

SIP WG
Internet-Draft
Expires: March 30, 2005

J. Peterson
NeuStar
C. Jennings
Cisco Systems
September 29, 2004

Enhancements for Authenticated Identity Management in the Session
Initiation Protocol (SIP)
draft-ietf-sip-identity-03

Status of this Memo

By submitting this Internet-Draft, I certify that any applicable patent or other IPR claims of which I am aware have been disclosed, and any of which I become aware will be disclosed, in accordance with [RFC 3668](#).

Internet-Drafts are working 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/lid-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 March 30, 2005.

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, 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.

Internet-Draft

SIP Identity

September 2004

Table of Contents

1.	Introduction	3
2.	Terminology	3
3.	Background	3
4.	Requirements	5
5.	Overview of Operations	6
6.	Authentication Service Behavior	7
7.	Verifying Identity	9
8.	User Agent Behavior	10
9.	Proxy Server Behavior	10
10.	Header Syntax	11
11.	Compliance Tests and Examples	13
11.1	Identity-Info with a Singlepart MIME body	14
11.2	Identity for a Request with no MIME body or Contact	16
12.	Identity and the TEL URI Scheme	19
13.	Privacy Considerations	20
14.	Security Considerations	21
15.	IANA Considerations	25
15.1	Header Field Names	25
15.2	Response Code	25
	Authors' Addresses	26
A.	Acknowledgments	26
B.	Bit-exact archive of example messages	27
B.1	Encoded Reference Files	27
16.	References	25
16.1	Normative References	25
16.2	Informative References	26
C.	Changelog	30
	Intellectual Property and Copyright Statements	32

Internet-Draft

SIP Identity

September 2004

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.example.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 (e.g., 'sip:alice@atlanta.example.com').

The address-of-record URI used for registration is also the URI with which a UA commonly populates the From header of requests in order to

provide their 'return address' identity to recipients. If you can prove you are eligible to register in a domain under a particular address-of-record, you are proving that you are capable of legitimately receiving requests for that address-of-record, and accordingly, when you place that address-of-record in the From header field of a SIP request other than a registration (like an INVITE), you are providing a 'return address' where you can legitimately be reached. In other words, if you are authorized to receive requests for that 'return address', you are also authorized to assert that 'return address' in your From header field.

In the context of registration, users already have a means of proving their identity to a registrar. However, the credentials with which a user agent proves their identity to a registrar cannot be validated by just any user agent or proxy server - these credentials are only shared between the user agent and their domain administrator. For the purposes of determining whether or not the 'return address' of a request can legitimately be asserted in the From header field of a request, SIP entities that are not operated by the domain administrator 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 requests 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 that 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 is 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 [\[8\]](#). 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, this document specifies a means of sharing a cryptographic assurance of end-user SIP identity in an interdomain context based on the concept of an 'authentication service' and a new SIP header, the Identity header. Note that the scope of this document is limited to providing this identity assurance for SIP requests; solving this problem for SIP responses is more complicated, and is a subject for future work.

This specification allows either a user agent or a proxy server to act as an authentication service. To maximize end-to-end security, it is obviously preferable for end users to hold their own certificates; if they do, they can act as an authentication service.

However, end-user certificates may be neither practical nor affordable, given the difficulties of establishing a PKI that extends to end users, and moreover, 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 certificates across multiple devices may be very complex, and requires quite a good deal of additional endpoint behavior. Managing several certificates for the various devices is also quite problematic and unpopular with users. Accordingly, in the initial use of this mechanism, it is likely that intermediaries will instantiate the authentication service role.

4. Requirements

This draft addresses the following requirements:

- o The mechanism must allow a UAC to provide a strong cryptographic identity assurance in a request that can be verified by a proxy server or UAS.
- o User agents that receive identity assurances must be able to validate these assurances without performing any network lookup.
- o User agents that hold certificates on behalf of their user must be capable of adding this identity assurance to requests.
- o Proxy servers that hold certificates on behalf of their domain must be capable of adding this identity assurance to requests; a UAC is not required to support the Identity header in order for identity to be added to a request in this fashion.

- o The mechanism must prevent replay of the identity assurance by an attacker.
- o 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).
- o 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.
- o 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 sent.

[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 new header field in the SIP message, the 'Identity' header.

The proxy, as the holder 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's domain receives the request, it verifies the signature provided in the Identity header, and thus can authenticate that the domain indicated by the host portion of the AoR in the From header

field authenticated the user, and permitted them to assert that From header field value.

[6.](#) Authentication Service Behavior

This document defines a new role for SIP entities called an authentication service. The authentication service role can be

instantiated by a proxy server, redirect server or a user agent. Any entity that instantiates the authentication service role MUST possess the private key of a domain certificate, and MUST be capable of authenticating one or more SIP users that can register in that domain. Commonly, this role will be instantiated by a proxy server or redirect server, since these entities are more likely to have a static hostname, hold a corresponding certificate, and access to SIP registrar capabilities that allow them to authenticate users in their domain.

SIP entities that act as an authentication service MUST add a Date header field to SIP requests if one is not already present. Similarly, authentication services MUST add a Content-Length header field to SIP requests if one is not already present; this can help the verifier to double-check that they are hashing exactly as many bytes of message-body as the authentication service when they verify the message.

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 from the request. The authentication service takes this value from the From header field; this AoR will be referred to here as the 'identity field'. If the identity field contains a SIP or SIPS URI, the authentication service MUST extract the hostname portion of the identity field and compare it 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; see [Section 12](#) for more information. If the authentication service is not responsible for the identity in question, it MAY handle the request normally, but it MUST NOT add an Identity header; see below for more information on authentication service handling of an existing Identity header.

Second, the authentication service needs to 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:

- o If the authentication service is instantiated by a SIP intermediary (proxy or redirect server), it may challenge 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](#)).
- o If the authentication service is instantiated by a SIP user agent, a user agent can be said to authenticate its user on the grounds that the user can provision the user agent with the private key of the domain, or by preferably by providing a password that unlocks said private key.

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 \[3\]](#)), then the proxy should authenticate that the user is a valid user in the domain and insert the signature over the From header field as usual.

Note that this check is performed on the addr-spec in the From header 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'). Some SIP user agents that receive requests render the display-name of the caller as the identity of the caller. However, there are many environments in which legislating the display-name isn't feasible, judging from experience with email, where users frequent make slight textual changes to their display-names. Ultimately, there is more value in focusing on the SIP address of the sender (which has some meaning in the network and provides a chain of accountability) than trying to constrain how the display-name is set. As such, authentication services MAY check the display-name as well, 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 MUST return a 403 'Inappropriate Display-Name' response code. However, in many environments this will not make sense. For more information on rendering identity in a user

Internet-Draft

SIP Identity

September 2004

interface, see [Section 8](#).

Third, the authentication service MUST form the identity signature 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](#). 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 (for example, based on a global policy, a per-sending-domain policy, or a per-sending-user policy), then a 428 'Use Identity Header' response MUST be sent.

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 not previously known to the recipient, SIP entities SHOULD discover this certificate by dereferencing the Identity-Info header, unless they have some more efficient implementation-specific way of acquiring certificates for that domain. 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. Once the certificate is acquired, it MUST be validated.

