

Network Working Group
Internet-Draft
Intended status: Informational
Expires: April 07, 2014

J. Peterson
NeuStar, Inc.
H. Schulzrinne
Columbia University
H. Tschofenig
Nokia Siemens Networks
October 04, 2013

Secure Telephone Identity Problem Statement
draft-ietf-stir-problem-statement-00.txt

Abstract

Over the past decade, Voice over IP (VoIP) systems based on SIP have replaced many traditional telephony deployments. Interworking VoIP systems with the traditional telephone network has reduced the overall security of calling party number and Caller ID assurances by granting attackers new and inexpensive tools to impersonate or obscure calling party numbers when orchestrating bulk commercial calling schemes, hacking voicemail boxes or even circumventing multi-factor authentication systems trusted by banks. Despite previous attempts to provide a secure assurance of the origin of SIP communications, we still lack of effective standards for identifying the calling party in a VoIP session. This document examines the reasons why providing identity for telephone numbers on the Internet has proven so difficult, and shows how changes in the last decade may provide us with new strategies for attaching a secure identity to SIP sessions.

Status of This Memo

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

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

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

This Internet-Draft will expire on April 07, 2014.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Problem Statement	3
3. Terminology	5
4. Use Cases	6
4.1. VoIP-to-VoIP Call	6
4.2. IP-PSTN-IP Call	7
4.3. PSTN-to-VoIP Call	8
4.4. VoIP-to-PSTN Call	8
4.5. PSTN-VoIP-PSTN Call	9
4.6. PSTN-to-PSTN Call	10
5. Limitations of Current Solutions	10
5.1. P-Asserted-Identity	11
5.2. SIP Identity	12
5.3. VIPR	15
6. Environmental Changes	17
6.1. Shift to Mobile Communication	17
6.2. Failure of Public ENUM	18
6.3. Public Key Infrastructure Developments	18
6.4. Pervasive Nature of B2BUA Deployments	19
6.5. Stickiness of Deployed Infrastructure	19
6.6. Relationship with Number Assignment and Management	19
7. Requirements	20
8. Acknowledgments	21
9. IANA Considerations	21
10. Security Considerations	21
11. Informative References	21
Authors' Addresses	23

1. Introduction

In many communication architectures that allow users to communicate with other users, the need for identifying the originating party that initiates a call or a messaging interaction arises. The desire for identifying the communication parties in the end-to-end communication attempt derives from the need to implement authorization policies (to grant or reject call attempts) but has also been utilized for charging. While there are a number of ways to enable identification this functionality has been provided by the Session Initiation Protocol (SIP) [2] by using two main types of approaches, namely using P-Asserted-Identity (PAI) [5] and SIP Identity [1], which are described in more detail in Section 5. The goal of these mechanisms is to validate that originator of a call is authorized to claim an originating identifier. Protocols, like XMPP, use mechanisms that are conceptional similar to those offered by SIP.

Although solutions have been standardized, it turns out that the current deployment situation is unsatisfactory and, even worse, there is little indication that it will be improved in the future. In [9] we illustrate what challenges arise. In particular, interworking with different communication architectures (e.g., SIP, PSTN, XMPP, RTCWeb) or other forms of mediation breaks the end-to-end semantic of the communication interaction and destroys any identification capabilities. Furthermore, the use of different identifiers (e.g., E.164 numbers vs. SIP URIs) creates challenges for determining who is able to claim "ownership" for a specific identifier; although domain-based identifiers (sip:user@example.com) might use certificate or DNS-related approaches to determine who is able to claim "ownership" of the URI, telephone numbers do not yet have any similar mechanism defined.

After the publication of the PAI and SIP Identity specifications various further attempts have been made to tackle the topic but unfortunately with little success. The complexity resides in the deployment situation and the long list of (often conflicting) requirements. A number of years have passed since the last attempts were made to improve the situation and we therefore believe it is time to give it another try. With this document we would like to start an attempt to develop a common understanding of the problem statement as well as requirements to develop a vision on how to advance the state of the art and to initiate technical work to enable secure call origin identification.

2. Problem Statement

In the classical public-switched telephone network, a limited number of carriers trusted each other, without any cryptographic validation, to provide accurate caller origination information. In some cases, national telecommunication regulation codified these obligations.

This model worked as long as the number of entities was relatively small, easily identified (e.g., through the concept of certificated carriers) and subject to effective legal sanctions in case of misbehavior. However, for some time, these assumptions have no longer held true. For example, entities that are not traditional telecommunication carriers, possibly located outside the country whose country code they are using, can act as voice service providers. While in the past, there was a clear distinction between customers and service providers, VoIP service providers can now easily act as customers, originating and transit providers. For telephony, Caller ID spoofing has become common, with a small subset of entities either ignoring abuse of their services or willingly serving to enable fraud and other illegal behavior.

For example, recently, enterprises and public safety organizations [15] have been subjected to telephony denial-of-service attacks. In this case, an individual claiming to represent a collections company for payday loans starts the extortion scheme with a phone call to an organization. Failing to get payment from an individual or organization, the criminal organization launches a barrage of phone calls, with spoofed numbers, preventing the targeted organization from receiving legitimate phone calls. Other boiler-room organizations use number spoofing to place illegal "robocalls" (automated telemarketing, see, for example, the FCC webpage [16] on this topic). Robocalls is a problem that has been recognized already by various regulators, for example the Federal Communications Commission (FCC) recently organized a robocall competition to solicit ideas for creating solutions that will block illegal robocalls [17]. Criminals may also use number spoofing to impersonate banks or bank customers to gain access to information or financial accounts.

In general, number spoofing is used in two ways, impersonation and anonymization. For impersonation, the attacker pretends to be a specific individual. Impersonation can be used for pretexting, where the attacker obtains information about the individual impersonated, activates credit cards or for harassment, e.g., by causing utility services to be disconnected, take-out food to be delivered, or by causing police to respond to a non-existing hostage situation ("swatting", see [19]). Some voicemail systems can be set up so that they grant access to stored messages without a password, relying solely on the caller identity. As an example, the News International phone-hacking scandal [18] has also gained a lot of press attention where employees of the newspaper were accused of engaging in phone hacking by utilizing Caller ID spoofing to get access to a voicemail. For numbers where the caller has suppressed textual caller identification, number spoofing can be used to retrieve this information, stored in the so-called Calling Name (CNAM) database. For anonymization, the caller does not necessarily care whether the

number is in service, or who it is assigned to, and may switch rapidly and possibly randomly between numbers. Anonymization facilitates automated illegal telemarketing or telephony denial-of-service attacks, as described above, as it makes it difficult to blacklist numbers. It also makes tracing such calls much more labor-intensive, as each such call has to be identified in each transit carrier hop-by-hop, based on destination number and time of call.

Secure origin identification should prevent impersonation and, to a lesser extent, anonymization. However, if numbers are easy and cheap to obtain, and if the organizations assigning identifiers cannot or will not establish the true corporate or individual identity of the entity requesting such identifiers, robocallers will still be able to switch between many different identities.

It is insufficient to simply outlaw all spoofing of originating telephone numbers, because the entities spoofing numbers are already committing other crimes and thus unlikely to be deterred by legal sanctions. Also, in some cases, third parties may need to temporarily use the identity of another individual or organization, with full consent of the "owner" of the identifier. For example:

The doctor's office: Physicians calling their patients using their cell phones would like to replace their mobile phone number with the number of their office to avoid being called back by patients on their personal phone.

Call centers: Call centers operate on behalf of companies and the called party expects to see the Caller ID of the company, not the call center.

3. Terminology

The following terms are defined in this document:

In-band Identity Conveyance: In-band conveyance is the presence of call origin identification information conveyed within SIP. It takes the nature of E.164 numbers and the prevalence of B2BUAs into account.

Out-of-Band Identity Verification: Out-of-band verification determines whether the E.164 number used by the calling party actually exists, whether the calling entity is entitled to use the number and whether a call has recently been made from this phone number. This approach is needed when the in-band technique does not work due to intermediaries or due to interworking with PSTN networks.

Authority Delegation Infrastructure: This functionality defines how existing authority over E.164 telephone numbers are used in number portability and delegation cases. It also describes how the existing numbering infrastructure is re-used to maintain the lifecycle of number assignments.

Canonical Telephone Number: In order for either in-band conveyance or out-of-band verification to work, entities in this architecture must be able to canonicalize telephone numbers to arrive at a common syntactical form.

