Enhancements for Authenticated Identity Management in the Session
Initiation Protocol (SIP)
draft-ietf-sip-identity-02

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026.

Internet-Drafts are draft documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on November 15, 2004.

Copyright Notice

Copyright (C) The Internet Society (2004). All Rights Reserved.

Abstract

The existing security mechanisms in the Session Initiation Protocol are inadequate for cryptographically assuring the identity of the end users that originate SIP requests and responses, especially in an interdomain context. This document recommends practices and conventions for identifying end users in SIP messages, and proposes a way to distribute cryptographically secure authenticated identities.
Table of Contents

1. Introduction ................................................. 3
2. Terminology ................................................ 3
3. Background .................................................. 3
4. Requirements ................................................. 5
5. Overview of Operations ................................. 6
6. User Agent Behavior: Sending Messages ............... 7
7. Authentication Service Behavior .................... 7
7.1 UAs acting as an Authentication service ........... 9
8. Verifying Identity .......................................... 9
9. Proxy Server Behavior ....................................... 10
10. Header Syntax ................................................ 12
11. Security Considerations .............................. 13
12. IANA Considerations ....................................... 16
   Authors’ Addresses ........................................ 17
A. Acknowledgments ............................................ 17
   Normative References ..................................... 16
   Informative References .................................. 16
B. Changelog ..................................................... 17
   Full Copyright Statement .................................. 19
1. Introduction

This document provides enhancements to the existing mechanisms for authenticated identity management in the Session Initiation Protocol (SIP [1]). An identity, for the purposes of this document, is defined as a canonical SIP address-of-record URI employed to reach a user (such as ‘sip:alice@atlanta.com’).

RFC3261 enumerates a number of places within a SIP request that a user can express an identity for themselves, notably 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 accurately, in the absence of some sort of cryptographic authentication mechanism.

RFC3261 specifies a number of security mechanisms that can be employed by SIP UAs, including Digest, 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 TLS or S/MIME, and furthermore Digest authentication is limited by the fact that the originator and destination must share a pre-arranged secret. It is desirable for SIP user agents to be able to send requests to destinations with they have no previous association - just as in the telephone network today, one can receive a call from someone with whom one has no previous association, and still have a reasonable assurance that their displayed Caller-ID is accurate.

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 RFC2119 [2] and indicate requirement levels for compliant SIP implementations.

3. Background

All RFC3261-compliant SIP user agents support a means of authenticating themselves to a SIP registrar – commonly with a shared secret (Digest authentication, which MUST be supported by SIP user agents, is typically used for this purpose). Registration allows a user agent to express that it is the proper entity to which requests should be sent for a particular address-of-record SIP URI.

Coincidentally, the address-of-record URI of a SIP user is also the URI with which a user commonly populates the From header of requests – in other words, the address-of-record is an identity. So in this
context, users already have a means of providing their identity, which makes good sense: since the contents of a From header field are essentially a 'return address' for SIP requests, being able to prove that you are eligible to receive requests for that 'return address' should be identical to proving that you are authorized to assert this identity.

However, the credentials with which a user agent proves their identity to a registrar cannot be validated by a user agent or proxy server outside your local domain - these credentials are currently only useful for registration. For the purposes of determining whether or not the 'return address' of a request can legitimately be asserted as the identity of the user, SIP entities in other domains require an assurance that the sender of a message is capable of authenticating themselves to a registrar in their own domain.

Ideally, then, SIP user agents should have some way of proving to recipients of SIP messages that their local domain has authenticated them. In the absence of end-user certificates in user agents, it is possible to implement a mediated authentication architecture for SIP in which requests are sent to a server in the user’s local domain which authenticates such requests (using the same practices by which the domain would authenticate REGISTER requests). Once a message has been authenticated, the local domain then needs some way to communicate to other SIP entities the sending user has been authenticated. This draft addresses how that imprimatur of authentication can be shared.

RFC3261 already describes an architecture very similar to this in Section 26.3.2.2, in which a user agent authenticates itself to a local proxy server which in turn authenticates itself to a remote proxy server via mutual TLS, creating a two-link chain of transitive authentication between the originator and the remote domain. While this works well in some architectures, there are a few respects in which this is impractical. For one, transitive trust in inherently weaker than an assertion that can be validated end-to-end. It is possible for SIP requests to cross multiple intermediaries in separate administrative domains, in which case transitive trust becomes even less compelling. It also requires intermediaries to act as proxies, rather than redirecting requests to their destinations (redirection lightens loads on SIP intermediaries).

