalize the URI before it is included in a PASSporT object in, for example, an "uri" claim. See Section 7.4 for that behavior.

7.1. Authority for Telephone Numbers

In order for telephone numbers to be used with the mechanism described in this document, authentication services must enroll with an authority that issues credentials authoritative 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 6.4, enrollment procedures and credential management are outside the scope of this document; approaches to credential management for telephone numbers are discussed in [I-D.ietf-stir-certificates].

7.2. Telephone Number Canonicalization Procedures

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

Implementations MUST drop any leading +'s, any internal dashes, parentheses or other non-numeric characters, excepting only the leading "#" 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, the routing procedures associated with those numbers will likely make sure that the verification service understands the context of their use.

Other transformations during canonicalization MAY 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 user portions to E.164 by prepending

country-code and region code digits; another domain might prefix usernames with trunk-routing codes and 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*DIGIT
```

If the result of this procedure forms a complete telephone number, that number is used for the purpose of creating and signing the signed-identity-string by both the authentication service and verification service. Practically, entities that perform the authentication service role will sometimes alter the telephone numbers that appear in the To and From header field values, converting them to this format (though note this is not a function that [RFC3261] permits proxy servers to perform). The result of the canonicalization process of the From header field value may also be recorded through the use of the "canon" parameter of the Identity(see Section 8).

If the result of the canonicalization of the From header field value does not form a complete and valid telephone number, the authentication service and/or verification service SHOULD treat the entire URI as a SIP URI, and apply the procedures in Section 7.4.

7.3. Authority for Domain Names

When a verifier processes a request containing an Identity-Info header with a domain signature, it must compare the domain portion of the URI in the From header field of the request with the domain name that is the subject of the credential acquired from the "info" parameter. While it might seem that this should be a straightforward process, it 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. Secondly, some SIP services may delegate SIP functions to a subordinate domain and utilize the procedures in RFC 3263 [RFC3263] that allow requests for, say, 'example.com' to be routed to 'sip.example.com'. As a result, a user with the AoR 'sip:jon@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 meet the second of these problems, a domain that deploys an authentication service on a subordinate host MUST be willing to 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 RFC 3263 are used to direct requests to a domain name other than the domain in the original Request-URI (e.g., for 'sip:jon@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.

A verifier MUST evaluate the correspondence between the user's identity and the signing credential by following the procedures defined in RFC 2818 [RFC2818], Section 3.1. While RFC 2818 [RFC2818] deals with the use of HTTP in TLS and is specific to certificates, the procedures described are applicable to verifying identity if one substitutes the "hostname of the server" in HTTP for the domain portion of the user's identity in the From header field of a SIP request with an Identity header.

7.4. URI Normalization

Just as telephone numbers may undergo a number of syntactic transformation 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 which captures the address-of-record 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 headers, from its ayntax. 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 syntax-based URI normalization procedures of [RFC3986] Section 6.2.2, especially to perform case normalization and percent-encoding normalization. However, note that normalization procedures face known challenges in some internationalized environments (see [I-D.ietf-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 eliminate the scheme component. A more exact definition is left to future specifications.

8. Header Syntax

The Identity and Identity-Info headers that were previously defined in RFC4474 are deprecated. This revised specification collapses the grammar of Identity-Info into the Identity header via the "info" parameter. Note that unlike the prior specification in RFC4474, the Identity header is now allowed to appear more than one time in a SIP request. The revised grammar for the Identity header is (following the ABNF [RFC4234] in RFC 3261 [RFC3261]):

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

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

In addition to "info" parameter, and the "alg" parameter previously defined in RFC4474, this specification includes the optional "canon" and "ppt" parameters. Note that in RFC4474, the signed-identity-digest (see ABNF above) was given as quoted 32LHEX, whereas here it is given as a quoted sequence of base64-char.

The 'absoluteURI' portion of ident-info-uri MUST contain a URI; see Section 6.3 for more on choosing how to advertise credentials through this parameter.