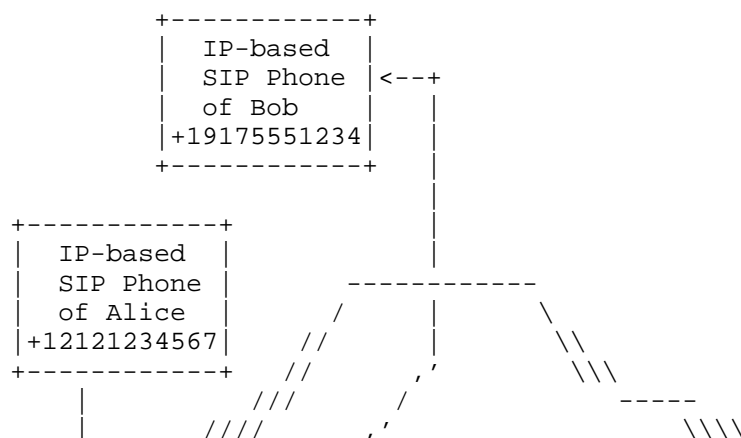
4. Use Cases

In order to explain the requirements and other design assumptions we will explain some of the scenarios that need to be supported by any solution. To reduce clutter, the figures do not show call routing elements, such as SIP proxies, of voice or text service providers. We generally assume that the PSTN component of any call path cannot be altered.

4.1. VoIP-to-VoIP Call

For the IP-to-IP communication case, a group of service providers that offer interconnected VoIP service exchange calls using SIP end-to-end, but may also deliver some calls via circuit-switched facilities, as described in separate use cases below. These service providers use telephone numbers as source and destination identifiers, either as the user component of a SIP URI (e.g., sip:12125551234@example.com) or as a tel URI [8].

As illustrated in Figure 1, if Alice calls Bob, the call will use SIP end-to-end. (The call may or may not traverse the Internet.)



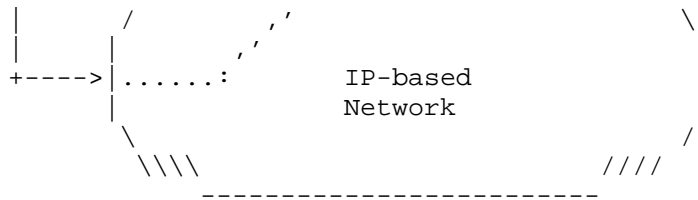


Figure 1: VoIP-to-VoIP Call.

4.2. IP-PSTN-IP Call

Frequently, two VoIP-based service providers are not directly connected by VoIP and use TDM circuits to exchange calls, leading to the IP-PSTN-IP use case. In this use case, Dan's VSP is not a member of the interconnect federation Alice's and Bob's VSP belongs to. As far as Alice is concerned Dan is not accessible via IP and the PSTN is used as an interconnection network. Figure 2 shows the resulting exchange.

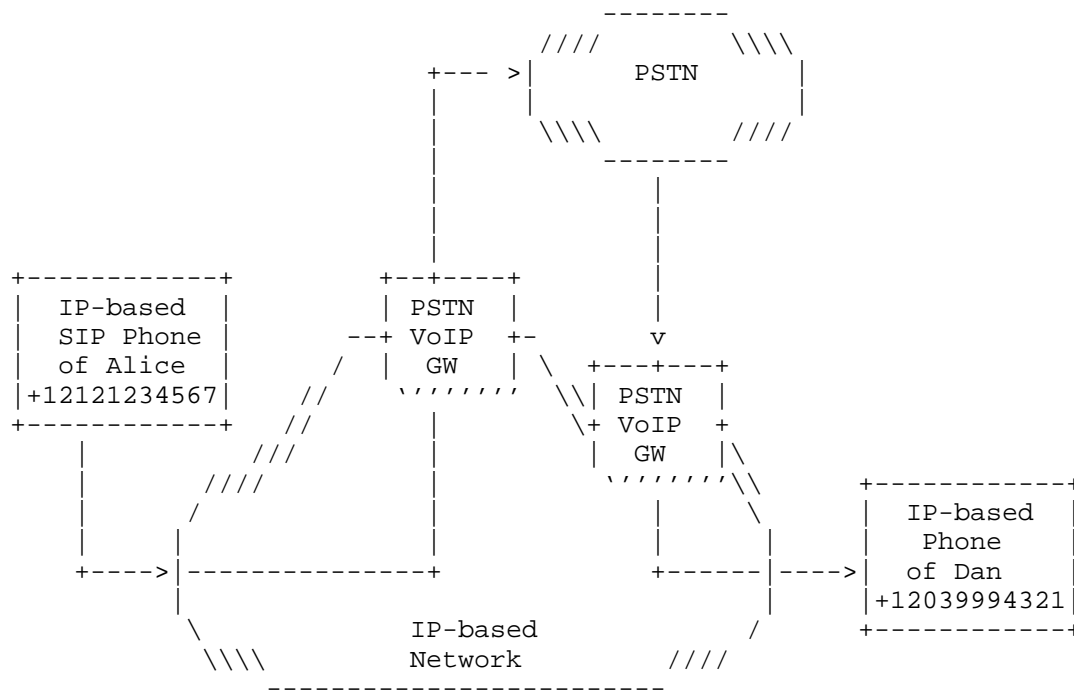


Figure 2: IP-PSTN-IP Call.

Note: A B2BUA/Session Border Controller (SBC) exhibits behavior that looks similar to this scenario since the original call content would, in the worst case, be re-created on the call origination side.

4.3. PSTN-to-VoIP Call

Consider Figure 3 where Carl is using a PSTN phone and initiates a call to Alice. Alice is using a VoIP-based phone. The call of Carl traverses the PSTN and enters the Internet via a PSTN/VoIP gateway. This gateway attaches some identity information to the call, for example based on the information it had received through the PSTN, if available.

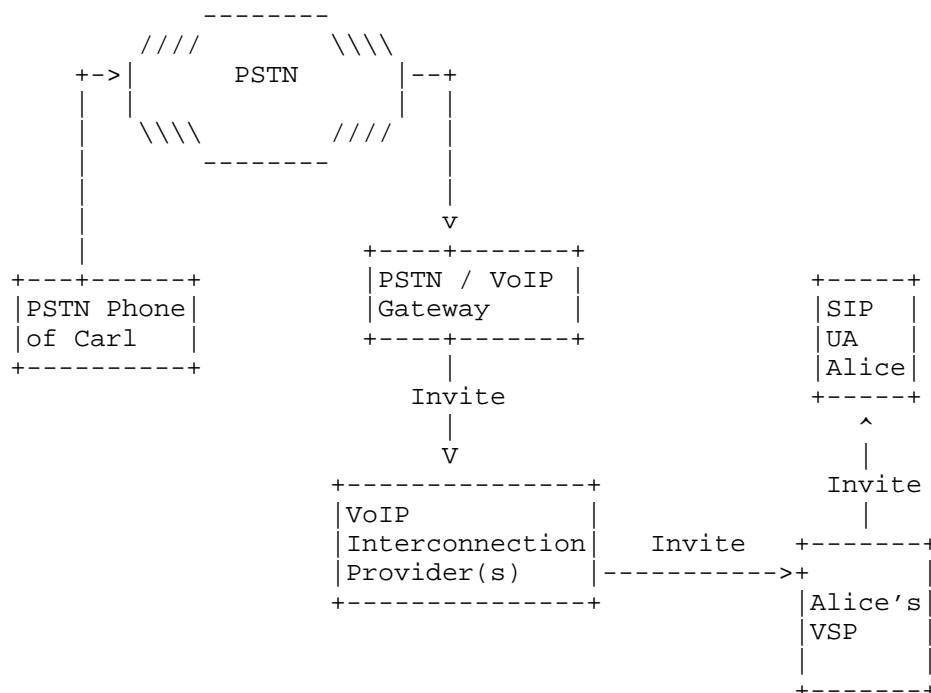


Figure 3: PSTN-to-VoIP Call.

4.4. VoIP-to-PSTN Call

Consider Figure 4 where Alice calls Carl. Carl uses a PSTN phone and Alice an IP-based phone. When Alice initiates the call the E.164 number needs to get translated to a SIP URI and subsequently to an IP address. The call of Alice traverses her VoIP provider where the call origin identification information is added. It then hits the PSTN/VoIP gateway. The gateway must verify that Alice can claim the

E.164 number she is using before it populates the corresponding calling party number field in telephone network signaling. Carl's phone must be able to verify that it is receiving a legitimate call from the calling party number it will render to Carl.

