SIP
Internet-Draft
Expires: December 21, 2006

J. Rosenberg
Cisco Systems
June 19, 2006

# Coexistence of P-Asserted-ID and SIP Identity
# draft-rosenberg-sip-identity-coexistence-00

## Status of this Memo



## Copyright Notice



## Abstract

Two mechanisms have been defined to support forms of authenticated caller identity in the Session Initiation Protocol (SIP). The first, specified in RFC 3325, is the P-Asserted-ID header field. The second, termed "SIP Identity", defines the Identity and Identity-Info header fields and provides cryptographically verifiable identities. This document discusses how to use these mechanisms together.

Table of Contents

## 1. Introduction

One of the most important security features in the Session Initiation Protocol (SIP) [1] is the ability to convey the identity of the initiating party of a request. This feature, sometimes known as "secure caller ID", has been the discussion of much discussion, and is supported by numerous specifications.

The first work in secure caller ID is within RFC 3261 itself. SIP provides support for S/MIME. This allows for the initiator of a SIP request to sign the request with their private key, which can then be verified by the recipient using their public key. This mechanism, while very secure, has seen little implementation and no deployment. It requires an easy to use certificate enrollment system by which end users can obtain, store, and manage certificates. To date, systems for providing certificates for end users have proven difficult if not impossible to deploy.

Consequently, implementations relied on the From header field in the SIP request, unsigned, to obtain the identity of the sender. This is easily spoofable and a clear risk. To combat it, the P-Asserted-ID header field was developed [6]. With this mechanism, the originating domain of the requestor authenticates them, typically using traditional SIP digest mechanisms. Once authenticated, the SIP proxy inserts a header field - the P-Asserted-ID header field - containing the authenticated identity of the request originator. This header field is not signed in any way. Instead, the header field is only conveyed between domains that have a specific trust relationship. Domains receiving requests with this header field from domains they don't trust remove the header field. Furthermore, the link between proxies in different domains is secured with SIP over TLS, allowing domains to mutually authenticate each other.

Due to its requirement for bilateral trust agreements between domains, RFC 3325 is only applicable to closed-knit communities of a small number of relatively large providers. For this reason, the P-Asserted-ID header field was granted "P-header" status [7], and was subsequently adopted by the 3gpp for use in the Internet Multimedia Subsystem (IMS).

However, it was recognized by IETF that this mechanism was a short-term solution, and a longer term one was required. It consequently developed the "SIP identity" mechanism [4]. The SIP identity mechanism defines the Identity and Identity-Info header field. As in RFC 3325, an originating proxy in the domain of the requestor authenticates the user, typically using SIP digest. Once authenticated, the originating proxy checks if the From header field value matches the authenticated identity. If it does, it signs

certain header fields, including the From header field, and places the result into the Identity header field. The proxy then populates the Identity-Info header field with a URI that can be used to obtain the certificate for the domain.

The SIP identity mechanism provides a far superior technical solution to secure caller ID than RFC 3325. Its cryptographically verifiable identies are the cornerstone of anti-spam mechanisms [8], which will not work properly with RFC 3325.

Unfortunately, deployment of SIP identity appears problematic due to several practical considerations. Firstly, RFC 3325 has enjoyed widespread deployment. It is build into numerous proxy and application server products, and is also widely used in end user devices. Many IP phones or adaptors will look for the P-Asserted-ID header field as the source for secure caller ID. To bring the SIP identity mechanism into the mix, the caller, proxy, and unfortunately, called party must all be upgraded to support it. Neither the originating proxy or the calling party have any way to know whether the called party supports RFC 3325 or SIP identity, making it difficult to know which to use. To maximize interoperability, it is more cost effective to use the mechanism that is most likely to work - RFC 3325. This produces a chicken-and-egg problem that will substantially hamper the deployment of SIP identity.