The signed-identity-digest is the signed hash component of a PASSporT object [I-D.ietf-stir-passport], a signature which PASSporT generates over a pair of JSON objects. The first PASSporT object contains header information, and the second contains claims, following the conventions of JWT [RFC7519]; some header and claim values will mirror elements of the SIP request. Once these two JSON objects have been generated, they will be encoded, then hashed with a SHA-256 hash. Those two hashes are then concatenated (header then claims) into a string separated by a single "." per baseline PASSporT. Finally, that string is signed to generate the signed-identity-digest value of the Identity header.

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

First, per baseline [I-D.ietf-stir-passport], the JSON key "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. Note if the "alg" parameter is absent, the default value is "ES256".

Third, the JSON key "x5u" MUST have a value equivalent to the quoted URI in the "info" parameter.

Fourth, the optional JSON key "ppt", if present, MUST have a value equivalent to the quoted value of the "ppt" parameter of the Identity header. If the "ppt" parameter is absent from the header, the "ppt" key MUST NOT not appear in the JSON heaer object.

For example:

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

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

First, the JSON "orig" array MUST be populated. If the originating identity is a telephone number, then the array MUST be populated with a "tn" claim with a value set to the value of the quoted originating identity, a canonicalized telephone number (see Section 7.2). Otherwise, the array MUST be populated with a "uri" claim, set to the value of the AoR of the UA sending the message as taken from addr-spec of the From header field, per the procedures in Section 7.4.

Second, the JSON "dest" array MUST be populated. If the destination identity is a telephone number, then the array MUST be populated with a "tn" claim with a value set to the value of the quoted destination identity, a canonicalized telephone number (see Section 7.2). Otherwise, the array MUST be populated with a "uri" claim, 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 7.4.

Third, the JSON key "iat" MUST appear, set to the value of a quoted encoding of the value of the SIP Date header field as a JSON NumericDate (as UNIX time, per [RFC7519] Section 2).

Fourth, if the request contains an 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 [I-D.ietf-stir-passport] Section 3.2.2.2.

For example:

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

For more information on the security properties of these SIP message elements, and why their inclusion mitigates replay attacks, see Section 12 and [RFC3893]. Note that future extensions to the PASSporT object could introduce new claims, and that further SIP procedures could be required to extract further information from the SIP request to populate the values of those claims; see Section 9.

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

"orig" and "dest" arrays might 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 provision multiple originating or destination identities is left as a subject for future specification.

After these two JSON objects, the header and the claims, have been constructed, they must each be hashed per [I-D.ietf-stir-passport] Section 3.3. The signed value of those concatenated hashes then becomes the signed-identity-string of the Identity header. The hashing and signing algorithm is specified by the 'alg' parameter of the Identity header and the mirrored "alg" parameter of PASSporT. This specification inherits from the PASSporT specification one value for the 'alg' parameter: 'ES256', as defined in [RFC7519], which connotes an ECDSA P-256 digital signature. All implementations of this specification MUST support the required signing algorithms of PASSporT.

The complete form of the Identity header will therefore look like the following example:

```
Identity: "sv5CTo05KqpSmtHt3dcEiO/1CWTSZtnG3iV+1nmurLXV/HmtyNS7Ltrg9dlxkWzo
    eU7d7OV8HweTTDobV3itTmgPwCFjaEmMyEI3d7SyN21yNDo2ER/Ovgtw0Lu5csIp
    pPqOg1uXndzHbG7mR6Rl9BnUhHufVRbp51Mn3w0gfUs="; \
        info=<https://biloxi.example.org/biloxi.cer>;alg=ES256
```

In a departure from JWT practice, the SIP usage of PASSporT MAY NOT include the base64 encoded version of the JSON objects in the Identity header: only the signature component of the PASSporT is REQUIRED. Optionally, as a debugging measure or optimization, the base64 encoded concatenation of the JSON header and claims may be included as the value of a "canon" parameter of the Identity header. Note that this may be lengthy string.

9. Extensibility

For the extensibility of baseline PASSporT with now claims, see [I-D.ietf-stir-passport] Section 4.

As future requirements may warrant increasing the scope of the Identity mechanism, this specification defines an optional "ppt" parameter of the Identity header, which mirrors the "ppt" header key 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 [I-D.ietf-stir-passport].

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.

If any claim in an extension contains a JSON value that does not correspond to any field of the SIP request, but then the optional "canon" parameter MUST be used for the Identity header containing that extension.