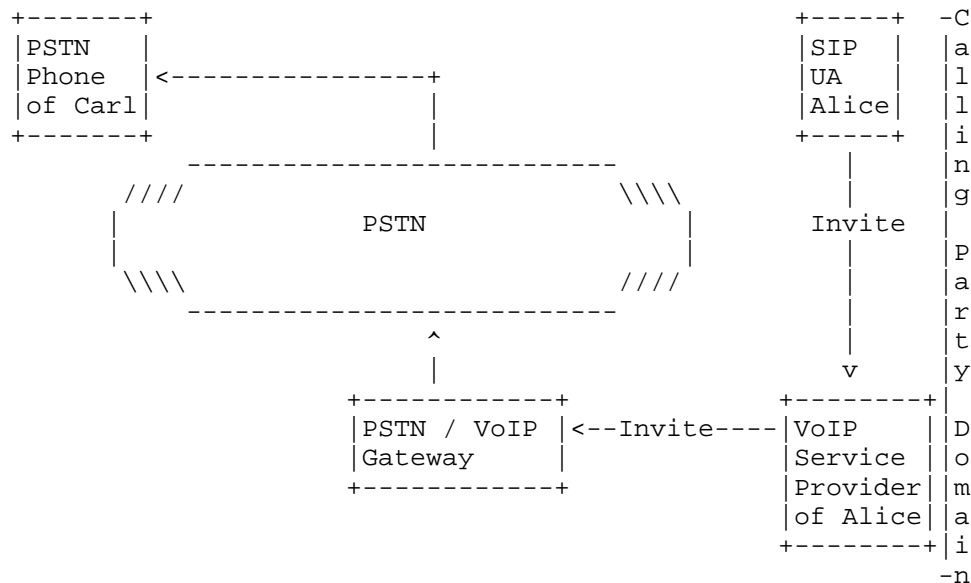
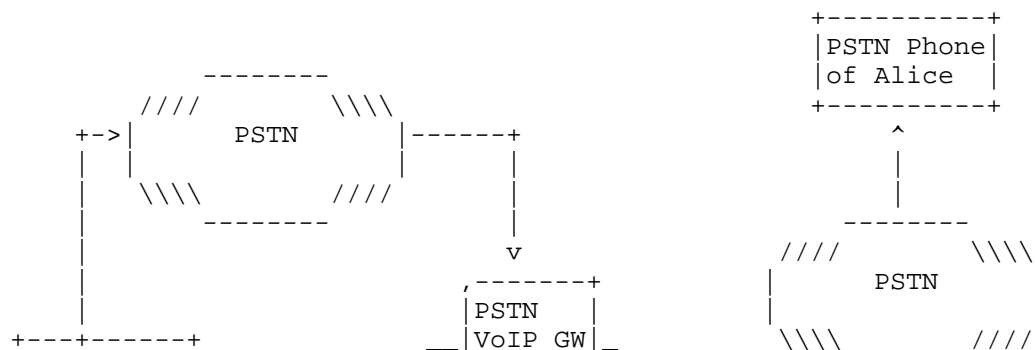


Figure 4: IP-to-PSTN Call.

4.5. PSTN-VoIP-PSTN Call

Consider Figure 5 where Carl calls Alice. Both users have PSTN phones but interconnection between the two PSTN networks is accomplished via an IP network. Consequently, Carl's operator uses a PSTN-to-VoIP gateway to route the call via an IP network to a gateway to break out into the PSTN again.



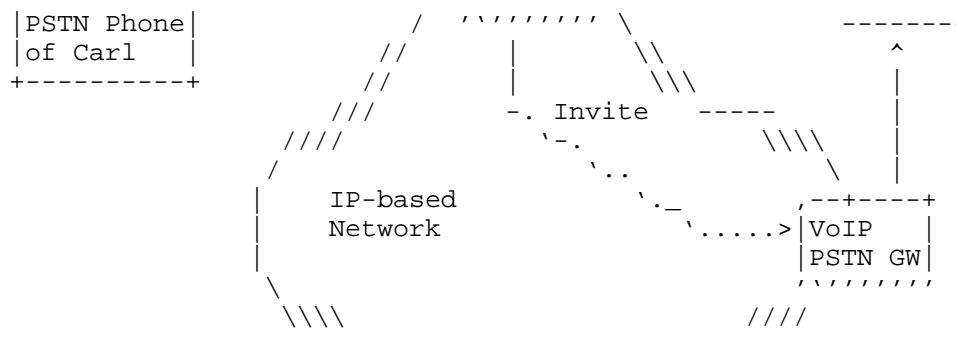


Figure 5: PSTN-VoIP-PSTN Call.

4.6. PSTN-to-PSTN Call

For the "legacy" case of a PSTN-to-PSTN call, otherwise beyond improvement, we may be able to use out-of-band IP connectivity at both the originating and terminating carrier to validate the call information.

5. Limitations of Current Solutions

From the inception of SIP, the From header field value has held an arbitrary user-supplied identity, much like the From header field value of an SMTP email message. During work on [2], efforts began to provide a secure origin for SIP requests as an extension to SIP. The so-called "short term" solution, the P-Asserted-Identity header described in [5], is deployed fairly widely, even though it is limited to closed trusted networks where end-user devices cannot alter or inspect SIP messages and offers no cryptographic validation. As P-Asserted-Identity is used increasingly across multiple networks, it cannot offer any protection against identity spoofing by intermediaries or entities that allow untrusted entities to set the P-Asserted-Identity information.

Subsequent efforts to prevent calling origin identity spoofing in SIP include the SIP Identity effort (the "long term" identity solution) [1] and Verification Involving PSTN Reachability (VIPR) [13]. SIP Identity attaches a new header field to SIP requests containing a signature over the From header field value combined with other message components to prevent replay attacks. SIP Identity is meant both to prevent originating calls with spoofed From headers and intermediaries, such as SIP proxies, from launching man-in-the-middle attacks to alter calls passing through. The VIPR architecture attacked a broader range of problems relating to spam, routing and identity with a new infrastructure for managing rendezvous and security, which operated alongside of SIP deployments.

As we will describe in more detail below, both SIP Identity and VIPR suffer from serious limitations that have prevented their deployment at significant scale, but they may still offer ideas and protocol building blocks for a solution.

5.1. P-Asserted-Identity

The P-Asserted-Identity header field of SIP [5] provides a way for trusted network entities to share with one another an authoritative identifier for the originator of a call. The value of P-Asserted-Identity cannot be populated by a user, though if a user wants to suggest an identity to the trusted network, a separate header (P-Preferred-Identity) enables them to do so. The features of the P-Asserted-Identity header evolved as part of a broader effort to reach parity with traditional telephone network signaling mechanisms for selectively sharing and restricting presentation of the calling party number at the user level, while still allowing core network elements to know the identity of the user for abuse prevention and accounting.

In order for P-Asserted-Identity to have these properties, it requires the existence of a trust domain as described in [4]. Any entity in the trust domain may add a P-Asserted-Identity header to a SIP message, and any entity in the trust domain may forward a message with a P-Asserted-Identity header to any other entity in the trust domain. If a trusted entity forwards a SIP request to an untrusted entity, however, the P-Asserted-Identity header must first be removed; most sorts of end user devices are outside trust domains. Sending a P-Asserted-Identity request to an untrusted entity could leak potentially private information, such as the network-asserted calling party number in a case where a caller has requested presentation restriction. This concept of a trust domain is modeled on the trusted network of devices that operate the traditional telephone network.

P-Asserted-Identity has been very successful in telephone replacement deployments of SIP. It is an extremely simple in-band mechanism, requiring no cryptographic operations. Since it is so reminiscent of legacy mechanisms in the traditional telephone network, and it interworks so seamlessly with those protocols, it has naturally been favored by providers comfortable with these operating principles.

In practice, a trust domain exhibits many of the same merits and flaws as the traditional telephone network when it comes to securing a calling party number. Any trusted entity may provide P-Asserted-Identity, and a recipient of a SIP message has no direct assurance of who generated the P-Asserted-Identity header field value: all trust is transitive. Trust domains are dictated by business arrangements more than by security standards, and thus the level of assurance of P-Asserted-Identity is only as good as the least trustworthy member of a trust domain. Since the contents of P-Asserted-Identity are not intended for consumption by end users, end users must trust that their service provider participates in an appropriate trust domain, as there will be no direct evidence of the trust domain in SIP signaling that end user devices receive. Since the mechanism is so closely modeled on the traditional telephone network, it is unlikely to provide a higher level of security than that.