One solution to this problem is to use ‘trusted’ SIP intermediaries that assert an identity for users in the form of a privileged SIP header. A mechanism for doing so (with the P-Asserted-Identity header) is given in [6]. However, this solution allows only hop-by-hop trust between intermediaries, not end-to-end cryptographic authentication, and it assumes a managed network of nodes with strict
mutual trust relationships, an assumption that is incompatible with widespread Internet deployment.

Accordingly, a new tactic is required for sharing a cryptographic assurance of end-user identity in an intradomain context. Furthermore, this new mechanism must work for both SIP requests and responses. However, there is an additional wrinkle specific to providing identity in a response. While the original address-of-record to which a request is sent is stored in the To header field of the request, it is possible, due to retargeting at intermediaries, it is possible that the request will be forwarded to an entity that has a different AoR (i.e. identity). Since the To header is not changed in responses to a SIP request, the UAC has no way of discovering that new AoR. This is generally known as the "response identity" or "connected party" problem.

4. Requirements

This draft addresses the following requirements:

The mechanism must allow a UAC to provide a strong cryptographic identity assurance to the UAS in a request.

The mechanism must allow a UAS to provide a strong cryptographic identity assurance to the UAC in a response.

User agents that receive identity assurances must be able to validate these assurances without performing any network lookup.

Proxy servers must be capable of adding this identity assurance to requests or responses.

The mechanism must prevent replay of the identity assurance by an attacker.

The mechanism must be capable of protecting the integrity of SIP message bodies (to ensure that media offers and answers are linked to the signaling identity).

It must be possible for a user to have multiple AoRs (i.e. accounts or aliases) under which it is known at a domain, and for the UAC to assert one identity while authenticating itself as another, related, identity, as permitted by the local policy of the domain.

It must be possible, in cases where a request has been retargeted to a different AoR than the one designated in the To header field, for the UAC to ascertain the AoR to which the request has been
5. Overview of Operations

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

Imagine the case where Alice, who has the home proxy of example.com and the address-of-record sip:alice@example.com, wants to communicate with sip:bob@example.org.

Alice generates an INVITE and places her identity in the From header field of the request. She then sends an INVITE over TLS to an authentication service proxy for her domain.

The authentication service authenticates Alice (possibly by sending a Digest authentication challenge) and validates that she is authorized to populate the value of the From header field (which may be Alice’s AoR, or it may be some other value that the policy of the proxy server permits her to use). It then computes a hash over some particular headers, including the From header field and the bodies in the message. This hash is signed with the certificate for the domain (example.com, in Alice’s case) and inserted in a header field (the new Identity header) in the SIP message. The proxy, as the holder of the private key of its domain, is asserting that the originator of this request has been authenticated and that she is authorized to claim the identity (the SIP address-of-record) which appears in the From header field. The proxy also inserts a companion header field that tells Bob how to acquire its certificate, if he doesn’t already have it.

When Bob returns a response to the INVITE (such as a 200 OK), a similar set of steps happen. Bob’s home proxy asserts his identity in the response. In this instance, the proxy has to insert the header directly into the request – redirection of responses is not possible. When Alice receives the response, she verifies Bob’s identity.

If Alice’s request for Bob were retargeted, one of two things might have happened. If it were retargeted to a domain that was also the responsibility of Bob’s home proxy (for example, retargeted from sip:bob@example.com to sip:carol@example.com), then the request would proceed normally and receive an Identity. If Bob’s home proxy would retarget the request to some other domain (e.g., sip:bob@example.ORG), then his home proxy would redirect the request rather than proxying it, and Alice would send a new request that could receive a response with an Identity from the new domain.
6. User Agent Behavior: Sending Messages

This mechanism requires one important change to existing user agent behavior for sending requests and responses: user agents using this mechanism to send requests or responses MUST support TLS; moreover, they MUST be capable of establishing a persistent TLS connection with a proxy server that acts as an authentication service. Additionally, there are several practices that should be highlighted in the context of this identity solution.