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 14](#).

Additionally, the Date, Contact and Call-ID headers MUST be analyzed in the manner described in [Section 14](#); 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.

If a verifier determines that the signature on the message does not correspond to the text of the message, then a 428 'Invalid Identity Header' response MUST be returned.

Once the identity of the sender of a request has been ascertained, various policies MAY be used to make authorization decisions about accepting communications and the like. Such policies are outside the scope of this document.

[8](#). User Agent Behavior

This mechanism requires one important change to existing user agent requirements for sending requests: user agents using this mechanism to send requests to an authentication service MUST support TLS. 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.

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](#) [3]). In general, UACs SHOULD NOT use the TEL URI form in the From header field (see [Section 12](#)).

The UAC MUST also be capable of sending requests, including mid-call requests, through an 'outbound' proxy (the authentication service). The best way to accomplish this is using pre-loaded Route headers and loose routing. UAC implementations MUST provide a way of provisioning pre-loaded Route headers in order for this mechanism to work for mid-call requests in the backwards direction of a dialog.

As a recipient of a request, a user agent that can verify signed identities should also support an appropriate user interface to render the validity of identity to a user. User agent implementations SHOULD differentiate signed From header field values from unsigned From header field values when rendering to an end user the identity of the sender of a request.

[9.](#) Proxy Server Behavior

Domain policy may require proxy servers to inspect and verify the identity provided in SIP requests. A 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 existence of an Identity before it makes a forwarding decision for a request. Proxy servers MUST NOT remove or modify an existing Identity or Identity-Info header in a request.

[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 = LDQUOTE 32LHEX RDQUOTE

Identity-Info = "Identity-Info" HCOLON ident-info
ident-info = LAQUOTE absoluteURI RAQUOTE

The signed-identity-digest is a signed hash of a canonical string generated from certain components of a SIP request. To create the contents of the signed-identity-digest, the following elements of a SIP message MUST be placed in a bit-exact string in the order specified here, separated by a colon:

- o 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.
- o The addr-spec component of the To header field, which is the AoR to which the request is being sent.
- o The callid from Call-Id header field.
- o The digit (1xDIGIT) and method (method) portions from CSeq header

field, separated by a single space (ABNF SP, or %x20). Note that the CSeq header field allows LWS rather than SP to separate the digit and method portions, and thus the CSeq header field may need to be transformed in order to be canonicalized.

- o 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. All requests that use the Identity mechanism MUST contain a Date header.
- o The addr-spec component of the Contact header field value. If the request does not contain a Contact header, this field MUST be empty (i.e., there will be no whitespace between the fourth and fifth colons in the canonical string).
- o The body content of the message with the bits exactly as they are in the Message (in the ABNF for SIP, the message-body). Note that the message-body does NOT include the CRLF separating the SIP headers from the message-body, but does include everything that follows that CRLF. If the message has no body, then message-body will be empty, and the final colon will not be followed by any

additional characters.

For more information on the security properties of these headers, and why their inclusion mitigates replay attacks, see [Section 14](#) and [5]. The precise formulation of this digest-string is, therefore (following the ABNF [6] in [RFC3261](#)):

```
digest-string = addr-spec ":" addr-spec ":" callid ":" 1*DIGIT SP method ":"  
                SIP-Date ":" [ addr-spec ] ":" message-body
```

Note again that the first addr-spec MUST be taken from the From header field value, the second addr-spec MUST be taken from the To header field value, and the third addr-spec MUST be taken from the Contact header field value, provided the Contact header is present in the request.

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 (without base64, as an aside, it would be about 130 bytes). 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 [\[1\]](#):

Header field	where	proxy	ACK	BYE	CAN	INV	OPT	REG
-----	-----	-----	---	---	---	---	---	---
Identity	R	a	o	o	-	o	o	-
			SUB	NOT	REF	INF	UPD	PRA
			---	---	---	---	---	---

			o	o	o	o	o	o
Header field	where	proxy	ACK	BYE	CAN	INV	OPT	REG
-----	-----	-----	---	---	---	---	---	---
Identity-Info	R	a	o	o	-	o	o	-
			SUB	NOT	REF	INF	UPD	PRA
			---	---	---	---	---	---
			o	o	o	o	o	o

Note, in the table above, that this mechanism does not protect the REGISTER method or the CANCEL method. The CANCEL method cannot be challenged, because it is hop-by-hop, and accordingly authentication service behavior for CANCEL would be significantly limited. The

REGISTER method uses Contact header fields in very unusual ways that complicate its applicability to this mechanism. Accordingly, the Identity and Identity-Info header MUST NOT appear in REGISTER or CANCEL.

11. Compliance Tests and Examples

The examples in this section illustrate the use of the Identity header in the context of a SIP transaction. Implementations MUST verify their compliance with these examples, i.e.:

- o Implementations of the authentication service role MUST generate identical base64 identity strings to the ones shown in the Identity headers in these examples when presented with the source message and utilizing the appropriate supplied private key for the domain in question.
- o Implementations of the verifier role MUST correctly validate the given messages containing the Identity header when utilizing the supplied certificates (with the caveat about self-signed certificates below).

Note that the following examples use self-signed certificates, rather than certificates issued by a recognized certificate authority. The use of self-signed certificates for this mechanism is NOT RECOMMENDED, and appear here only for illustrative purposes. Therefore, in compliance testing, implementations of verifiers SHOULD generated appropriate warnings about the use of self-signed certificates.

Bit-exact reference files for these messages and their various transformations are supplied in [Appendix B](#).

11.1 Identity-Info with a Singlepart MIME body

Consider the following private key and certificate pair assigned to 'atlanta.example.com'.

-----BEGIN RSA PRIVATE KEY-----

MIICXQIBAAKBgQC8HmM8b9E4WNhb7tZAoBVSkKyV9rAEX3nyQbg4hXte1oW1BxC+
43MQHrG3nk6Kc9afPR6VloKwWoUoAcCnbTJ/zEiZ6dq+C5EsQGI0owYkSgqd02po

joCnRgzgJgvAl41R2J6CE1kMw0QxNCxPnTco8l8UGdKbNLXIuNdUM1MG8QIDAQAB
AoGAAtPOGAVyNo+XS0JxE+2UBHaqMWLQyHAK7Coys57F+OnufocJqGTQwOhFMYZO
leQh0Kjhgcw0UMo7gBtuotWQUbbLHTGKXiBR6Pqbm6CvhwJSuNYv0v0NuTb1SMl1
Kadg43na4B9kQeytn1y6lfkTkK2oYqkDVZ2AAmLSLrfhl1UCQQDp7VFIItgmnybwK
PKwJs8gnF+u+K9j+sac/3vgGgr0vpxVqwoMXl6eWN//pZ/cqshanDLmtr9ahjWCD
DxYVyk1rAkEAzd6JLJAhg8cZymVCS5Jf0F7FAVxpx0BgRPHwJliYUg604jPY+ASg
cLP6nz9a38wWZQj6rRygffGZHXbBFm+8EwJBAJmZef5ESSK6+5VdMTlNqubAdjJw
aBMUY1U0+naL66AyfYWUIq+jDI8+RfLkKQ8H0IfvexvokW2SfwSPK1kzcfECQD/O
MQW2xgwt8ThhmeKCQ1/5f2WklsRCL5PGyH+aDeqQyIgj0aPlCzTjE1I3+JpUTryR
w9/Td4qRTrtrCv1BNDECQCgHIzF8LFtI003w9MAEAoCyDbtHFPEj71b+qG22Yc4
SPFBAbo3JG0+mrB0MX/GwJr+3DfgzMHauX/tinPr+u1D