Since [5] was written, the whole notion of P- headers intended for use in private SIP domains has also been deprecated, largely because of overwhelming evidence that these headers were being used outside of private contexts and leaking into the public Internet. It is unclear how many deployments that make use of P-Asserted-Identity in fact conform with the Spec-T requirements of RFC3324.

P-Asserted-Identity also complicates the question of which URI should be presented to a user when a call is received. Per RFC3261, SIP user agents would render the contents of the From header field to a user when receiving an INVITE request, but what if the P-Asserted-Identity contains a more trustworthy URI, and presentation is not restricted? Subsequent proposals have suggested additional header fields to carry different forms of identity related to the caller, including billing identities. As the calling identities in a SIP request proliferate, the question of how to select one to render to the end user becomes more difficult to answer.

5.2. SIP Identity

The SIP Identity mechanism [1] provided two header fields for securing identity information in SIP requests: the Identity and Identity-Info header fields. Architecturally, the SIP Identity mechanism assumes a classic "SIP trapezoid" deployment in which an authentication service, acting on behalf of the originator of a SIP

request, attaches identity information to the request which provides partial integrity protection; a verification service acting on behalf of the recipient validates the integrity of the request when it is received.

The Identity header field value contains a signature over a hash of selected elements of a SIP request, including several header field values (most significantly, the From header field value) and the entirety of the body of the request. The set of header field values was chosen specifically to prevent cut-and-paste attacks; it requires the verification service to retain some state to guard against replays. The signature over the body of a request has different properties for different SIP methods, but all prevent tampering by man-in-the-middle attacks. For a SIP MESSAGE request, for example, the signature over the body covers the actual message conveyed by the request: it is pointless to guarantee the source of a request if a man-in-the-middle can change the content of the message, as in that case the message content is created by an attacker. Similar threats exist against the SIP NOTIFY method. For a SIP INVITE request, a signature over the SDP body is intended to prevent a man-in-the-middle from changing properties of the media stream, including the IP address and port to which media should be sent, as this provides a means for the man-in-the-middle to direct session media to resource that the originator did not specify, and thus to impersonate an intended listener.

The Identity-Info header field value contains a URI designating the location of the certificate corresponding to the private key that signed the hash in the Identity header. That certificate could be passed by-value along with the SIP request, in which case a "cid" URI appears in Identity-Info, or by-reference, for example when the Identity-Info header field value has the URL of a service that delivers the certificate. [1] imposes further constraints governing the subject of that certificate: namely, that it must cover the domain name indicated in the domain component of the URI in the From header field value of the request.

The SIP Identity mechanism, however, has two fundamental limitations that have precluded its deployment: first, that it provides Identity only for domain names rather than other identifiers; second, that it does not tolerate intermediaries that alter the bodies, or certain header fields, of SIP requests.

As deployed, SIP predominantly mimics the structures of the telephone network, and thus uses telephone numbers as identifiers. Telephone numbers in the From header field value of a SIP request may appear as the user part of a SIP URI, or alternatively in an independent tel URI. The certificate designated by the Identity-Info header field as

specified, however, corresponds only to the domain portion of a SIP URI in the From header field. As such, [1] does not have any provision to identify the assignee of a telephone number. While it could be the case that the domain name portion of a SIP URI signifies a carrier (like "att.com") to whom numbers are assigned, the SIP Identity mechanism provides no assurance that a number is assigned to any carrier. For a tel URI, moreover, it is unclear in [1] what entity should hold a corresponding certificate. A caller may not want to reveal the identity of its service provider to the callee, and may thus prefer tel URIs in the From header field.

This lack of authority gives rise to a whole class of SIP identity problems when dealing with telephone numbers, as is explored in [11]. That document shows how the Identity header of a SIP request targeting a telephone number (embedded in a SIP URI) could be dropped by an intermediate domain, which then modifies and resigns the request, all without alerting the verification service: the verification service has no way of knowing which original domain signed the request. Provided that the local authentication service is complicit, an originator can claim virtually any telephone number, impersonating any chosen Caller ID from the perspective of the verifier. Both of these attacks are rooted in the inability of the verification service to ascertain a specific certificate that is authoritative for a telephone number.

As deployed, SIP is moreover highly mediated, and mediated in ways that [2] did not anticipate. As request routing commonly depends on policies dissimilar to [14], requests transit multiple intermediate domains to reach a destination; some forms of intermediaries in those domains may effectively re-initiate the session.

One of the main reasons that SIP deployments mimic the PSTN architecture is because the requirement for interconnection with the PSTN remains paramount: a call may originate in SIP and terminate on the PSTN, or vice versa; and worse still, a PSTN-to-PSTN call may transit a SIP network in the middle, or vice versa. This necessarily reduces SIP's feature set to the least common denominator of the telephone network, and mandates support for telephone numbers as a primary calling identifier.

Interworking with non-SIP networks makes end-to-end identity problematic. When a PSTN gateway sends a call to a SIP network, it creates the INVITE request anew, regardless of whether a previous leg of the call originated in a SIP network that later dropped the call to the PSTN. As these gateways are not necessarily operated by entities that have any relationship to the number assignee, it is unclear how they could provide an identity signature that a verifier should trust. Moreover, how could the gateway know that the calling

party number it receives from the PSTN is actually authentic? And when a gateway receives a call via SIP and terminates a call to the PSTN, how can that gateway verify that a telephone number in the From header field value is authentic, before it presents that number as the calling party number in the PSTN?

Similarly, some SIP networks deploy intermediaries that act as back-to-back user agents (B2BUAs), typically in order to provide policy or interworking functions at network boundaries (hence the nickname "Session Border Controller"). These functions range from topology hiding, to alterations necessary to interoperate successfully with particular SIP implementations, to simple network address translation from private address space. To achieve these aims, these entities modify SIP INVITE requests in transit, potentially changing the From, Contact and Call-ID header field values, as well as aspects of the SDP, including especially the IP addresses and ports associated with media. Consequently, a SIP request exiting a B2BUA has no necessary relationship to the original request received by the B2BUA, much like a request exiting a PSTN gateway has no necessary relationship to any SIP request in a pre-PSTN leg of the call. An Identity signature provided for the original INVITE has no bearing on the post-B2BUA INVITE, and, were the B2BUA to preserve the original Identity header, any verification service would detect a violation of the integrity protection.

The SIP community has long been aware of these problems with [1] in practical deployments. Some have therefore proposed weakening the security constraints of [1] so that at least some deployments of B2BUAs will be compatible with integrity protection of SIP requests. However, such solutions do not address one key problem identified above: the lack of any clear authority for telephone numbers, and the fact that some INVITE requests are generated by intermediaries rather than endpoints. Removing the signature over the SDP from the Identity header will not, for example, make it any clearer how a PSTN gateway should assert identity in an INVITE request.

5.3. VIPR

Verification Involving PSTN Reachability (VIPR) directly attacks the twin problems of identifying number assignees on the Internet and coping with intermediaries that may modify signaling. To address the first problem, VIPR relies on the PSTN itself: it discovers which endpoints on the Internet are reachable via a particular PSTN number by calling the number on the PSTN to determine whom a call to that number will reach. As VIPR-enabled Internet endpoints associated with PSTN numbers are discovered, VIPR provides a rendez-vous service that allows the endpoints of a call to form an out-of-band connection over the Internet; this connection allows the endpoints to exchange

information that secures future communications and permits direct, unmediated SIP connections.

VIPR provides these services within a fairly narrow scope of applicability. Its seminal use case is the enterprise IP PBX, a device that has both PSTN connectivity and Internet connectivity, which serves a set of local users with telephone numbers; after a PSTN call has connected successfully and then ended, the PBX searches a distributed hash-table to see if any VIPR-compatible devices have advertised themselves as a route for the unfamiliar number on the Internet. If advertisements exist, the originating PBX then initiates a verification process to determine whether the entity claiming to be the assignee of the unfamiliar number in fact received the successful call: this involves verifying details such as the start and stop times of the call. If the destination verifies successfully, the originating PBX provisions a local database with a route for that telephone number to the URI provided by the proven destination. The destination moreover gives a token to the originator that can be inserted in future call setup messages to authenticate the source of future communications.

Through this mechanism, the VIPR system provides a suite of properties, ones that go well beyond merely securing the origins of communications. It also provides a routing system which dynamically discovers mappings between telephone numbers and URIs, effectively building an ad hoc ENUM database in every VIPR implementation. The tokens exchanged over the out-of-band connection established by VIPR moreover provide an authorization mechanism for accepting calls over the Internet that significantly reduces the potential for spam. Because the token can act as a nonce due to the presence of this out-of-band connectivity, the VIPR token is less susceptible to cut-and-paste attacks and thus needs to cover with its signature far less of a SIP request.