10. Backwards Compatibililty with RFC4474

This specification introduces several significant changes from the RFC4474 version of the Identity header. However, due to the problems enumerated in [I-D.rosenberg-sip-rfc4474-concerns], it is not believed that the original Identity header 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 RFC4474-compliant implementations, nor any related backwards-compatibility provisions. Hypothetically, were an RFC4474-compliant implementation to receive messages containing this revised version of the Identity header, it would likely fail the request due to the absence of an Identity-Info header with a 436 response code. Implementations of this specification, for debugging purposes, might interpret a 436 with a reason phrase of "Bad Identity-Info" as an indication that the request has failed because it reached a (hypothetical) RFC4474-compliant verification service.

11. Privacy Considerations

The purpose of this mechanism is to provide a strong identification of the originator of a SIP request, specifically a cryptographic assurance that an authority asserts the originator can claim the URI given in the From header field. This URI may contain 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 both as a privacy service and as an

authentication service. Since a user agent 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 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 authentications 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 of the request; accordingly, using domains like 'anonymous.invalid' will not be possible for privacy services that also 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 "anonymous.invalid" domain 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. The sort of assertion provided by the P-Asserted-Identity header is very different from the Identity header presented in this document. It contains additional information about the sender of a message that may go beyond what appears in the From header field; P-Asserted-Identity holds a definitive identity for the sender 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, and privacy services MUST NOT remove the Identity header when a priv-value of "id" appears in a Privacy header.

The optional "canon" parameter of the Identity header specified in this document provides the complete JSON objects used to generate the signed-identity-digest of the Identity header, 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 which differs substantially from the original From header field. Depending on those policies, potentially the "canon" parameter 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 "canon" would need to be removed 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 sender's identity could not be reassembled by a verifier, and thus the Identity signature validation process would fail, using P-Asserted-Identity with the Identity "canon" parameter 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.

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 headers are the same as those given in [RFC3261] for including headers in tunneled 'message/sip' MIME bodies (see Section 23 of RFC3261 in particular). The following 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 sender of the message. The SIP address-of-record URI, or an embedded telephone number, in the From header field is the identity of a SIP user, for the purposes of this document. Note that in some deployments the identity of the sender may reside in P-

Asserted-Id instead. The sender'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. Implementations of this specification MUST NOT deem valid a request with an outdated Date header field (the RECOMMENDED interval is that the Date header must indicate a time within 60 seconds of the receipt of a message). Note that per baseline [RFC3261] behavior, servers keep state of recently received requests, and thus if an Identity header 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, and that alternative behavior might be necessary to accommodate that use case. Verification services that observe a signature validation failure MAY therefore reconstruct the Date header field component of the signature from the "iat" carried in PASSporT via the "canon" parameter: 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 "canon" parameter 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 from 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, which 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 forward the INVITE to the target's voicemail service within the Date interval, and the voicemail service would have no way knowing that the Identity header 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 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, per [RFC7258], reduces the potential for passive monitoring 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

The mechanism in this document provides a signature over the identity information in the To header field value of requests. This provides a means for verifiers to detect replay attacks where a signed request originally sent to one target is modified and then forwarded by an attacker to another, unrelated target. Armed with the original value of the To header field, the recipient of a request may 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 [I-D.peterson-sipping-retarget].

Any means for authentication services or verifiers to anticipate retargeting is outside the scope of this document, and likely to have equal 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

for requests in the backwards dialog when the user is authorized to assert the identity given in the From header field, and if they are not, an Identity header 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 had protections for the Contact, Call-ID and CSeq. These are removed from RFC4474bis. The absence of these header values creates some opportunities for determined attackers to impersonate based on cut-and-paste attacks; however, the absence of these headers does not seem impactful to preventing the simple unauthorized claiming of an identity for the purposes of robocalling, voicemail hacking, or swatting, which is the primary scope of the current document.

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, an attacker could remove that Via header 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 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, which has exactly the same end result. Although it is possible that an intermediary-based authentication service could guarantee that no Via hops are inserted between the sending user agent 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 cannot protect itself. Any attacker could remove the header 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 user agent-based authentication services, the assurance provided by this mechanism is strongest when a user agent 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 the header and bodies is provided by this mechanism only to prevent replay attacks. However, it is possible that applications relying on the presence of the Identity header could leverage this integrity protection for services other than replay protection.