Secondly, the SIP identity mechanism provides a signature over the request which covers key parts of it, including the body. This means that any elements on the request path between the originating proxy and the terminating user agent which modify the body in any way will invalidate the signature. Though proxies are not supposed to modify the body, the industry has seen widespread usage of back-to-back user agents with media (B2BUA). These components, to facilitate NAT traversal, call admission control, and other functions, modify the body of SIP requests. SIP identity will not function with such elements on the request path. This adds further to the difficulties in deploying SIP identity.

To combat this problem, this document defines a mechanism for co-existence of SIP identity and P-Asserted-ID which greatly reduces the barriers to deployment for SIP identity.

## 2. Overview of Operation

The essential idea is to use the SIP identity mechanism between proxies, rather than P-Asserted-ID, but to retain the use of P-Asserted-ID as the mechanism for transfer of asserted identity

within a domain. The overall architecture is shown in Figure 1.

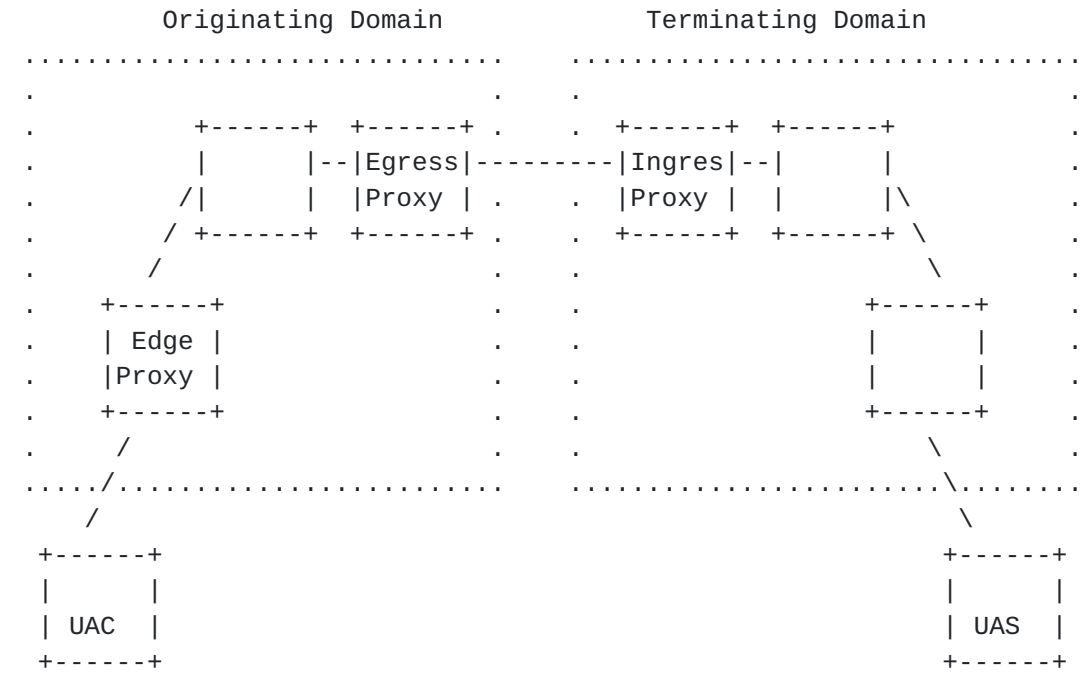


Figure 1

When a UA initiates a request, it is authenticated by the originating edge proxy. Once the originator has been authenticated, the edge proxy inserts the P-Asserted-ID header field, per RFC 3325. This header field remains in the request as long as it stays within the domain of the originator. Once the request reaches the last proxy in the originating domain (the egress proxy), the egress proxy checks the P-Asserted-ID header field against the From header field. If they match, the egress proxy removes the P-Asserted-ID header field, and adds an Identity and Identity-Info header field per [4]. This is sent to the proxy at the edge of the terminating domain (the ingress proxy). The ingress proxy will verify the signature, and if it validates, insert the P-Asserted-ID header field containing the identity in the From header field. However, the Identity and Identity-Info header fields remain in the request.