Due to its narrow scope of applicability, and the details of its implementation, VIPR has some significant limitations. The most salient for the purposes of this document is that it only has bearing on repeated communications between entities: it has solution to the classic "robocall" problem, where the target typically receives a call from a number that has never called before. All of VIPR's strengths in establishing identity and spam prevention kick in only after an initial PSTN call has been completed, and subsequent attempts at communication begin. Every VIPR-compliant entity moreover maintains its own stateful database of previous contacts and authorizations, which lends itself to more aggregators like IP PBXs that may front for thousands of users than to individual phones. That database must be refreshed by periodic PSTN calls to determine that control over the number has not shifted to some other entity;

figuring out when data has grown stale is one the challenges of the architecture. As VIPR requires compliant implementations to operate both a PSTN interface and an IP interface, it has little apparent applicability to ordinary desktop PCs or similar devices with no ability to place direct PSTN calls.

The distributed hash table also creates a new attack surface for impersonation. Attackers who want to pose as the owners of telephone numbers can advertise themselves as routes to a number in the hash table. VIPR has no inherent restriction on the number of entities that may advertise themselves as routes for a number, and thus an originator may find multiple advertisements for a number on the DHT even when an attack is not in progress. As for attackers, even if they cannot successfully verify themselves to the originators of calls (because they lack the call detail information), they may learn from those verification attempts which VIPR entities recently placed calls to the target number: it may be that this information is all the attacker hopes to glean. The fact that advertisements and verifications are public results from the public nature of the DHT that VIPR creates. The public DHT prevents any centralized control, or attempts to impede communications, but those come at the cost of apparently unavoidable privacy losses.

Because of these limitations, VIPR, much like SIP Identity, has had little impact in the marketplace. Ultimately, VIPR's utility as an identity mechanism is limited by its reliance on the PSTN, especially its need for an initial PSTN call to complete before any of VIPR's benefits can be realized, and by the drawbacks of the highly-public exchanges requires to create the out-of-band connection between VIPR entities. As such, there is no obvious solution to providing secure origin services for SIP on the Internet today.

6. Environmental Changes

6.1. Shift to Mobile Communication

In the years since [1] was conceived, there have been a number of fundamental shifts in the communications marketplace. The most transformative has been the precipitous rise of mobile smart phones, which are now arguably the dominant communications device in the developed world. Smart phones have both a PSTN and an IP interface, as well as an SMS and MMS capabilities. This suite of tools suggests that some of the techniques proposed by VIPR could be adapted to the smart phone environment. The installed base of smart phones is moreover highly upgradable, and permits rapid adoption out-of-band rendezvous services for smart phones that circumvent the PSTN: for example, the Apple iMessage service, which allows iPhone users to send SMS messages to one another over the Internet rather than over

the PSTN. Like VIPR, iMessage creates an out-of-band connection over the Internet between iPhones; unlike VIPR, the rendezvous service is provided by a trusted centralized database of iPhones rather than by a DHT. While Apple's service is specific to customers of its smart phones, it seems clear that similar databases could be provided by neutral third parties in a position to coordinate between endpoints.

6.2. Failure of Public ENUM

At the time [1] was written, the hopes for establishing a certificate authority for telephone numbers on the Internet largely rested on public ENUM deployment. The e164.arpa DNS tree established for ENUM could have grown to include certificates for telephone numbers or at least for number ranges. It is now clear however that public ENUM as originally envisioned has little prospect for adoption. That said, national authorities for telephone numbers are increasingly migrating their provisioning services to the Internet, and issuing credentials that express authority for telephone numbers to secure those services. These new authorities for numbers could provide to the public Internet the necessary signatory authority for securing calling parties' numbers. While these systems are far from universal, the authors of this draft believe that a solution devised for the North American Numbering Plan could have applicability to other country codes.

6.3. Public Key Infrastructure Developments

Also, there have been a number of recent high-profile compromises of web certificate authorities. The presence of numerous (in some cases, of hundreds) of trusted certificate authorities in modern web browsers has become a significant security liability. As [1] relied on web certificate authorities, this too provides new lessons for any work on revising [1]: namely, that innovations like DANE [6] that designate a specific certificate preferred by the owner of a DNS name could greatly improve the security of a SIP identity mechanism; and moreover, that when architecting new certificate authorities for telephone numbers, we should be wary of excessive pluralism. While a chain of delegation with a progressively narrowing scope of authority (e.g., from a regulatory entity to a carrier to a reseller to an end user) is needed to reflect operational practices, there is no need to have multiple roots, or peer entities that both claim authority for the same telephone number or number range.

6.4. Pervasive Nature of B2BUA Deployments

Given the prevalence of established B2BUA deployments, we may have a further opportunity to review the elements signed by [1] and to decide on the value of alternative signature mechanisms. Separating the elements necessary for (a) securing the From header field value and preventing replays, from (b) the elements necessary to prevent men-in-the-middle from tampering with messages, may also yield a strategy for identity that will be practicable in some highly mediated networks. Solutions in this space must however remain mindful of the requirements for securing cryptographic material necessary to support DTLS-SRTP or future security mechanisms.

6.5. Stickiness of Deployed Infrastructure

One thing that has not changed, and is not likely to change in the future, is the transitive nature of trust in the PSTN. When a call from the PSTN arrives at a SIP gateway with a calling party number, the gateway will have little chance of determining whether the originator of the call was authorized to claim that calling party number. Due to roaming and countless other factors, calls on the PSTN may emerge from administrative domains that have no relationship with the number assignee. This use case will remain the most difficult to tackle for an identity system, and may prove beyond repair. It does however seem that with the changes in the solution space, and a better understanding of the limits of [1] and VIPR, we are today in a position to reexamine the problem space and find solutions that can have a significant impact on the secure origins problem.

6.6. Relationship with Number Assignment and Management

Currently, telephone numbers are typically managed in a loose delegation hierarchy. For example, a national regulatory agency may task a private, neutral entity with administering numbering resources, such as area codes, and a similar entity with assigning number blocks to carriers and other authorized entities, who in turn then assign numbers to customers. Resellers with looser regulatory obligations can complicate the picture, and in many cases it is difficult to distinguish the roles of enterprises from carriers. In many countries, individual numbers are portable between carriers, at least within the same technology (e.g., wireline-to-wireline). Separate databases manage the mapping of numbers to switch identifiers, companies and textual caller ID information.

As the PSTN transitions to using VoIP technologies, new assignment policies and management mechanisms are likely to emerge. For example, it has been proposed that geography could play a smaller

role in number assignments, and that individual numbers are assigned to end users directly rather than only to service providers, or that the assignment of numbers does not depend on providing actual call delivery services.

Databases today already map telephone numbers to entities that have been assigned the number, e.g., through the LERG (originally, Local Exchange Routing Guide) in the United States. Thus, the transition to IP-based networks may offer an opportunity to integrate cryptographic bindings between numbers or number ranges and service providers into databases.

7. Requirements

This section describes the high level requirements of the effort:

Generation: Intermediaries as well as end system must be able to generate the source identity information.

Validation: Intermediaries as well as end system must be able to validate the source identity information.

Usability: Any validation mechanism must work without human intervention, e.g., CAPTCHA-like mechanisms.

Deployability: Must survive transition of the call to the PSTN and the presence of B2BUAs.

Reflecting existing authority: Must stage credentials on existing national-level number delegations, without assuming the need for an international golden root on the Internet.

Accommodating current practices: Must allow number portability among carriers and must support legitimate usage of number spoofing (doctor's office and call centers)

Minimal payload overhead: Must lead to minimal expansion of SIP headers fields to avoid fragmentation in deployments that use UDP.

Efficiency: Must minimize RTTs for any network lookups and minimize any necessary cryptographic operations.

Privacy: Any out-of-band validation protocol must not allow third parties to learn what numbers have been called by a specific caller.

Some requirements specifically outside the scope of the effort include:

Display name: This effort does not consider how the display name of the caller might be validated.

Response authentication: This effort only considers the problem of providing secure telephone identity for requests, not for responses to requests; no solution is here proposed for the problem of determining to which number a call has connected.

8. Acknowledgments

We would like to thank Alissa Cooper, Bernard Aboba, Sean Turner, Eric Burger, and Eric Rescorla for their discussion input that lead to this document.

9. IANA Considerations

This memo includes no request to IANA.

10. Security Considerations

This document is about improving the security of call origin identification.