When a UAC sends a request, it MUST accurately populate the header field that asserts its identity (for a SIP request, this is the From header field). In a request it MUST set the URI portion of its From header to match a SIP, SIPS or TEL URI AoR under which the UAC can register (including anonymous URIs, as described in RFC 3323). The UAC MUST also be capable of sending requests through an ‘outbound’ proxy (the authentication service), and of course MUST support the Digest authentication mechanism described in RFC3261. Because this mechanism does not provide integrity protection for the first hop to the authentication service, the UAC MUST send requests to an authentication service only over a TLS connection. Additionally, in order to provide identity for responses, user agents MUST form a persistent TLS connection to a proxy server when a REGISTER is sent.

Since a UAS cannot send a response that does not replicate the contents of the To and From header fields in the corresponding request, UAS response-sending behavior is unchanged. Again, because this mechanism does not provide integrity protection for the first hop of the response path, the UAS SHOULD send responses only over a TLS connection.

7. Authentication Service Behavior

The authentication service authenticates the identity of the message sender and validates that the identity given in the message can legitimately be asserted by the sender. Then it computes a signature over the canonical form of several headers and all the bodies, and inserts this signature into the message.

First, an authentication service MUST extract the identity of the sender. For requests, it inspects the From header field; for responses, the To header field (henceforth the result of this inspection will be referred to as the "identity field"). If the identity field contains a SIP or SIPS URI, the authentication service MUST extract the hostname portion of the URI in that header field, and compare this to the domain(s) for which it is responsible. If the identity field uses the TEL URI scheme, the policy of the
authentication service determines whether or not it is responsible for this identity. Some example policies are described in [TODO]. If the authentication service is not responsible for the identity in question, it MAY handle the request as a normal proxy server; see below for more information on authentication service handling of an existing Identity header.

Second, the authentication service must 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:

For requests, challenging the request with a 407 response code using the Digest authentication scheme (or viewing a Proxy-Authentication header sent in the request which was sent in anticipation of a challenge using cached credentials, as described in RFC 3261 Section 22.3)

For requests and responses that are sent over a persistent TLS connection, relying on some prior authentication that was performed at the formation of the connection (most likely, the authentication service previously challenged a REGISTER request sent after the TLS connection was formed, or possibly a prior challenged INVITE that was sent over the TLS connection)

Authorization of the assertion of a particular username in the From header field of a SIP message is a matter of local policy for the authorization service which depends greatly on the manner in which authentication is performed. A RECOMMENDED policy is as follows: the username asserted during Digest authentication MUST correspond exactly to the username in the From header field of the SIP message. However, there are many cases in which a user might manage multiple accounts in the same administrative domain. Accordingly, provided the authentication service is aware of the relationships between these accounts, it might allow a user providing credentials for one account to assert a username associated with another account controlled by the user name. Furthermore, if the AoR asserted in the From header field is anonymous (per RFC3323), then the proxy should authenticate that the user is any valid user in the domain and insert the signature over the From header field as usual.

Third, the authentication service MUST form the identity signature, as described in Section 10, and add an Identity header to the request containing this signature. After the Identity header has been added to the request, the authentication service MUST also add an Identity-Info header. The Identity-Info header contains a URI from which its certificate can be acquired. Details are provided in section Section
10.

Finally, the authentication service MUST forward the message normally.

7.1 UAs acting as an Authentication service

There are some instances in which a user agent may hold the private key of the domain Certificate for its address-of-record. In these cases, the UA MAY perform the services, and add the headers, that the authentication service would normally add.

8. Verifying Identity

When a user agent or proxy server receives a SIP message containing an Identity header, it can inspect the signature to verify the identity of the sender of the message. If an Identity header is not present in a request, and one is required by local policy, then a 428 Use Identity response is sent. If an Identity header is not present in a response, and one is required by local policy, then the recipient of the response MUST communicate this lapse to its user, and MAY immediately terminate any created dialog or ignore transactions, as policy dictates.

In order to verify the identity of the sender of a message, the user agent or proxy server MUST first acquire the certificate for the signing domain. Implementations supporting this specification should have some means of retaining domain certificates (in accordance with normal practices for certificate lifetimes and revocation) in order to prevent themselves from needlessly downloading the same certificate every time a request from the same domain is received. Certificates retained in this manner should be indexed by the URI given in the Identity-Info header field value.

Provided that the domain certificate used to sign this message is unknown, SIP entities discover this certificate by dereferencing the Identity-Info header. The client processes this certificate in the usual ways including checking that it has not expired, that the chain is valid back to a trusted CA, and that it does not appear on revocation lists.