-----END RSA PRIVATE KEY-----

-----BEGIN CERTIFICATE-----

MIIC/TCCAmagAwIBAgIBADANBgkqhkiG9w0BAQQFADBZMQswCQYDVQQGEwJVUzEQ
MA4GA1UECBMHR2VvcmdpYTESMBAGA1UEBxQJQXR5YXQIbnRhMQ0wCwYDVQQKEwRJ
RVRGMRUwEwYDVQQLFAxTT0lQCAgISVAgV0cwHhcNMDQwOTEzMTAxMzAzWhcNMDUw
OTEzMTAxMzAzWjBZMQswCQYDVQQGEwJVUzEQMA4GA1UECBMHR2VvcmdpYTESMBAG
A1UEBxQJQXR5YXQIbnRhMQ0wCwYDVQQKEwRJRVBGMRUwEwYDVQQLFAxTT0lQCAgI
SVAgV0cwZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBALweYzXv0ThY2Fvu1kCg
FVKQrJX2sARfefJBuDiFe17WhbUHEL7jcxAesbeeTopz1p89HpWWgrBahSgBwKdt
Mn/MSJnp2r4LkSxAYg6jBiRKCP07amiOgKdGDOCOC8CXjVHYnoITWQzA5DE0LE+d
NyjyXxQZ0ps0tci411QzUwbxAgMBAAGjgdQwgdEwHQYDVR00BBYEFgfCU7cNxxqSK
NurvFqz8gj5px8uoMIGBBGNVHSMEEjB4gBRnwL03Dcakijbq7xas/II+acFLqKFd
pFswWTELMakGA1UEBhMCMVVMxEDA0BgNVBAGTB0dlb3JnaWExEjAQBGNVBACUCUF0
bGF0CG50YTENMASGA1UEChMESUVURjEVMBMGA1UECXMU09JUAgICElQIFdHggEA
MAwGA1UdEwQFMAMBAf8wHgYDVR0RBBCwFYITYXR5YW50YS5leGFtcGxllmNvbTAN
BgkqhkiG9w0BAQQFAA0BgQA0a/5hU6yqRTxwqoBuRk/iSqDnJD/B0QQnSFLqdjy
QV/Pm+aluA05aLRDwq6w/ufwX2HPLOvXYubpnNzjpaWCx30Lr4b5NwnsfNSxtKBJ
vI9PWwhSW6Vmo/cT2llhNudCmN+LXPd/SLy3gnGvXtwcrWAT8MVYmkCUQTRvbWaR
fQ==

-----END CERTIFICATE-----

A user of atlanta.example.com, Alice, wants to send an INVITE to bob@biloxi.example.org. She therefore creates the following INVITE request, which she forwards to the atlanta.example.org proxy server that instantiates the authentication service role:

Via: SIP/2.0/TLS pc33.atlanta.example.com;branch=z9hG4bKnashds8
To: Bob <sip:bob@biloxi.example.org>
From: Alice <sip:alice@atlanta.example.com>;tag=1928301774
Call-ID: a84b4c76e66710
CSeq: 314159 INVITE
Max-Forwards: 70
Date: Thu, 21 Feb 2002 13:02:03 GMT
Contact: <sip:alice@pc33.atlanta.example.com>
Content-Type: application/sdp
Content-Length: 147

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

When the authentication service receives the INVITE, it authenticates Alice by sending a 407 response. As a result, Alice adds an Authorization header to her request, and resends to the atlanta.example.com authentication service. Now that the service is sure of Alice's identity, it calculates an Identity header for the request. The canonical string over which the identity signature will be generated is the following (note that the first line wraps because of RFC editorial conventions):

sip:alice@atlanta.example.com:sip:bob@biloxi.example.org:a84b4c76e66710:314159
o=UserA 2890844526 2890844526 IN IP4 pc33.atlanta.example.com
s=Session SDP
c=IN IP4 pc33.atlanta.example.com
t=0 0
m=audio 49172 RTP/AVP 0
a=rtpmap:0 PCMU/8000

The resulting signature (sha1WithRsaEncryption) using the private RSA key given above, with base64 encoding, is the following:

CyI4+nAkHrH3ntmaxgr01TMxTmtjP7MASwliNRdupRI1vpkXRvZXx1ja9k0nB2sN
3W+v1PDsy32MaqZi0M5WfEkXxbgTnPYW0jIoK8HMyY1VT7egt0kk4XrKFCHYWGC1
sM9CG4hq+YJZTMaSR0oMUBhikVIjnQ8ykeD6UXN0yfI=

Accordingly, the atlanta.example.com authentication service will create an Identity header containing that base64 signature string (175 bytes). It will also add an HTTPS URL where its certificate is made available. With those two headers added, the message looks

like:

```
INVITE sip:bob@biloxi.example.org SIP/2.0
Via: SIP/2.0/TLS pc33.atlanta.example.com;branch=z9hG4bKnashds8
To: Bob <sip:bob@biloxi.example.org>
From: Alice <sip:alice@atlanta.example.com>;tag=1928301774
Call-ID: a84b4c76e66710
CSeq: 314159 INVITE
Max-Forwards: 70
Date: Thu, 21 Feb 2002 13:02:03 GMT
Contact: <sip:alice@pc33.atlanta.example.com>
Identity: CyI4+nAkHrH3ntmaxgr01TMxTmtjP7MASwliNRdupRI1vpkXRvZXx1ja9k0n
        3W+v1PDsy32MaqZi0M5WfEkXxbgTnPYW0jIoK8HMyY1VT7egt0kk4XrKFCHY
        sM9CG4hq+YJZTMaSR0oMUBhikVIjnQ8ykeD6UXN0yfI=
Identity-Info: https://atlanta.example.com/cert
Content-Type: application/sdp
Content-Length: 147

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

atlanta.example.com then forwards the request normally. When Bob receives the request, if he does not already know the certificate of atlanta.example.com, he de-references the URL the Identity-Info header to acquire the certificate. Bob then generates the same canonical string given above, from the same headers of the SIP request. Using this canonical string, the signed digest in the Identity header, and the certificate discovered by de-referencing the Identity-Info header, Bob can verify that the given set of headers and the message body have not been modified.

11.2 Identity for a Request with no MIME body or Contact

Consider the following private key and certificate pair assigned to "biloxi.example.org".

Internet-Draft

SIP Identity

September 2004

-----BEGIN RSA PRIVATE KEY-----