11. Informative References

- [1] Peterson, J. and C. Jennings, "Enhancements for Authenticated Identity Management in the Session Initiation Protocol (SIP)", RFC 4474, August 2006.
- [2] 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.
- [3] 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.
- [4] Watson, M., "Short Term Requirements for Network Asserted Identity", RFC 3324, November 2002.
- [5] 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.

- [6] Hoffman, P. and J. Schlyter, "The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA", RFC 6698, August 2012.
- [7] Elwell, J., "Connected Identity in the Session Initiation Protocol (SIP)", RFC 4916, June 2007.
- [8] Schulzrinne, H., "The tel URI for Telephone Numbers", RFC 3966, December 2004.
- [9] Cooper, A., Tschofenig, H., Peterson, J., and B. Aboba, "Secure Call Origin Identification", draft-cooper-iab-secure-origin-00 (work in progress), November 2012.
- [10] Peterson, J., "Retargeting and Security in SIP: A Framework and Requirements", draft-peterson-sipping-retarget-00 (work in progress), February 2005.
- [11] Rosenberg, J., "Concerns around the Applicability of RFC 4474", draft-rosenberg-sip-rfc4474-concerns-00 (work in progress), February 2008.
- [12] Kaplan, H. and V. Pascual, "Loop Detection Mechanisms for Session Initiation Protocol (SIP) Back-to- Back User Agents (B2BUAs)", draft-ietf-straw-b2bua-loop-detection-02 (work in progress), September 2013.
- [13] Barnes, M., Jennings, C., Rosenberg, J., and M. Petit-Huguenin, "Verification Involving PSTN Reachability: Requirements and Architecture Overview", draft-jennings-vipr-overview-04 (work in progress), February 2013.
- [14] Rosenberg, J. and H. Schulzrinne, "Session Initiation Protocol (SIP): Locating SIP Servers", RFC 3263, June 2002.
- [15] Krebs, B., "DHS Warns of 'TDoS' Extortion Attacks on Public Emergency Networks", URL: <http://krebsonsecurity.com/2013/04/dhs-warns-of-tdos-extortion-attacks-on-public-emergency-networks/>, Apr 2013.
- [16] FCC, ., "Robocalls", URL: <http://www.fcc.gov/guides/robocalls>, Apr 2013.
- [17] FCC, ., "FCC Robocall Challenge", URL: <http://robocall.challenge.gov/>, Apr 2013.

- [18] Wikipedia, ., "News International phone hacking scandal", URL: http://en.wikipedia.org/wiki/News_International_phone_hacking_scandal, Apr 2013.
- [19] Wikipedia, ., "Don't Make the Call: The New Phenomenon of 'Swatting'", URL: <http://www.fbi.gov/news/stories/2008/february/swatting020408>, Feb 2008.

Authors' Addresses

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

Email: jon.peterson@neustar.biz

Henning Schulzrinne
Columbia University
Department of Computer Science
450 Computer Science Building
New York, NY 10027
US

Phone: +1 212 939 7004
Email: hgs+ecrit@cs.columbia.edu
URI: <http://www.cs.columbia.edu>

Hannes Tschofenig
Nokia Siemens Networks
Linnoitustie 6
Espoo 02600
Finland

Phone: +358 (50) 4871445
Email: Hannes.Tschofenig@gmx.net
URI: <http://www.tschofenig.priv.at>

Network Working Group
Internet-Draft
Intended status: Informational
Expires: November 10, 2014

J. Peterson
NeuStar, Inc.
H. Schulzrinne
Columbia University
H. Tschofenig

May 9, 2014

Secure Telephone Identity Problem Statement and Requirements
draft-ietf-stir-problem-statement-05.txt

Abstract

Over the past decade, Voice over IP (VoIP) systems based on SIP have replaced many traditional telephony deployments. Interworking VoIP systems with the traditional telephone network has reduced the overall security of calling party number and Caller ID assurances by granting attackers new and inexpensive tools to impersonate or obscure calling party numbers when orchestrating bulk commercial calling schemes, hacking voicemail boxes or even circumventing multi-factor authentication systems trusted by banks. Despite previous attempts to provide a secure assurance of the origin of SIP communications, we still lack of effective standards for identifying the calling party in a VoIP session. This document examines the reasons why providing identity for telephone numbers on the Internet has proven so difficult, and shows how changes in the last decade may provide us with new strategies for attaching a secure identity to SIP sessions. It also gives high-level requirements for a solution in this space.

Status of This Memo

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

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

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

This Internet-Draft will expire on November 10, 2014.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
2. Problem Statement	4
3. Terminology	6
4. Use Cases	6
4.1. VoIP-to-VoIP Call	6
4.2. IP-PSTN-IP Call	7
4.3. PSTN-to-VoIP Call	8
4.4. VoIP-to-PSTN Call	9
4.5. PSTN-VoIP-PSTN Call	10
4.6. PSTN-to-PSTN Call	11
5. Limitations of Current Solutions	11
5.1. P-Asserted-Identity	12
5.2. SIP Identity	14
5.3. VIPR	17
6. Environmental Changes	19
6.1. Shift to Mobile Communication	19
6.2. Failure of Public ENUM	19
6.3. Public Key Infrastructure Developments	20
6.4. Prevalence of B2BUA Deployments	20
6.5. Stickiness of Deployed Infrastructure	20
6.6. Concerns about Pervasive Monitoring	21
6.7. Relationship with Number Assignment and Management	21
7. Basic Requirements	21
8. Acknowledgments	22
9. IANA Considerations	23
10. Security Considerations	23
11. Informative References	23
Authors' Addresses	25

1. Introduction

In many communication architectures that allow users to communicate with other users, the need arises for identifying the originating party that initiates a call or a messaging interaction. The desire for identifying communication parties in end-to-end communication attempt derives from the need to implement authorization policies (to grant or reject call attempts) but has also been utilized for charging. While there are a number of ways to enable identification this functionality has been provided by the Session Initiation Protocol (SIP) [RFC3261] by using two main types of approaches, namely using P-Asserted-Identity (PAI) [RFC3325] and SIP Identity [RFC4474], which are described in more detail in Section 5. The goal of these mechanisms is to validate that originator of a call is authorized to claim an originating identifier. Protocols, like XMPP, use mechanisms that are conceptually similar to those offered by SIP.

Although solutions have been standardized, it turns out that the current deployment situation is unsatisfactory and, even worse, there is little indication that it will be improved in the future. In [I-D.cooper-iab-secure-origin] we illustrate what challenges arise. In particular, interworking with different communication architectures (e.g., SIP, PSTN, XMPP, RTCWeb) or other forms of mediation breaks the end-to-end semantic of the communication interaction and destroys any identification capabilities. Furthermore, the use of different identifiers (e.g., E.164 numbers vs. SIP URIs) creates challenges for determining who is able to claim "ownership" for a specific identifier; although domain-based identifiers (sip:user@example.com) might use certificate or DNS-related approaches to determine who is able to claim "ownership" of the URI, telephone numbers do not yet have any similar mechanism defined.

After the publication of the PAI and SIP Identity specifications various further attempts have been made to tackle the topic but unfortunately with little success. The complexity resides in the deployment situation and the long list of (often conflicting) requirements. A number of years have passed since the last attempts were made to improve the situation and we therefore believe it is time to give it another try. With this document we would like to start to develop a common understanding of the problem statement as well as basic requirements to develop a vision on how to advance the state of the art and to initiate technical work to enable secure call origin identification.

2. Problem Statement

In the classical public-switched telephone network, there were a limited number of carriers, all of whom trusted each other to provide accurate caller origination information, in an environment without any cryptographic validation. In some cases, national telecommunication regulation codified these obligations. This model worked as long as the number of entities was relatively small, easily identified (e.g., in the manner carriers are certified in the US) and subject to effective legal sanctions in case of misbehavior. However, for some time, these assumptions have no longer held true. For example, entities that are not traditional telecommunication carriers, possibly located outside the country whose country code they are using, can act as voice service providers. While in the past, there was a clear distinction between customers and service providers, VoIP service providers can now easily act as customers, originating and transit providers. The problem is moreover not limited to voice communications, as growth in text messaging has made it another vector for bulk unsolicited commercial messaging relying on impersonation of a source telephone number (sometimes a short code). For telephony, Caller ID spoofing has become common, with a small subset of entities either ignoring abuse of their services or willingly serving to enable fraud and other illegal behavior.