When the request arrives at the terminating UA, it first checks for the Identity and Identity-Info header fields. If present, the identity in the From header field is used as the caller ID. If not present, but P-Asserted-ID is present, the UA uses the P-Asserted-ID header field as the caller ID.

In order for a UA to determine if its domain supports the mechanisms in this specification, a UA will include a Supported header field in

its REGISTER request with the option tag "id-coexist". If the domain also supports the mechanism, it will include the same option tag in the REGISTER response.

## 3. User Agent Behavior

### 3.1. Registration

When a UA compliant to this specification generates a REGISTER request, it SHOULD include a Supported header field in the request with the option tag "id-coexist". When it receives a successful response to its registration, it checks for the Supported header field and the presence of this option tag. If present, the UA knows that its domain supports the mechanisms of this specification. This information is used in subsequent processing.

OPEN ISSUE: this is a little hoakey. We're using the option tag returned from a registrar to infer the behavior of a different element - the ingress proxy. Is that OK?

### 3.2. Generating a Request

When originating a request besides a REGISTER, there is no special processing required.

### 3.3. Receiving a Request

When receiving an incoming request, the UA first checks for the presence of the Identity and Identity-Info header fields. If present, the UA verifies the signature per [4]. If it verifies, the identity in the From header field is used as the identity of the sender. If it does not verify, but its domain supports the coexistence mechanism (based on presence of the id-coexist option tag in the REGISTER response) and the request contained a P-Asserted-ID header field, the UA interprets this as the presence of a B2BUA with media in the terminating domain. It uses the identity in the P-Asserted-ID header field as the identity of the sender.

If the request did not contain either the Identity or Identity-Info header fields, but did contain the P-Asserted-ID header field, that identity is used as the identity of the sender if the clients domain supports the coexistence mechanism. If the domain of the client doesn't support the co-existence mechanism, but the P-Asserted-ID header field is present, the identity of the sender of the request SHOULD be considered suspect. This specification makes no normative recommendations on how to treat the request. However, implementations should consider that, in this case, the identity

cannot be trusted unless all of the conditions in the limited scope of applicability of RFC 3325 apply.

If the request did not contain the Identity or Identity-Info header fields, and did not contain a P-Asserted-ID header field, the identity of the sender of the request SHOULD be considered suspect. The From header field, without a verified Identity header field, is extremely susceptible to spoofing.

## 4. Proxy Behavior

### 4.1. Edge Proxy

An edge proxy is one that authenticates the originating party and asserts their identity. An edge proxy SHOULD follow the procedures of RFC 3325, with one addition. If there is no P-Preferred-ID header field in the request, it SHOULD use the From header field as the preferred ID.

### 4.2. Egress Proxy

An egress proxy is one that meets the following conditions:

- The next hop proxy is in a different administrative domain.
- The domain of the proxy matches the domain in the From header field of the request.

If a request contains a P-Asserted-ID header field, and is received from a proxy inside of its domain, an egress proxy SHOULD act as an authentication service per [4]. It SHOULD use the P-Asserted-ID header field as the identity of the sender, rather than attempting to authenticate the request using SIP digest or some other mechanism. The ingress proxy SHOULD remove the P-Asserted-ID header field before forwarding the request.

### 4.3. Ingress Proxy

An ingress proxy is one whose previous hop was not in the same administrative domain. When an ingress proxy receives a request, it MUST remove the P-Asserted-ID header field if present. This is done regardless of the trust relationship with the originating domain, and is different from the procedures in RFC 3325, where the P-Asserted-ID is retained if it comes from a trusted peer. Once removed, the ingress proxy checks for the presence of the Identity and Identity-Info header fields. If present, the ingress proxy SHOULD verify the identity of the sender using the procedures in [4]. If the identity

is verified, the ingress proxy SHOULD insert a P-Asserted-ID header field containing the identity contained in the From header field of the request. The proxy SHOULD NOT remove the Identity and Identity-Info header fields. This allows for SIP networks where there are more than two administrative domains in a request path, and also allows for a UA to verify the signature on its own if it should desire.