```
MIICXQIBAAKBgQDDIREMIIS9vBBET2FFHss2Lbwri/nK+AMoUZ74UT3amG/bYgDn
H86eUUEjGfV3cfXErFXSnI86sUALoKjjwGYBoiUuaMhyerZyF+D9St2pInBeq6fq
rbaPpL6bvIAF636/02+GFP3LSLj6KS4HQwnsaUBr2YzykBD05PfwrH28VQIDAQAB
AoGAZLRJFwglWcKYZpjNK54T5HdAGP1Zwo2zG3jcYW2UTZ/EguWWb7HzsbNfuZzp
GWcgHwuOE28nYHQgCKA26avf0GuebFH2z2WLAFC3TCOVjMzJEWawtxIc7oX9vziTF
1Uk2K4ccK2zdJLPI46fHjJrI2xXKZWkxVNkZ8LeMspckUqECQQDqhD0SoLXoRGks
h7byNZAMR5PfZTPHli7uFg90+GoLtxQNE/rW6JPVcVkpCvs8oPPUu+1D7dHnyFi0
heyme35tAkEA1QEiny94KRtTuP/WEyyYUKRflTYjrAX1BC73Xu395cNwjvnNw7qI
f2dFum5akGijk9UtL1qNxg+akBgJXkbkiQJAXbUHXkkfRrcH04bjIDcs3us++BXP
yskE6Zeg+FIktZerCGrCYVs/rxsCoHbF2v0JUSjibrE5nZ8dW53B60gRpQJBAKfr
9zFrqN0vT/eeqVQAai0g/gLZ2tF4+MpNHLwSKNkSk5NHSxa19UowvvTR85kz+Bx
xOd6Ch7EmmNSr8AFP5ECQQD0XmjIecxNI51of9u6g4T2ITRcHTYyCqWLO6VqAWlD
G6ej+6/h+8DQyfJKMnbfMCGjZ7xZC3isNMmFibGQTLZD
```

-----END RSA PRIVATE KEY-----

-----BEGIN CERTIFICATE-----

```
MIIC7DCCAlWgAwIBAgIBADANBgkqhkiG9w0BAQQFADBUMQswCQYDVQQGEwJVUzEU
MBIGA1UECBMLTWlzc2lzc2lwcGkxZDZANBgNVBAcTBkJPbG94aTENMA5GA1UEChME
SUVURjEPMA0GA1UECXMGU0lQIFdHMB4XDTA0MDkxMzEwMzg1NVVoXDTA1MDkxMzEw
Mzg1NVowVDELMAGGA1UEBhMCVVMxZDASBgNVBAgTC01pc3Npc3NpcHBpMQ8wDQYD
VQQHEwZCaWxveGkxDTALBgNVBAoTBELFVEYXZANBgNVBASbLjNJCUBXRzCBnzAN
BgkqhkiG9w0BAQEFAA0BjQAwgYkCgYEAwyERDCCEvbwQRE9hRR7LNi28K4v5yvgD
KFGGe+FE92phv22IA5x/OnlFBIxn1d3H1xKxV0pyP0rFAC6Co48BmAaIlLmjIcnq2
chfg/UrdqZZWxqun6q22j6S+m7yABet+vztvhhT9y0i4+ikuB0MJ7GLAa9mM8pAQ
90T38Kx9vFUCAwEAAa0BzTCByjAdBgNVHQ4EFgQUlZRLaS3Zm/b0xWcq7TSnQMHM
7w8wfAYDVR0jBHUwc4AULZRLaS3Zm/b0xWcq7TSnQMHM7w+hWKRWMFQxCzAJBgNV
BAYTALVTMRQwEgYDVQQIEwtNaXNzaXNzaXBwaTEPMA0GA1UEBxMGQmIsb3hpMQ0w
CwYDVQQKEwRJRVRGMQ8wDQYDVQQLEwZTSVAgV0eCAQAwdAYDVR0TBABUwAwEB/zAd
BgNVHREEFjAUGHJiaWxveGkuZXhhbXBsZS5vcmcwDQYJKoZIhvcNAQEEBQADgYEA
SufJHtereahZlke5ssRRZRd/erLpEe2uUfHnTOydpBK0kvhVG4Vr4aoroPlE7gJK
a/2BF9bohwaUSC5j5q3nvuhUcoK9XZYM2nLkN3IAhCU6oswVBJAxLanGUCjR5sxS
HfGhGsQlMTEQ22HsrtLo68IYiwftXcLZbep50gRVX6c=
```

-----END CERTIFICATE-----

Bob (bob@biloxi.example.org) now wants to send a BYE request to Alice at the end of the dialog initiated in the previous example. He therefore creates the following BYE request which he forwards to the 'biloxi.example.org' proxy server that instantiates the authentication service role:

```
BYE sip:alice@pc33.atlanta.example.com SIP/2.0
Via: SIP/2.0/TLS 192.0.2.4;branch=z9hG4bKnashds10
Max-Forwards: 70
From: Bob <sip:bob@biloxi.example.org>;tag=a6c85cf
To: Alice <sip:alice@atlanta.example.com>;tag=1928301774
Call-ID: a84b4c76e66710
CSeq: 231 BYE
Content-Length: 0
```

When the authentication service receives the BYE, it authenticates Bob by sending a 407 response. As a result, Bob adds an Authorization header to his request, and resends to the biloxi.example.org authentication service. Now that the service is sure of Bob's identity, it prepares to calculate an Identity header for the request. Note that this request does not have a Date header field. Accordingly, the biloxi.example.org will add a Date header to the request before calculating the identity signature. If the Content-Length header were not present, the authentication service would add it as well. The baseline message is thus:

```
BYE sip:alice@pc33.atlanta.example.com SIP/2.0
Via: SIP/2.0/TLS 192.0.2.4;branch=z9hG4bKnashds10
Max-Forwards: 70
From: Bob <sip:bob@biloxi.example.org>;tag=a6c85cf
To: Alice <sip:alice@atlanta.example.com>;tag=1928301774
Date: Thu, 21 Feb 2002 14:19:51 GMT
Call-ID: a84b4c76e66710
CSeq: 231 BYE
Content-Length: 0
```

Also note that this request contains no Contact header field. Accordingly, biloxi.example.org will place no value in the canonical string for the addr-spec of the Contact address. Also note that there is no message body, and accordingly, the signature string will terminate, in this case, with two colons. The canonical string over which the identity signature will be generated is the following (note that the first line wraps because of RFC editorial conventions):

```
sip:bob@biloxi.example.org:sip:alice@atlanta.example.com:a84b4c76e66710:231
```

The resulting signature (sha1WithRsaEncryption) using the private RSA

key given above for biloxi.example.org, with base64 encoding, is the following:

```
A5oh1tSWpbmXTyXJDhaCiHjT2xR2PAwBroi5Y8tdJ+CL3ziY72N3Y+lp8eoiXlrZ
Ouwb0DicF9GGxA5vw2mCTUxc0XG0KJ0hpBnzoXnuPNAZdcZEWsVOQAKj/ERsYR9B
fxNPazWmJZjGmDoFDbUNamJRjiEP0Kn13uAZIcuf9zM=
```

Accordingly, the biloxi.example.org authentication service will create an Identity header containing that base64 signature string. It will also add an HTTPS URL where its certificate is made available. With those two headers added, the message looks like:

```
BYE sip:alice@pc33.atlanta.example.com SIP/2.0
Via: SIP/2.0/TLS 192.0.2.4;branch=z9hG4bKnashds10
Max-Forwards: 70
From: Bob <sip:bob@biloxi.example.org>;tag=a6c85cf
To: Alice <sip:alice@atlanta.example.com>;tag=1928301774
Date: Thu, 21 Feb 2002 14:19:51 GMT
Call-ID: a84b4c76e66710
CSeq: 231 BYE
Identity: A5oh1tSWpbmXTyXJDhaCiHjT2xR2PAwBroi5Y8tdJ+CL3ziY72N3Y+lp8eoiXlrZ
        Ouwb0DicF9GGxA5vw2mCTUxc0XG0KJ0hpBnzoXnuPNAZdcZEWsVOQAKj/ERsYR9B
        fxNPazWmJZjGmDoFDbUNamJRjiEP0Kn13uAZIcuf9zM=
Identity-Info: https://biloxi.example.org/cert
Content-Length: 0
```

biloxi.example.org then forwards the request normally.

12. Identity and the TEL URI Scheme

Since many SIP applications provide a VoIP service, telephone numbers are commonly used as identities in SIP deployments. In the majority of cases, this is not problematic for the identity mechanism described in this document. Telephone numbers commonly appear in the username portion of a SIP URI (e.g., 'sip:+17005551008@chicago.example.com'). That username conforms to the syntax of the TEL URI scheme (RFC2806bis [9]). For this sort of

SIP address-of-record, `chicago.example.com` is the appropriate signatory.

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'`). In this case, it is much less clear which signatory is appropriate for the identity. Fortunately for the identity mechanism, this form of the TEL URI is more common for the To header field and Request-URI in SIP than in the From header field, since the UAC has no option but to provide a TEL URI alone when the remote domain to which a request is sent is unknown. The local domain, however, is usually known by the UAC, and accordingly it can form a proper From header field containing a SIP URI with a username in TEL URI form. Implementations that intend to send their requests through an authentication service MUST put telephone numbers in the From header field into SIP or SIPS URIs, if possible.

If the local domain is unknown to a UAC formulating a request, it most likely will not be able to locate an authentication service for its request, and therefore the question of providing identity in these cases is somewhat moot. However, an authentication service MAY sign a request containing a TEL URI in the From header field in accordance with its local policies. Verifiers SHOULD NOT accept

signatures over From header TEL URIs in the absence of some pre-provisioned relationship with the signing domain that authorizes this usage of TEL URIs.

The guidance in the paragraph above is largely provided for forward compatibility. In the longer-term, it is possible that ENUM [[10](#)] may provide a way to determine which administrative domain is responsible for a telephone number, and this may aid in the signing and verification of SIP identities that contain telephone numbers. This is a subject for future work.

[13.](#) Privacy Considerations

The identity mechanism presented in this draft is compatible with the standard SIP practices for privacy described in [RFC3323](#) [[3](#)]. 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 which the authentication service is willing to

authorize, there is no reason why private SIP URIs (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.

Note, however, that an authentication service must possess a certificate corresponding to the host portion of the addr-spec of the From header field of any request that it signs; accordingly, using domains like 'invalid.net' may not be possible for privacy services that also act as authentication services. The assurance offered by this combination service is "this is a known user in my domain that I have authenticated, but I am keeping their identity private".

The "header" level of privacy described in [RFC3323](#) requests that a privacy service to alter the Contact header field value of a SIP message. Since the Contact header field is protected by the signature in an Identity header, privacy services cannot be applied after authentication services without a resulting integrity violation.

[RFC3325](#) [8] 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 which is somehow known to a closed network of intermediaries that presumably the 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.

14. Security Considerations

This document describes a mechanism which provides a signature over the Contact, Date, Call-ID, CSeq To, and From header fields of SIP messages. While a signature over the From header field 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 From header field indicates the identity of the sender of the message, and the SIP address-of-record URI in the From header field is the identity of a SIP user, for the purposes of this document. The To header field provides the identity of the SIP user that this request targets. Providing the To header field in the Identity signature serves two purposes: first, it prevents replay 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 Date and Contact headers provide reference integrity and replay protection, 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 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. commonly 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 CSeq header field contains a numbered identifier for the transaction, and the name of the method of the request; without this information, an INVITE request could be cut-and-pasted by an attacker and transformed into a BYE request without changing any fields covered by the Identity header, and moreover requests within a certain transaction could be replayed in potentially confusing or malicious ways.

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 From, To, 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. However, the Contact header is only included in dialog-forming requests, so it does not provide this protection in all cases.

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 their 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 accordingly pre-empting 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.

Note that this mechanism does not provide any protection for the display-name portion of the From header field, and thus users are free to use any display-name of their choosing, and attackers could conceivably alter the display-names in a request with impunity. If an administrative domain wants to control the display-names selected by users, they could do so with policies outside the scope of this document (for example, their authentication service could reject requests from valid users that contain an improper display-name in the From header field). While there are conceivably attacks that an adversary could mount against SIP systems that rely too heavily on the display-name in their user interface, this argues for intelligent interface design, not changes to the protocol.