For example, recently, enterprises and public safety organizations [TDOS] have been subjected to telephony denial-of-service attacks. In this case, an individual claiming to represent a collections company for payday loans starts the extortion scheme with a phone call to an organization. Failing to get payment from an individual or organization, the criminal organization launches a barrage of phone calls, with spoofed numbers, preventing the targeted organization from receiving legitimate phone calls. Other boiler-room organizations use number spoofing to place illegal "robocalls" (automated telemarketing, see, for example, the US Federal Communications Commission webpage [robocall-fcc] on this topic). Robocalls are a problem that has been recognized already by various regulators; for example, the US Federal Trade Commission (FTC) recently organized a robocall competition to solicit ideas for creating solutions that will block illegal robocalls [robocall-competition]. Criminals may also use number spoofing to impersonate banks or bank customers to gain access to information or financial accounts.

In general, number spoofing is used in two ways, impersonation and anonymization. For impersonation, the attacker pretends to be a specific individual. Impersonation can be used for pretexting, where the attacker obtains information about the individual impersonated, activates credit cards or for harassment, e.g., by causing utility

services to be disconnected, take-out food to be delivered, or by causing police to respond to a non-existing hostage situation ("swatting", see [swatting]). Some voicemail systems can be set up so that they grant access to stored messages without a password, relying solely on the caller identity. As an example, the News International phone-hacking scandal [news-hack] has also gained a lot of press attention where employees of the newspaper were accused of engaging in phone hacking by utilizing Caller ID spoofing to get access to a voicemail. For numbers where the caller has suppressed textual caller identification, number spoofing can be used to retrieve this information, stored in the so-called Calling Name (CNAM) database. For anonymization, the caller does not necessarily care whether the number is in service, or who it is assigned to, and may switch rapidly and possibly randomly between numbers. Anonymization facilitates automated illegal telemarketing or telephony denial-of-service attacks, as described above, as it makes it difficult to identify perpetrators and craft policies to block them. It also makes tracing such calls much more labor-intensive, as each call has to be identified in each transit carrier hop-by-hop, based on destination number and time of call.

It is insufficient to simply outlaw all spoofing of originating telephone numbers, because the entities spoofing numbers are already committing other crimes and thus unlikely to be deterred by legal sanctions. Secure origin identification should prevent impersonation and, to a lesser extent, anonymization. However, if numbers are easy and cheap to obtain, and if the organizations assigning identifiers cannot or will not establish the true corporate or individual identity of the entity requesting such identifiers, robocallers will still be able to switch between many different identities.

The problem space is further complicated by a number of use cases where entities in the telephone network legitimately send calls on behalf of others, including "Find-Me/Follow-Me" services. Ultimately, any SIP entity can receive an INVITE and forward it to any other entity, and the recipient of a forwarded message has little means to ascertain which recipient a call should legitimately target (see [I-D.peterson-sipping-retarget]). Also, in some cases, third parties may need to temporarily use the identity of another individual or organization, with full consent of the "owner" of the identifier. For example:

The doctor's office: Physicians calling their patients using their cell phones would like to replace their mobile phone number with the number of their office to avoid being called back by patients on their personal phone.

Call centers: Call centers operate on behalf of companies and the called party expects to see the Caller ID of the company, not the call center.

3. Terminology

The following terms are defined in this document:

In-band Identity Conveyance: In-band conveyance is the presence of call origin identification information conveyed within the control plane protocol(s) setting up a call. Any in-band solution must accommodate prevalence of in-band intermediaries such as B2BUAs.

Out-of-Band Identity Verification: Out-of-band verification determines whether the telephone number used by the calling party actually exists, whether the calling entity is entitled to use the number and whether a call has recently been made from this phone number. This approach is needed because the in-band technique does not work in all cases, as when certain intermediaries are involved or due to interworking with PSTN networks.

Authority Delegation Infrastructure: This functionality defines how existing authority over telephone numbers are used in number portability and delegation cases. It also describes how the existing numbering infrastructure is re-used to maintain the lifecycle of number assignments.

Canonical Telephone Number: In order for either in-band conveyance or out-of-band verification to work, entities in this architecture must be able to canonicalize telephone numbers to arrive at a common syntactical form.

4. Use Cases

In order to explain the requirements and other design assumptions we will explain some of the scenarios that need to be supported by any solution. To reduce clutter, the figures do not show call routing elements, such as SIP proxies, of voice or text service providers. We generally assume that the PSTN component of any call path cannot be altered.

4.1. VoIP-to-VoIP Call

For the IP-to-IP communication case, a group of service providers that offer interconnected VoIP service exchange calls using SIP end-to-end, but may also deliver some calls via circuit-switched facilities, as described in separate use cases below. These service providers use telephone numbers as source and destination

identifiers, either as the user component of a SIP URI (e.g., sip:12125551234@example.com) or as a tel URI [RFC3966].

As illustrated in Figure 1, if Alice calls Bob, the call will use SIP end-to-end. (The call may or may not traverse the Internet.)

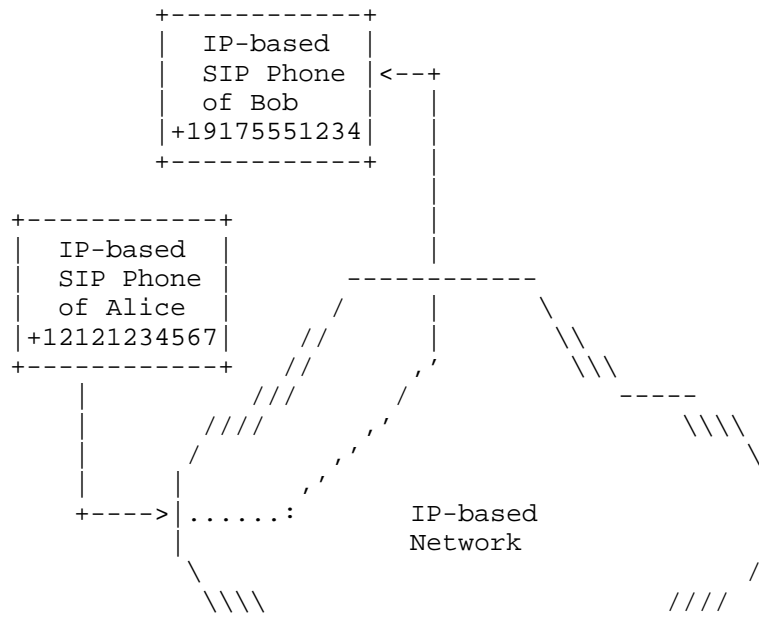


Figure 1: VoIP-to-VoIP Call.

4.2. IP-PSTN-IP Call

Frequently, two VoIP-based service providers are not directly connected by VoIP and use TDM circuits to exchange calls, leading to the IP-PSTN-IP use case. In this use case, Dan's VSP is not a member of the interconnect federation Alice's and Bob's VSP belongs to. As far as Alice is concerned Dan is not accessible via IP and the PSTN is used as an interconnection network. Figure 2 shows the resulting exchange.

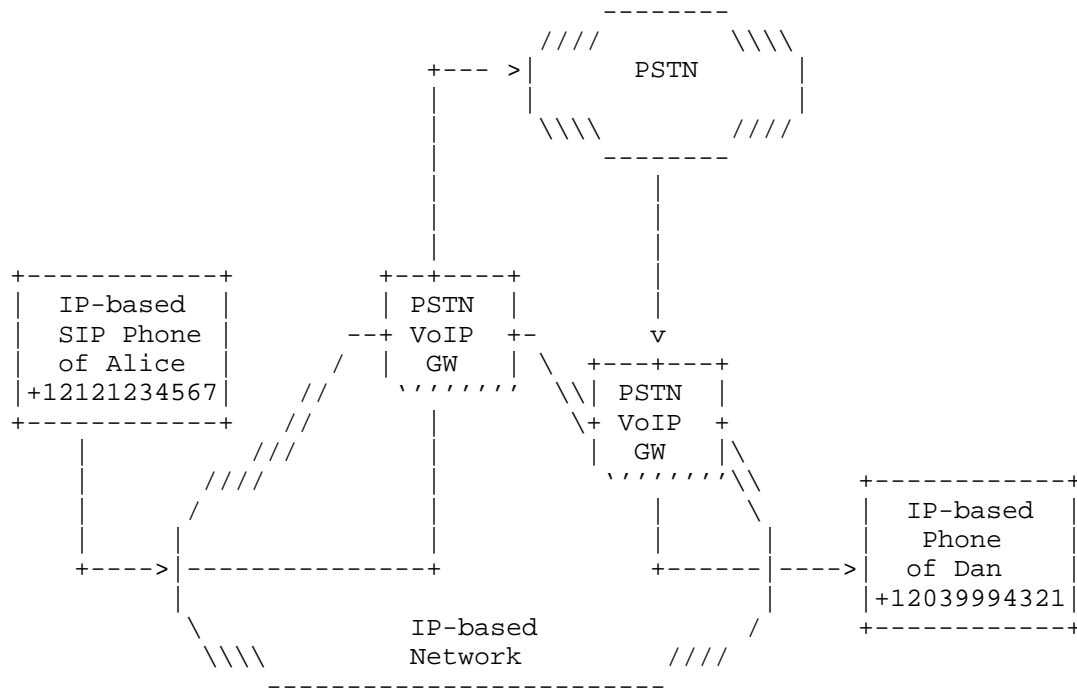


Figure 2: IP-PSTN-IP Call.