## 5. Interactions with B2BUAs

By using the SIP identity mechanism between domains, it will continue to work in the presence of Contact or body-modifying B2BUAs in the request path. In particular, any B2BUA on the request path prior to the egress proxy, and any B2BUA on the request path subsequent to the ingress proxy in the domain of the UAS, will not cause the mechanism described here to fail. Note that, in cases where a proxy serves as both a B2BUA and an egress proxy, it MUST perform the B2BUA function prior to the egress functions described here. Similarly, in cases where a proxy serves as a B2BUA and an ingress proxy, it MUST perform the B2BUA function after the ingress functions described here.

It is important to note that the mechanism described here will not work properly in transit networks that contain a B2BUA. A transit network is defined as a SIP domain that is between the domain of the originator and the domain of the terminating UA. If a B2BUA in a transmit network touches the fields covered by the signature, verification will fail at the ingress proxy in the termindating case.

## 6. Interactions with Privacy

Privacy has always been a complicated issue with the various identity mechanisms. The privacy specification, RFC 3323 [2] is used by RFC 3325. However, it has a significant problem in that a UAS cannot differentiate between a private caller (where the P-Asserted-ID has been removed from the request) and identity unavailable (where the domain of the originator didn't support P-Asserted-ID, or where the originating network was the PSTN and no identity could be obtained). The interactions with SIP identity and privacy are even more complicated. RFC 3323 does not work with SIP identity; this is documented in detail in [5].

Combining together RFC 3325 and SIP identity requires privacy mechanisms to be combined as well.

A UA wishing to be anonymous would include an anonymous URI in the From header field. This specification proposes that an anonymous

identity be indicated with the "user=anonymous" URI parameter, extending the existing "user" URI parameter with this value. A UA can either obtain an anonymous URI from its domain with mechanisms TBD, or merely make up a random value for the user part of the URI, using a domain of "anonymous.invalid".

The edge proxy would authenticate the request, and insert P-Asserted-ID as normal. This would be stripped by the egress proxy. If the From header field contained an anonyous URI that matched the P-Asserted-ID header field (possible only if the anonymous URI had been obtained from its domain), the egress proxy would sign the request using SIP identity. Otherwise, it would not.

At the ingress proxy, the signature is verified if present. If not present, no P-Asserted-ID is inserted. At the receiving UAS, if a P-Asserted-ID was present, the "user=anonymous" URI parameter tells the UAS that the requestor is anonymous and verified. If a P-Asserted-ID is not present, the presence of the "user=anonymous" URI parameter tells the UAS that the requestor is anonymous, and that its identity is unverified. It can then render "Anonymous" or whatever is appropriate for the user interface.

TODO: fold in the normative recommendations here into the UA and proxy behaviors described above.

## 7. P-header or not?

With the recommendations in this document, we believe that the applicability of P-Asserted-ID is now no longer limited. It becomes applicable within the intra-domain signaling of any SIP domain. This begs the question of whether the header field name should now be changed to "Asserted-ID". The answer is an emphatic no! One of the benefits of the coexist mechanism is that it is backwards compatible with RFC 3325, which uses the P-Asserted-ID header field. This exposes a weakness in the concepts in RFC 3427, since we will now have a header with the P prefix which is not actually a P-header.

Procedurally, we'd recommend that, if this document moves forward, it be done as an update to both SIP identity and RFC 3325, and be at proposed standard status. Indeed, it should probably be done as an actual revision to RFC 3325.