Subsequently, the recipient MUST verify the signature in the Identity header, and compare the identity of the signer (the subjectAltName of the certificate) with the domain portion of the URI in the From header field of the request as described in Section 11. Additionally, the Date, Contact and Call-ID headers MUST be analyzed in the manner described in Section 11; recipients that wish to verify Identity signatures MUST support all of the operations described
there. Any discrepancies or violations MUST be reported to the user.

When the originating user agent of a request receives a response containing an Authenticated Identity Body (AIB, see [4]), it SHOULD compare the identity in the From header field of the AIB of the response with the original value of the To header field in the request. If these represent different identities, the user agent SHOULD render the identity in the AIB of the response to its user. Note that a discrepancy in these identity fields is not necessary an indication of a security breach; normal retargeting may simply have directed the request to a different final destination. Implementers therefore may consider it unnecessary to alert the user of a security violation in this case.

9. Proxy Server Behavior

In most respects, a proxy server behaves normally when it receives a SIP request or response containing an Identity header. This mechanism is fully backwards-compatible with existing RFC3261 proxy behavior. However, if a proxy intends to act as an authentication service for responses to requests it receives, it must exhibit some additional behavior to ensure that retargeting requests are handled properly. Essentially, a proxy server MUST NOT provide an Identity header for a request that it retargets to a different administrative domain. It is the responsibility of that administrative domain to provide its own identity assertion, if it can. However, proxying the request to a remote domain where identity services may be provided has its own problems - the originator of the request has no way to know whether the request was legitimately retargeted, or if any responses it receives from the new domain are spoofed or otherwise illegitimate. It is thus much more secure for the proxy server to redirect in cases where it might otherwise retarget.

If a proxy server intends to act as an authentication service for a response to a SIP request that it is forwarding, it MUST do ALL of the following:

Ascertain if it is responsible for the domain indicated in the Request-URI field of the request. If not, it MUST forward the request normally. If so, it must then:

Determine the route set of targets to which this request might be forwarded. From that target list, the proxy must determine which contact addresses are associated with persistent TLS connections that have been established to the proxy server. It places all such targets (if any) into a primary route set for the call, and places remaining targets into a secondary route set for the call. It performs this operations irrespective of any qvalues associated
with the contact addresses.

The proxy then MUST follow normal administrative policies for forwarding the request to any targets in the primary route set (which may involve qvalue calculations or any other behaviors described in RFC3261). Before the proxy forwards any responses to this request upstream, the proxy server MUST act as an authentication service (as described in Section 7), adding an Identity and Identity-Info header.

If there are no appropriate responses from the primary route set for the proxy server to forward upstream, it moves on to the secondary route set (essentially, the proxy server forks sequentially, exploring the primary route set as one cluster, and then moves on to the secondary set). The proxy server is unable to act as an authentication service for those contact addresses. Accordingly, the proxy server MUST NOT explore these route targets itself; instead, it MUST redirect the request with a 3xx class response containing the contact addresses that constitute the secondary route set.

In order to build the primary route set for the call, the location service associated with the domain of the proxy server MUST implement additional intelligence to determine which contact addresses are associated with a persistent TLS connection - this is used to determine when the server should act as a proxy and when it should redirect. When the SIP registrar receives a REGISTER request over a persistent TLS connection, it MUST compare any contact addresses appearing in Contact header fields to the topmost Via header field in the REGISTER request. If the host portion of a contact address matches the hostname given in the topmost Via header field, then that contact address is said to be "associated" with the persistent TLS connection over which the REGISTER was received. Location services must mark or flag these contact addresses accordingly, and remember the identity that the user provided when they were authenticated during registration. Only these contact addresses are added to the primary route set by a proxy server that wishes to act as an authentication service for responses.

Additionally, domain policy may require proxy servers to inspect and verify the identity provided in SIP requests. The proxy server may wish to ascertain the identity of the sender of the message to provide spam prevention or call control services. Even if a proxy server does not act as an authentication service, it MAY verify the signature present in an Identity header before it makes a forwarding decision for a request. Proxy servers MUST NOT remove or modify the Identity or Identity-Info headers.
10. Header Syntax

This document specifies two new SIP headers: Identity and Identity-Info. Each of these headers can appear only once in a SIP message.