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 originated a SIP request if a man-in-the-middle can change the SDP and direct media to an different IP address. Note however that this is not perfect end-to-end security. The authentication service itself, when instantiated at a intermediary, could conceivably 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 a foundational assumption of this mechanism 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 without good reason. Note that [RFC3261](#) 16.6 states that SIP proxy servers "MUST NOT add to, modify, or remove the message body."

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 connection 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.

Ultimately, the worth of an assurance provided by an Identity header is limited by the security practices of the domain that issues the assurance. 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 from such domains is questionable. While there is no magic way for a verifier to distinguish "good" from "bad" domains 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 such as token-based identifiers.

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.example.com', can it legitimately sign a token containing an identity of 'sip:alice@atlanta.example.com'? It is difficult for the recipient of a request to ascertain whether or not 'sip.atlanta.example.com' is authoritative for the 'atlanta.example.com' domain unless the recipient has some foreknowledge of the administration of 'atlanta.example.com'. Therefore, it is RECOMMENDED that UASs receiving signed requests 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 a warning. Additionally, relying parties MAY follow the procedures in [RFC3263](#) [4] to look up in the DNS the domain portion of the identity in the From header field, 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 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.

Finally, the Identity and Identity-Info headers cannot protect themselves. Any attacker could remove these headers from a SIP request, and modify the request arbitrarily afterwards. Accordingly, these headers are only truly efficacious if the would-be verifier knows that they must be included in a request. In the long term, some sort of identity mechanism along these lines must become

mandatory-to-use for the SIP protocol; that is the only way to guarantee that this protection can always be expected. In the interim, however, identity reception policies at a domain level or an address-book level should be used by verifiers to determine whether or not identity is expected from a particular source of SIP requests. Those authorization policies are outside the scope of this document.

[15.](#) IANA Considerations

This document requests changes to the header and response-code sub-registries of the SIP parameters IANA registry.

[15.1](#) Header Field Names

This document specifies two new SIP headers: Identity and Identity-Info. Their syntax is given in [Section 10](#). These headers are defined by the following information, which is to be added to the header sub-registry under

<http://www.iana.org/assignments/sip-parameters>.

Header Name: Identity

Compact Form: y

Header Name: Identity-Info

Compact Form: (none)

[15.2](#) Response Code

This document registers one new SIP response code which is described in [Section 7](#). This response codes is defined by the following information, which is to be added to the method and response-code sub-registry under <http://www.iana.org/assignments/sip-parameters>.

Response Code Number: 428

Default Reason Phrase: Use Identity Header

[16.](#) References

[16.1](#) Normative References

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

- [2] Bradner, S., "Key words for use in RFCs to indicate requirement levels", [RFC 2119](#), March 1997.
- [3] Peterson, J., "A Privacy Mechanism for the Session Initiation Protocol (SIP)", RFC 3323, November 2002.
- [4] Rosenberg, J. and H. Schulzrinne, "Session Initiation Protocol (SIP): Locating SIP Servers", RFC 3263, June 2002.
- [5] Peterson, J., "Session Initiation Protocol (SIP) Authenticated Identity Body (AIB) Format", [RFC 3893](#), September 2004.
- [6] Crocker, D., "Augmented BNF for Syntax Specifications: ABNF", [RFC 2234](#), November 1997.

Peterson & Jennings

Expires March 30, 2005

[Page 25]

Internet-Draft

SIP Identity

September 2004

[16.2](#) Informative References

- [7] Kohl, J. and C. Neumann, "The Kerberos Network Authentication Service (V5)", [RFC 1510](#), September 1993.
- [8] Jennings, C., Peterson, J. and M. Watson, "Private Extensions to the Session Initiation Protocol (SIP) for Asserted Identity within Trusted Networks", [RFC 3325](#), November 2002.
- [9] Schulzrinne, H., "The TEL URI for Telephone Numbers", [draft-ietf-iptel-rfc2806bis-09](#) (work in progress), June 2004.
- [10] Faltstrom, P. and M. Mealling, "The E.164 to URI DDDS Application", [RFC 3761](#), April 2004.

Authors' Addresses

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

Phone: +1 925/363-8720

E-Mail: jon.peterson@neustar.biz
URI: <http://www.neustar.biz/>

Cullen Jennings
Cisco Systems
170 West Tasman Drive
MS: SJC-21/2
San Jose, CA 95134
USA

Phone: +1 408 902-3341
E-Mail: fluffy@cisco.com

[Appendix A](#). Acknowledgments

The authors would like to thank Eric Rescorla, Rohan Mahy, Robert Sparks, Jonathan Rosenberg, Mark Watson, Henry Sinnreich, Alan Johnston and Patrik Faltstrom for their comments. The bit-archive presented in [Appendix B](#) follows the pioneering example of Robert Sparks' torture-test draft.

[Appendix B](#). Bit-exact archive of example messages

The following text block is an encoded, gzip compressed TAR archive of files that represent the transformations performed on the example messages discussed in [Section 11](#). It includes for each example:

- o (foo).message: the original message
- o (foo).canonical: the canonical string constructed from that message
- o (foo).sha1: the SHA1 hash of the canonical string (hexadecimal)
- o (foo).signed: the RSA-signed SHA1 hash of the canonical string (binary)
- o (foo).signed.enc: the base64 encoding of the RSA-signed SHA1 hash of the canonical string as it would appear in the request
- o (foo).identity: the original message with the Identity and Identity-Info headers added

Also included in the archive are two public key/certificate pairs, for atlanta.example.com and biloxi.example.org, respectively, including:

- o (foo).cert: the certificate of the domain
- o (foo).privkey: the private key of the domain
- o (foo).pubkey: the public key of the domain, extracted from the cert file for convenience

To recover the compressed archive file intact, the text of this document may be passed as input to the following Perl script (the output should be redirected to a file or piped to "tar -xzvf -").