## 8. Benefits

This mechanism brings many benefits:

- Does not require the originating domain to know whether the terminating domain supports the mechanism.
- Works in the presence of B2BUAs in the originating and terminating domains.
- Makes P-Asserted-ID applicable only in intra-domain environments, eliminating its primary security weakness
- Provides the cryptographic strengths of SIP identity to the terminating domain, so that mechanisms such as spam detection can be effective.
- The proposed privacy solution clearly defines a URI as anonymous independent of the locale and language of the terminating domain
- The proposed privacy solution clearly differentiates an anonymous request (one whose From header field contains the user=anonymous parameter) from one where identity could not be provided

## 9. Security Considerations

The combination of RFC 3325 with SIP identity provides a mechanism that is, overall, less secure than just SIP identity alone. With SIP identity, the signature is inserted by the first hop proxy (edge proxy) which performs the authentication. This allows all other proxies in the originating domain to determine the identity of the originator by verifying the signature. With the mechanism proposed here, proxies within the domain of the originator would use the P-Asserted-ID header field, which lacks any cryptographic signature. This requires a greater degree of trust within the proxies of the domain. A rogue proxy in the domain of the originator could insert a fake P-Asserted-ID header field, and it would not be caught by the coexist mechanism.

In addition, the mechanism here relies on a UAS to trust its terminating domain to follow the procedures defined here and verify the signature in the Identity header field. If a rogue proxy in the terminating domain should insert a fake P-Asserted-ID header field, this would not be caught by the coexist mechanism.

Though its overall security is weaker than SIP identity, we fear that the perfect is the enemy of the good. Without changes, SIP identity will be undeployable, and the industry will instead stick with the much-worse P-Asserted-ID solution. The coexist mechanism trades some security in exchange for a mechanism that is far more deployable.

As with RFC 3325, the links over which P-Asserted-ID is transmitted SHOULD be secured with SIP over TLS. This prevents against MITM attacks.

## 10. IANA Considerations

This specification defines a new SIP option tag and a new SIP URI parameter.

### 10.1. Option Tag

This specification registers a new SIP option tag, as per the guidelines in Section 27.1 of RFC 3261.

Name: id-coexist

Description: This option tag is used to identify the ID coexist mechanism. It is primarily used to tell a UA that its domain supports mechanisms which allow for the coexistence of P-Asserted-ID and SIP identity.

### 10.2. URI Parameter

This specification extends the value of the user URI parameter, as per the registry created by [3].

Name of the Parameter: user

Predefined Values: anonymous

RFC Reference: RFC XXXX [[NOTE TO IANA: Please replace XXXX with the RFC number of this specification.]]

## 11. References

### 11.1. 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] Peterson, J., "A Privacy Mechanism for the Session Initiation Protocol (SIP)", RFC 3323, November 2002.

[3] Camarillo, G., "The Internet Assigned Number Authority (IANA) Uniform Resource Identifier (URI) Parameter Registry for the

Session Initiation Protocol (SIP)", BCP 99, RFC 3969, December 2004.

[4] Peterson, J. and C. Jennings, "Enhancements for Authenticated Identity Management in the Session Initiation Protocol (SIP)", draft-ietf-sip-identity-06 (work in progress), October 2005.

[5] Rosenberg, J., "Identity Privacy in the Session Initiation Protocol (SIP)", draft-rosenberg-sip-identity-privacy-00 (work in progress), July 2005.

### 11.2. Informative References

[6] 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.

[7] Mankin, A., Bradner, S., Mahy, R., Willis, D., Ott, J., and B. Rosen, "Change Process for the Session Initiation Protocol (SIP)", BCP 67, RFC 3427, December 2002.

[8] Rosenberg, J., "The Session Initiation Protocol (SIP) and Spam", draft-ietf-sipping-spam-02 (work in progress), March 2006.

Author's Address

Jonathan Rosenberg
Cisco Systems
600 Lanidex Plaza
Parsippany, NJ 07054
US

Phone: +1 973 952-5000
Email: jdrosen@cisco.com
URI: http://www.jdrosen.net

## Intellectual Property Statement



## Disclaimer of Validity



## Copyright Statement



## Acknowledgment

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