Identity = "Identity" HCOLON signed-identity-digest
signed-identity-digest = LDQUOT 32LHEX RDQUOT

Identity-Info = "Identity-Info" HCOLON ident-info
Ident-info = LAQUOT absoluteURI RAQUOT

To create the contents of the signed-identity-digest, the following elements of a SIP message MUST placed in the string in the order specified here, separated by a colon:

The AoR of the UA sending the message, or the ‘identity field’. For a request, this is the addr-spec from the From header field; for responses, the addr-spec of the To header field. This needs to be in lower case and to be represented as a SIP URI.

The callid from Call-Id header field.

The Date header field, with exactly one space each for each SP and the weekday and month items case set as shown in BNF in 3261. The first letter is upper case and the rest of the letters are lower case.

The addr-spec component of the Contact header field value.

The body content of the message with the bits exactly as they are in the message.

For more information on the security properties of these headers, and why their inclusion mitigates replay attacks, see [4]. The precise formulation of this digest-string is, therefore:

digest-string = addr-spec HCOLON callid HCOLON SIP-Date HCOLON addr-spec HCOLON message-body

Note again that the first addr-spec MUST be taken from the From header field value, and the second addr-spec from the Contact header field value.

After the digest-string is formed, it MUST be hashed and signed with the certificate for the domain, as follows: compute the results of signing this string with sha1WithRSAEncryption as described in RFC 3370 and base64 encode the results as specified in RFC 3548. Put the result in the Identity header.
Note on this choice: Assuming a 1024 bit RSA key, the raw signature will result in about 170 octets of base64 encoded data. For comparison’s sake, a typical HTTP Digest Authorization header (such as those used in RFC3261) with no cnonce is around 180 octets. From a speed point of view, a 2.8GHz Intel processor does somewhere in the range of 250 RSA 1024 bits signs per second or 1200 RSA 512 bits signs; verifies are roughly 10 times faster. Hardware accelerator cards are available that speed this up.

The Identity-Info header MUST contain either an HTTPS URI or a SIPS URI. If it contains an HTTPS URI, the URI must dereference to a resource that contains a single MIME body containing the certificate of the authentication service. If it is a SIPS URI, then the authentication service intends for a user agent that wishes to fetch the certificate to form a TLS connection to that URI, acquire the certificate during normal TLS negotiation, and close the connection.

This document adds the following entries to Table 2 of [1]:

<table>
<thead>
<tr>
<th>Header field</th>
<th>where</th>
<th>proxy</th>
<th>ACK</th>
<th>BYE</th>
<th>CAN</th>
<th>INV</th>
<th>OPT</th>
<th>REG</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identity</td>
<td>a</td>
<td></td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Identity-Info</td>
<td>a</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

11. Security Considerations

This document describes a mechanism which provides a signature over the Contact, Date, Call-ID, and ‘identity fields’ (addr-spec of the From header field for requests, and To header field for responses) of SIP messages. While a signature over the identity field alone would be sufficient to secure a URI alone, the additional headers provide replay protection and reference integrity necessary to make sure that the Identity header will not be used in cut-and-paste attacks. 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 in particular).
The identity field indicates the identity of the sender of the message. The Date and Contact headers provide reference integrity and replay protection, as described in RFC3261 Section 23.4.2. Implementations of this specification MUST NOT consider valid a request with an outdated Date header field (the RECOMMENDED interval is that the Date header must indicate a time within 3600 seconds of the receipt of a message). Implementations MUST also record Call-IDs received in valid requests containing an Identity header, and MUST remember those Call-IDs for at least the duration of a single Date interval (i.e. 3600 seconds). Accordingly, if an Identity header is replayed within the Date interval, receivers will recognize that it is invalid because of a Call-ID duplication; if an Identity header is replayed after the Date interval, receivers will recognize that it is invalid because the Date is stale. The Contact header field is included to tie the Identity header to a particular device instance that generated the request. Were an active attacker to intercept a request containing an Identity header, and cut-and-paste the Identity header field into their own request (reusing the identity fields, Contact, Date and Call-ID fields that appear in the original message), they would not be eligible to receive SIP requests from the called user agent, since those requests are routed to the URI identified in the Contact header field.