Accordingly, direct TLS connections SHOULD be used between the 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 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 is limited by the security practices of the authentication service that issues the assurance. Relying on an Identity header 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 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 to be supported by every SIP entity overnight. This leaves the verifier in a compromising position; when it receives a request from a given SIP user, how can it know whether or not the sender'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 sources without Identity headers with skepticism.

A verifier could consult some sort of directory that indications 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 the SIP protocol; 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 headers 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 headers are always good.

12.6. Display-Names and Identity

As a matter of interface design, SIP user agents 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 sender; 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

This document relies on the headers and response codes defined in RFC 4474. It also retains the requirements for the specification of new algorithms or headers related to the mechanisms described in that document.

13.1. Identity-Info Parameters

The IANA has already created a registry for Identity-Info parameters. This specification defines a new value called "canon" as defined in Section 6.3. Note however that unlike in RFC4474, Identity-Info parameters now appear in the Identity header.

13.2. Identity-Info Algorithm Parameter Values

The IANA has already created a registry for Identity-Info "alg" parameter values. Note that now, the "alg" parameter appears in the Identity header rather than the deprecated Identity-Info header. Since the algorithms for signing PASSporT objects are defined in PASSporT rather than in this specification, there is no longer a need for an algorithm parameter registry for the Identity header. This registry is therefore deprecated.

13.3. Response Codes defined in RFC4474

RFC4474 defined four response codes for failure conditions specific to the Identity header and its original mechanism. These status codes are retained in this specification, with some modifications.

The semantics of the 428 'Use Identity Header' response code are slightly altered by the potential presence of the "ppt" parameter. Now, a 428 response MUST be sent when an Identity header is required, but no Identity header without a "ppt" parameter, or with a supported "ppt" value, has been received. In the case where one or more Identity headers with unsupported "ppt" values have been received, then a verification service SHOULD send a 428 with the reason phrase "Use Supported PASSporT Format". Note however that this specification gives no guidance on how a verification service might decide to require an Identity header for a particular SIP request. Such authorization policies are outside the scope of this specification.

For 436 'Bad Identity-Info' response, the default reason phrase is now renamed 'Bad Identity info', as the deprecation of the Identity-Info header has made 'info' a parameter of the Identity header. Again, given the potential presence of multiple Identity headers, this response code is sent when the verification service is unable to deference the URIs and/or acquire the credentials associated with all Identity headers 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 Certificate' default reason phrase is now changed to 'Unsupported Credential'. This response is sent when a verification service can acquire, or already holds, the credential represented by the 'info' parameter of at least one Identity header in the request, but does not support said credential(s), for reasons such as failing to trust the issuing CA, or failing to support the algorithm with which the credential was signed.

Finally, the 438 'Invalid Identity Header' response now indicates that of the set of Identity headers in a request, no header 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 is grounds for rejecting a request. Note that in some cases, an Identity header 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 of the request. Relying on the full PASSporT object presented through the "canon" parameter can repair some of these conditions (see Section 5.2.1), so the recommended way to attempt to repair this failure is to retry the request with "canon".

14. Acknowledgments

The authors would like to thank Stephen Kent, Brian Rosen, Alex Bobotek, Paul Kyzviat, Jonathan Lennox, Richard Shockey, Martin Dolly, Andrew Allen, Hadriel Kaplan, Sanjay Mishra, Anton Baskov, Pierce Gorman, David Schwartz, Eric Burger, Alan Ford, Philippe Fouquart, Michael Hamer, Henning Schulzrinne, and Richard Barnes for their comments.

15. Changes from RFC4474

The following are salient changes from the original RFC 4474:

Generalized the credential mechanism; credential enrollment, acquisition and trust is now outside the scope of this document

Reduced the scope of the Identity signature to remove CSeq, Call-ID, Contact, and the message body

Removed the Identity-Info header and relocated its components into parameters of the Identity header

The Identity header can now appear multiple times in one request

Replaced previous signed-identity-digest format with PASSporT (signing algorithms now defined there)

Revised status code descriptions

16. References

16.1. Normative References

[I-D.ietf-stir-passport]
Wendt, C. and J. Peterson, "Persona Assertion Token", draft-ietf-stir-passport-03 (work in progress), June 2016.