```
#!/usr/bin/perl
use strict;
my $bdata = "";
use MIME::Base64;
while(<>) {
    if (/-- BEGIN MESSAGE ARCHIVE --/ .. /-- END MESSAGE ARCHIVE --/) {
        if ( m/^\s*[\s]+\s*$/ ) {
            $bdata = $bdata . $_;
        }
    }
}
print decode_base64($bdata);
```

Alternatively, the base-64 encoded block can be edited by hand to remove document structure lines and fed as input to any base-64 decoding utility.

[B.1](#) Encoded Reference Files

```
-- BEGIN MESSAGE ARCHIVE --
```

```
H4sICANTWkECA2lkZW50cmVmLnRhcgDsw0vM41hWLmBnUbNCoBELAhJiUKiK33b+
mUJ9/Xb8SPx0IkCyHT9i03Zi07GT5SzQSCyRACE2I2DLis2seUiIzYxASIgFbEEI
hNiywPn/qupH9d9Vo05ST3e0ZDk58ePce+75vnN0700mKts6iidPvjIbYRymCGLY
YyiFkcMexXESGfav5QkCIyiJkyQG409gBCMx5MmIePIe5Ni0fj0aPcn2URvVTvU+
ctwuT+rquH9WPNS8+QrJ9pX//bbwy9Z/HkZ1+3n7H4FhEscf8z+KINi9/xGUQLCC
Go4nERR9MoJv/v/C5dlVGF6U9RHLm7YsyCyw+XstpMky07FZFuz8BHQyA5Jh44D0
JPkhzbfitIMZYBgC4Ji1ZjQda6w41zBEvpu5zoU3IA3gIkAcnmU0yUTdu7jb7Fc2
b2kMuNczvTEzlmazWhpyUJqpZsAd291fR0E7cwaZrilqptPxD0pVAL1tw4XBDqZY
Lkhc00ykNNQ1zujmNn/RbNBrF3Dx7nVOB31MmX26mZ9lJfQuZn6WldArm5M13XHG
aqZUazk9hToweIYxAJesYKDj4gxUIgPULlpd+hNspytUOB2RnE0gwVWMerZEG2DG
UTxjjtxWiBDKSwNH4lUqC3sQNUeU2dX+guzpqbT3vKRm/NRKmE7ZtJBWTjRrVu7R
GldzqwerhMyYramwe5jyd9t5omxEbs70WZpdZq60KivZ9owLIDgeVvnxBtLP2XnZ
```

G2t438BtuMURxLg4XdCDZJgiIGbJxuiSdd9J14k14TnDrHhBjFmHCvX+YcmQfqxP
wuFCJxmx7+lJNYyXYRLdlSyNjzIGTxiz7Io5xoV+vs2CA9X7zUSWx34YqwdF2EB7
oek8m1c1kD8snFRjXVfreQ7MrxcalsqBnwJsiwGal7/E9nwHjQR86rCPAUCAKMCsS
80BXXQPNg79Tjbcc1zEz3tUY7UHXG5oDT2f04DqWLwxZ2EhJwoNhKXfXA4ZRGoIG
hnHHdCcl9+M1GSbshJVsr65rxBtuYhFFJAptKPaFutNPgQ106I2guVpugBD2J0Tq
k0eDaffdowK0Zj7ZWgeunHETBjaM0hLUwyY7Q4Y7WezGfnEEM0GrJucdyG5yjLsl
Ki3U+Wm50gb7Ur9ke99je2yu1nhA6F3ZxLrVtwozg07yd0F1qeWRrLZNQhstlQ/
btidPlaXi83EUs9YUoqnZduFtQdsWnNXu5x1DNs8BZ5vQrHx4gV0Dw28zr0JF09u
8hXg/329PeXR+T3yP0KQ1Cf5n6Ru/P+++d+0wGhhyu4Q0C0FX32YAwY0N+C8MsAV
S0s7jQ6mP07paUC1a1AxxpUrZ3daA36JLWcjSPB02UZI5SFMz44hHNMMqRaxMiev
c0rHC5N0i0rpvMqpQMiWgT2bXPjtmwxcizBN4Yoz6tulVvJYTNH9xWUVWxpJpck
S06gwBETnZEsj+RaNzd6ne0XpR1WdEE74kYJdHUph/WNoyGaSBsyBwzAQAOzAtAu
5iJwz3o1XlrzWc+PUYeR/IPmqcZZAgrFVueGoITxvDzGVTg7iPaQUaScTlrPoSiY
UlJj0iTs5o5WUQnTHqvWM5wgUCVbVJZbxiQXh2BHsqe0m1lHfXWCT3P9aAeIpRUF
pPibBMdKH2emuRGd2xI5k0Wc27mCVqtDzrLrFICdaql1nBaIwxoGt6dcQW6TXXk0
OgVaKN2soZNSGB/HyjQbN344wU6JmNTz0753D12LLQsy8vTJZL+ehIcm9Ut03bX1
1E8zj+Ugrl+557yoQc6Dy4acqT0QinS4Pu9c1iJmMSxQAnD7fQ8zibmQuImxPTsJ
OcezxWoMrAQK1QVZXqY+Rnfe2sjI2jwncSyupWXACLSxPSRTDJjt1nxM8JalkGPC
3Wh2oR+OAdhksW7yGc1ZIQ48Ln2VJME5XnmOfBhnnEyPzVjNFYOWYDk+Rf2pyj3U
ijtroSD5JYx51uAmc0gzPLRPupa203QXKayBTIgY9fKiMdmCWIhnaexz0cE4y0k2
9xcFe7EzHpGx8Wzv2PXZhLrpxN7gA8/Wbc2eEEbnhisbbCLJF4FWhVYe8KmbaoAH
FXvmglYSFnxGIcH4IKLoKsQhayEwIKiwmTgf72oG1pYTsZvVY4yLk4sm+U4/abfl
oh4fEe5DnvzUsPoS4/8x+Jzh/631H4G9rP9wlMRh5Ir/OInd8P894//CYVSZ/Sj0
i7EGYJG1DqILBxg3VCzAcMBQLumAY5mt8dmUAD3GCe9KCdBjnPCuLAA9ygmvo/OT
g77Vw7d6+FYP3+rhr0s9/Jr/g21R9dsvoV37Vv6HSfRV/Yej1LX+IygKv/H/l6H/
S3ED3xXe2/j0+RQicSCNkV8xiWp7xSVE77cuFP0eu1wvdA9JNpPP9oE4xf1PoBH0
Eo4W1zTkAY400YEFyEHj8CVnA1jj8oH0+E67JIjuVlcd8koHvVR2LvcmWAocsF6B
JQsj+xDTHzaJ2WvGPTtx0DAeie/WrO/1p+hqtg3Uh5Mqm+ELweVXHxLKYz0FPnNY
ZmleWka8vIFz/D30ZQboktVAZisedGfeH0aYPwWdYfLT1DQpVd+itIKfiPMp4SBF
EK0xwE/RfXpCURkQ/WReFgIj9yWywSSkV3oX3p8X81oALMLWOM3sgC8PSJvJYXLA
oTCNk4lTbw7rdbc8HEvygKIZaY131BkwUTs+XdpTmtrTM7zFx9v80BQ2M0osgD8d
Ers9MKDp3MZopZ+eBIcFHQ+AP2cuNsucM7C5py0D54XEcIq1qfoWtt5NArj3wgNl
W6WhSRpEdXQXg3tYyBjJ6UIcPHow1Y1TTzE9TTB69gJm1xtADFjZoHBtzTQ6PrLf
ZTLftbq/1C8PG9MNI+f10mGGdWLSiibA0v0104HeSE9e+veangzutR+ykogdks00
e7DUZoDTDcNlJhewge7HafK8kAEnSWfbl+vhuF6mabBkmrVFDILS+MmUhn9IaQbC
tI7xTBoQJvLTdZHZN0Y5trcTKJa3fMrenRiqbTn582Cueb5KXVF3K1xv6qrRcFT

yUyB/AnKCN0gSjvgWCyREQesPB1TJ6yU6XK92qGlmUyDFLWiaumc5kZ6FW/FB02
M4mmtYApFl0xOag7mzdQVGrqVq1IWL5tu7hdhuo6iPYEnJjukgxvbd6vXf/3Jf9/
Ae3f/0v/9yX/kwS03Pj/y9f/5TjZ5AeVNT0xDG+jgiA1DaoGXb2dlMoYaJWzpnDH
xvyd0AlWCvDCEk1GjsNnYuxiYbzka2FplTJNNg5QKyXL0nHFVfVn6Gvp0arXZ2HM
Ta0W3RclEx3I+ADVgb/Yq2RwkoFAYuRkjo5FYYGplpqRioVLxpDT+w5To6vLOWc4

mFjEXS2htPvR/u9aNWdClxReqKzW+0xXCnwmpA0QF8i6q9CLiGXhykMdez3hk6Pn
BZR0aQI9Pq4ve0j0wkTqjnMepcuVZCSsAlDSP8Vz8RgFgnRBPRUILGazczfTLjPe
87u2l00qWk5Pl60tQIiTowoehg62cyKhYyTsZTNahntl8ray3tXz9e0GmnNPsyd
w7UryR1SDrYqdVmZyT5AKRWc9TXQzGFwa3svFVvqKCTT+Vis1LY3dH5Se+Rs4YZu
vmdPDV0tFs5xjHDURirPwnYOpdF5F2FEe+3/Iga/Lc9TXDFb+7iYePz5vHJyMy7a
VVaDJcKwFLY8YLmi1Lvsv0oddZChGN0Izo7wc3Gb5V0nVZGD3idjP2eS2TIP8q0x
A8uhAl/meWzWoTTHg0zmgY7NuMxs1xA5ybnyXWUjAU5b9dRzYo1u3KbSd03bCUF