This mechanism also provides a signature over the bodies of SIP requests. The most important reason for doing so is to protect SDP bodies carried in SIP requests. There is little purpose in establishing the identity of the user agent that provided the signaling if a man-in-the-middle can change the SDP and direct media to an alternate address. Note however that this is not perfect end-to-end security. The authentication service itself, when instantiated at a intermediary, can change the SDP (and SIP headers, for that matter) before providing a signature. Thus, while this mechanism reduces the chance that a man-in-the-middle will interfere with sessions, it does not eliminate it entirely. Since it is assumed that the user trusts their local domain to vouch for their security, they must also trust the service not to violate the integrity of their message bodies without good reason.

Users SHOULD NOT provide credentials to an authentication service to which they cannot initiate a direct connection, preferably one secured by TLS. If a user does not receive a certificate from the authentication service over this TLS that corresponds to the expected domain (especially when they receive a challenge via a mechanism such as Digest), then it is possible that a rogue server is attempting to pose as a authentication service for a domain that it does not control, possibly in an attempt to collect shared secrets for that domain. If a user cannot connect directly to the desired authentication service, the user SHOULD at least use a SIPS URI to
ensure that mutual TLS authentication will be used to reach the remote server.

Relying on an Identity header generated by a remote administrative domain assumes that the issuing domain uses some trustworthy practice to authenticate its users. However, it is possible that some domains will implement policies that effectively make users unaccountable (such as accepting unauthenticated registrations from arbitrary users). The value of an Identity header for such domains is questionable.

Since a domain certificate is used by an authentication service (rather than individual certificates for each identity), certain problems can arise with name subordination. For example, if an authentication service holds a common certificate for the hostname ‘sip.atlanta.com’, can it legitimately sign a token containing an identity of ‘sip:alice@atlanta.com’? It is difficult for the recipient of a request to ascertain whether or not ‘sip.atlanta.com’ is authoritative for the ‘atlanta.com’ domain unless the recipient has some foreknowledge of the administration of ‘atlanta.com’. Therefore, it is RECOMMENDED that user agent recipients of authentication tokens notify end users if there is ANY discrepancy between the subjectAltName of the signers certificate and the identity within the authentication token. Minor discrepancies MAY be characterized as such. Additionally, relying parties MAY follow the procedures in RFC3264 to look up on the domain portion of the identity in the From header field in the DNS, and compare the SIP services listed for that domain with the subjectAltName of the certificate; this can give the relying party a better sense of the canonical SIP services for that domain.

Because the domain certificates that can be used by authentication services need to assert only the hostname of the authentication service, existing certificate authorities can provide adequate certificates for this mechanism. However, not all proxy servers and user agents will be able support the root certificates of all certificate authorities, and moreover there are some significant differences in the policies by which certificate authorities issue their certificates. This document makes no recommendations for the usage of particular certificate authorities, nor does it describe any particular policies that certificate authorities should follow, but it is anticipated that operational experience will create de facto standards for the purposes of authentication services. Some federations of service providers, for example, might only trust certificates that have been provided by a certificate authority operated by the federation.
12. IANA Considerations

This document specifies two new SIP headers: Identity and Identity-Info. Their syntax is given in Section 10. This document requests that IANA add these headers to the SIP header registry.

This document also defines a new SIP response code, 428 "Use Identity", as described in Section 8.

Normative References


Informative References


Appendix A. Acknowledgments

The authors would like to thank Eric Rescorla, Rohan Mahy, Robert Sparks, Jonathan Rosenberg, Mark Watson and Patrik Faltstrom for their comments.

Appendix B. Changelog

Changes from draft-ietf-sip-identity-01:

- Completely changed underlying mechanism - instead of using an AIB, the mechanism now recommends the use of the Identity header and Identity-Info header
- Numerous other changes resulting from the above
- Various other editorial corrections

Changes from draft-peterson-sip-identity-01:

- Split off child draft-ietf-sip-authid-body-00 for defining of the AIB
- Clarified scope in introduction
- Removed a lot of text that was redundant with RFC3261 (especially about authentication practices)

- Added mention of content indirection mechanism for adding token to requests and responses

- Improved Security Considerations (added piece about credential strength)

Changes from draft-peterson-sip-identity-00:

- Added a section on authenticated identities in responses

- Removed hostname convention for authentication services

- Added text about using ‘message/sip’ or ‘message/sipfrag’ in authenticated identity bodies, also RECOMMENDED a few more headers in sipfrags to increase reference integrity

- Various other editorial corrections
Full Copyright Statement

Copyright (C) The Internet Society (2004). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.