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

[RFC2818] Rescorla, E., "HTTP Over TLS", RFC 2818, DOI 10.17487/RFC2818, May 2000, <http://www.rfc-editor.org/info/rfc2818>.

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

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

[RFC3280] Housley, R., Polk, W., Ford, W., and D. Solo, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 3280, DOI 10.17487/RFC3280, April 2002, <http://www.rfc-editor.org/info/rfc3280>.

[RFC3370] Housley, R., "Cryptographic Message Syntax (CMS) Algorithms", RFC 3370, DOI 10.17487/RFC3370, August 2002, <http://www.rfc-editor.org/info/rfc3370>.

[RFC3966] Schulzrinne, H., "The tel URI for Telephone Numbers", RFC 3966, DOI 10.17487/RFC3966, December 2004, <http://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, <http://www.rfc-editor.org/info/rfc3986>.

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

[RFC6919] Barnes, R., Kent, S., and E. Rescorla, "Further Key Words for Use in RFCs to Indicate Requirement Levels", RFC 6919, DOI 10.17487/RFC6919, April 2013, <http://www.rfc-editor.org/info/rfc6919>.

16.2. Informative References

[I-D.ietf-iri-comparison]
Masinter, L. and M. DĂźrst, "Comparison, Equivalence and Canonicalization of Internationalized Resource Identifiers", draft-ietf-iri-comparison-02 (work in progress), October 2012.

[I-D.ietf-stir-certificates]
Peterson, J. and S. Turner, "Secure Telephone Identity Credentials: Certificates", draft-ietf-stir-certificates-06 (work in progress), July 2016.

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

[I-D.peterson-sipping-retarget]
Peterson, J., "Retargeting and Security in SIP: A Framework and Requirements", draft-peterson-sipping-retarget-00 (work in progress), February 2005.

[I-D.rosenberg-sip-rfc4474-concerns]
Rosenberg, J., "Concerns around the Applicability of RFC 4474", draft-rosenberg-sip-rfc4474-concerns-00 (work in progress), February 2008.

[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, <http://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, <http://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, <http://www.rfc-editor.org/info/rfc3325>.

[RFC3548] Josefsson, S., Ed., "The Base16, Base32, and Base64 Data Encodings", RFC 3548, DOI 10.17487/RFC3548, July 2003, <http://www.rfc-editor.org/info/rfc3548>.

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

[RFC4234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", RFC 4234, DOI 10.17487/RFC4234, October 2005, <http://www.rfc-editor.org/info/rfc4234>.

[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, <http://www.rfc-editor.org/info/rfc4474>.

[RFC4501] Josefsson, S., "Domain Name System Uniform Resource Identifiers", RFC 4501, DOI 10.17487/RFC4501, May 2006, <http://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, <http://www.rfc-editor.org/info/rfc4916>.

[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, <http://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, <http://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, <http://www.rfc-editor.org/info/rfc6973>.

[RFC7159] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", RFC 7159, DOI 10.17487/RFC7159, March 2014, <http://www.rfc-editor.org/info/rfc7159>.

[RFC7258] Farrell, S. and H. Tschofenig, "Pervasive Monitoring Is an Attack", BCP 188, RFC 7258, DOI 10.17487/RFC7258, May 2014, <http://www.rfc-editor.org/info/rfc7258>.

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

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

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

Authors' Addresses

Jon Peterson
Neustar, Inc.
1800 Sutter St Suite 570
Concord, CA 94520
US

Email: jon.peterson@neustar.biz

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

Email: fluffy@iii.ca

Eric Rescorla
RTFM, Inc.
2064 Edgewood Drive
Palo Alto, CA 94303
USA

Email: ekr@rtfm.com

Chris Wendt
Comcast
One Comcast Center
Philadelphia, PA 19103
USA

Email: chris-ietf@chriswendt.net