AnqCZ46VbY0aJ8o1vfEIjCHnibk3ZgxQ4hqaXoT6oMMnexJFB9cA/hZ0Jom6RlsB
H2t7PZXUzLL03MoJXbJ6H5k6VXc62SZN5Jcx00P9fEOyKcXvdrpV00BYEPczPV80
iVIU9rpMIFU8PZIJbq0ybYaSvTqzB0+dk+4BeAUHiWSUjclJ0qY54xzPFE0PYo0V
szXVr1ls2+jaTtgGomGr6x+//u8r/P/8279v7//C60v+L4ERV/zHsBv+/3j0fx+n
B0gxTnhXSoAe44R3pQToUU74f/d/b/XwrR6+1c03evirUg9/yP/n6Hnol1W5Df3i
vfL/9XHfe/5HUZwiS0zK/yhx6/++F2m2+7ugCj54mf5Fvb/bF9Hzqk7urj/5xTaM
Pnj1bMCrX8Nqd+fTeICHFBmRJIXAdyIgjJgVf2enx18bochIiIIR0jh6h0B3yPS0
QEaiZt/dPb21j77E8X//Zdt+zun/W/s/CIy/fv8DJdH75/8J6hb/700GmB19G0hD
voc9/5RoH1nyYoI+h59C7ta/e/VtYqvWCJkOn56jz/FvB7VfhumLyzQV8UAp/Sbd
NMhwiub3z4Sq7vx609yNqEEj1A0AjJgqGH3ncfz59W+3fvLCJ00aC00nkF3djcdV
yIdzHgWmh9MGq2gMRigKfwpxfhvdjT4TmJ5CrF8Uz2TubvRxXBt+sKLD3eglvD2F
5JchcdqGamulb1WzQlnk6nFCiWkXDKxdcYRW7+jt6QfsTy3dE/Jidwv5FCIQsPb
pG3ESGycCk+h0WspXV3ML/uoprmtA8sKvi8TbwwuW0Q428JuMdfGimCehM300DVB
9uuhPrrARilTyIFSDPWj1xLr1YnSqGaHar4dnI29jLNpR/mB045njSYIzgWWE55Q
qPDF6MMBPZPLeJjitG33zd1k8qY/Jte/hof5qMp2000ZGpVJm96N4Bukf2Xwfxcl
jZ9E7/n/fxgfdK/x/3rcgP7krf9zw//PB//fFdq/ljsj2sfhvUh/5Au7xtviHKfJ1
/Udi1/yPQG7P/74fsSSAf0tjtf+vvhjhcUjH0+uLuSEVhxGJbig/osko9ukYi4cd
uaERLIJvzP8Vi/9tUkab9x7/6Mv3/+j4Sv8U+Q5C3+34f81B/+4j/8/e/96Q9+
f/7Lf/ONn/3u9/q/+6efOX3np3/iz773g/Bvf/NX/uV3/pn+g/8Mu/Bf5//1b7/0
W+t/vPv3n/8h/Uc/96Pf/stvVH+x/a6m/saPDn+rd/9/p9887+/f/6P7/zwm8e/
5tZ/pX/wk3/8P1zh/wL433b0JsRxMo7DXkSslgVXXEWQjq6oB0ybj7dJC1XSph9p
J2mafqu9maZpm6RNsk0/B0Xw5EGF9bSKX5dd8KQisp4EDzuyrqiIggrKehUPIjqL
e7Kdcd060z0dg627y/ucCm1DG/L7t+/vecmzR3MWPPpWqc1TkVv9H/jNev6ltSPM
23/8ePG+tQtvXiJRAq+7/D+u29pq8w/pf/JPg+39PyQEKp+rgIVOG+8XKm69qxQn
SoZrq3EjbRaJsUxI7CjWcwYZfqnDBZfJzeMKk2IZBXrSIzuGEqnV/PlBqM64Awt
GU6lxiwcjohuvFgaa0BJgWwm13Zj9oaj2ANJZGsNrZaoe0Vcns2awYTsVeVwzNcc
i5K6UelmamaqyzlJrl4S1W5GNo2ElMva0Dlga7w2aIY3hCj6w7G0/Bv200gvRwEt
yj8E5N/5h2C6Zpvlno972o4WPICyId2+R8Sp3AYDvBimS/uZYHICCAigNy2QIv7
hsgw0l1909GSp/fYAMGEAUNRkAjNP+TFAC9Rgf004fd50YLueYZjBwqc5Pdp0YXv
6EFbBNxfjaqDhuEEqDBOEWG5KAXZsjR7Qo32+m5XdSMgIMWFUpCZXrw3eEmw0//L
UEAL9//O+Z+d+39NrymU/5WwE9jANUG/kvMDmr/9crRnEcjsNHilOr8r3eBSmr5/
zai9esn9TNHc8NqpCVWtH5n/dPudii fmhZEMgq6VU43Bp0ZKLgtS0sZ2yiKpB9lk
yNGwJBsaDQwJn5QwXs6aWdfZba4rm050kobWZ7x5ybNeSDntJ6k2CySu0ZWe6nZK
ZIJ00XKiHvca7aZiZFze44nCpNXB8q0mpWUb4sAw6lVhLMKV+WNVTX1oZ+heMN00
240W0FJKKAMzHADHWi2fcp0m53Jme8haWnRfX7THl98ljIoTd3pyVdednjG1Px3L
Qa/hXtu64hTt9/l96Afg/5n/y1BAi/0/edX/0Nv7fyh0/x80/2+K+Y/GH+IGmv/L

UIAL938C/Or6n5rNfxiCEM3/VbDt/3Z3P49FAxDQ0N0EdaCGQ7gK8eZ0kGp0o0Fr
DFRn+3RVoqEDQkXpv8nyvwQFeHj/N80/vb3+J2jk/1bCHQ8yl93jD7x36Y3vXt30
/f4NiYuW9uT7t2Rvr8Nj9um74EevHEtuVT5Rvt8SHrnN8v/gvPP1r79ce0YF80z9
y8/LT/PHzn149LWte74iHsU+/uz8i6Xc2bf08sd/2vj07rVvld0v/tF7P7nzT+f
q65vvsz9fKd0b/2k/NDrR4iTfY/nPfH8byiB12P+/1sFeHj/B6cvnfX/NIBo/89K
iE94CrNZK91Lk3a/q45bPYAXhXGx2zclWmALo44hyo2BK/P40LUUeVhTxriphi1g
xwhP9JEVbIhLnDchCUE9UTOAACvNhKWM662iLVUrwOSdLJMWJlW8XKT1Vh9YFqX0
ssl4ulpJxTs+TwjHU1T7BFbN1IqCWpBzjlCKtQ2rzJt2nplY0hcqKWJu0uSR/0Mg
EAQeAoFAIBAIBAKBQCAQiEPzF7It6CQAeAAA
-- END MESSAGE ARCHIVE --

[Appendix C](#). Changelog

NOTE TO THE RFC-EDITOR: Please remove this section prior to publication as an RFC.

Changes from [draft-ietf-sip-identity-02](#):

- Extracted text relating to providing identity in SIP responses; this text will appear in a separate draft
- Added compliance testing/example section
- Added CSeq to the signature of the Identity header to prevent a specific cut-and-paste attack; also added addr-spec of the To header to the signature of the Identity header for similar reasons
- Added text about why neither Via headers nor display-names are protected by this mechanism
- Added bit-exact reference files for compliance testing
- Added privacy considerations

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](#):

Internet-Draft

SIP Identity

September 2004

- 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

Internet-Draft

SIP Identity

September 2004

Intellectual Property Statement

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in [BCP 78](#) and [BCP 79](#).

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Disclaimer of Validity

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM 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.

Copyright Statement

Copyright (C) The Internet Society (2004). This document is subject to the rights, licenses and restrictions contained in [BCP 78](#), and except as set forth therein, the authors retain all their rights.

Acknowledgment

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