Note: A B2BUA/Session Border Controller (SBC) exhibits behavior that looks similar to this scenario since the original call content would, in the worst case, be re-created on the call origination side.

4.3. PSTN-to-VoIP Call

Consider Figure 3 where Carl is using a PSTN phone and initiates a call to Alice. Alice is using a VoIP-based phone. The call from Carl traverses the PSTN and enters the Internet via a PSTN/VoIP gateway. This gateway attaches some identity information to the call, for example, based on the caller identification information it had received through the PSTN, if available.

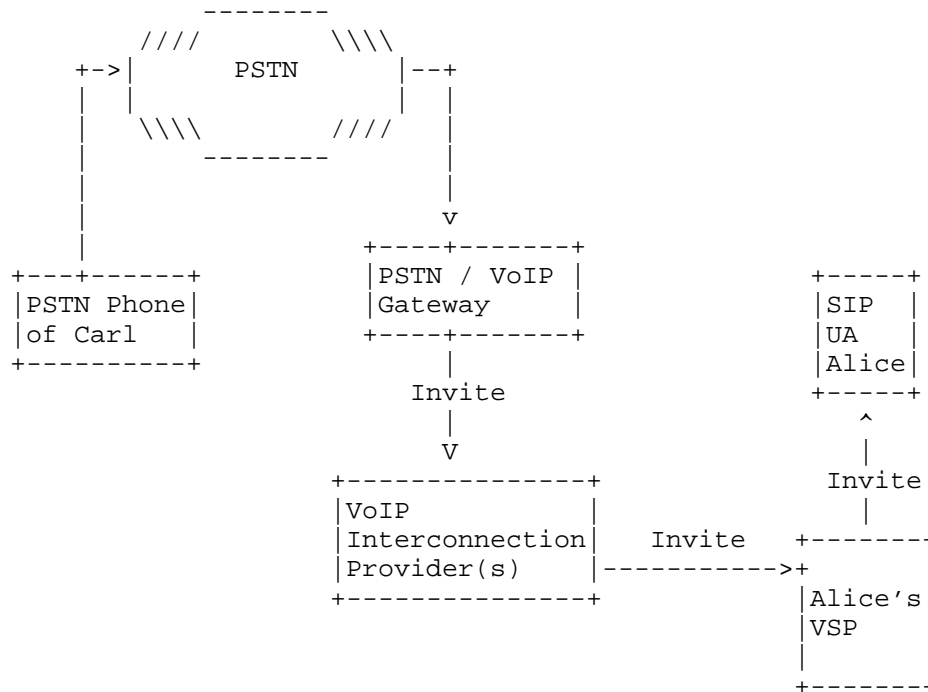


Figure 3: PSTN-to-VoIP Call.

4.4. VoIP-to-PSTN Call

Consider Figure 4 where Alice calls Carl. Carl uses a PSTN phone and Alice an IP-based phone. When Alice initiates the call, the E.164 number is get translated to a SIP URI and subsequently to an IP address. The call of Alice traverses her VoIP provider where the call origin identification information is added. It then hits the PSTN/VoIP gateway. It is desirable that the gateway verify that Alice can claim the E.164 number she is using before it populates the corresponding calling party number field in telephone network signaling. Carl's phone must be able to verify that it is receiving a legitimate call from the calling party number it will render to Carl.

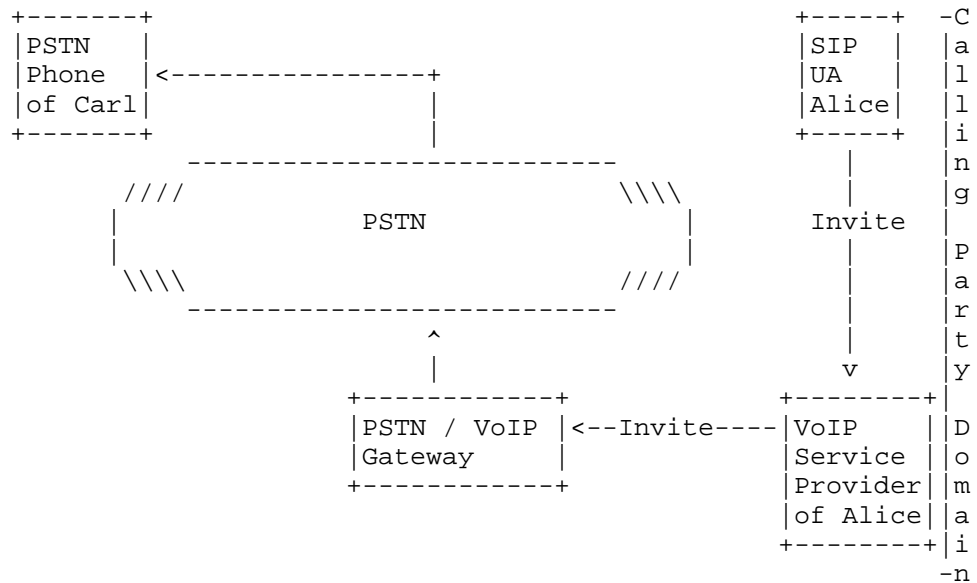


Figure 4: IP-to-PSTN Call.

4.5. PSTN-VoIP-PSTN Call

Consider Figure 5 where Carl calls Alice. Both users have PSTN phones but interconnection between the two PSTN networks is accomplished via an IP network. Consequently, Carl's operator uses a PSTN-to-VoIP gateway to route the call via an IP network to a gateway to break out into the PSTN again.

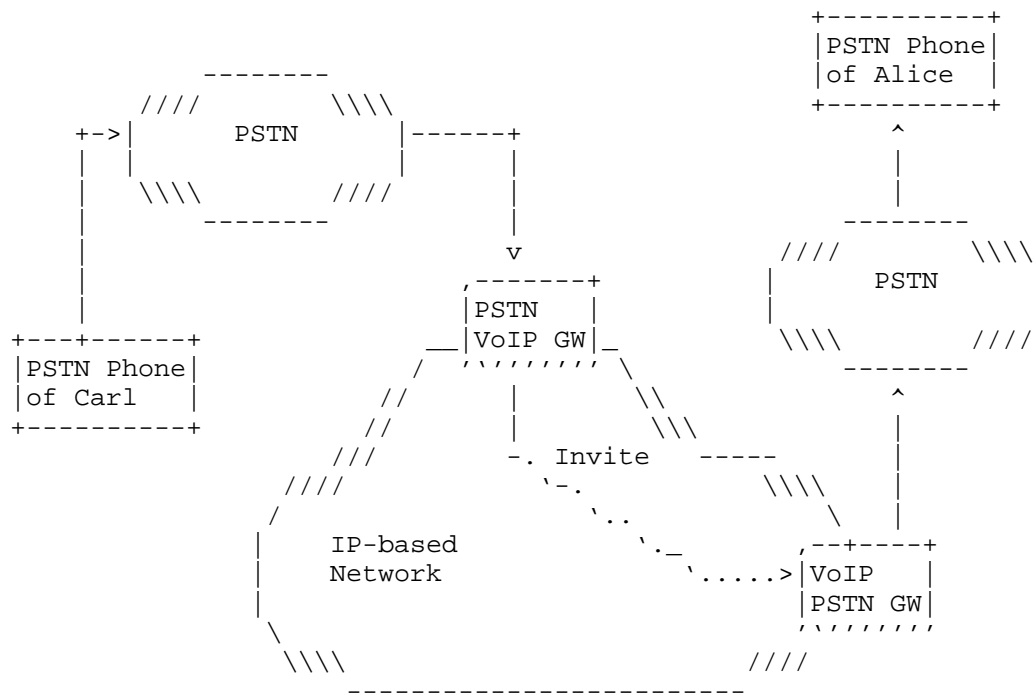


Figure 5: PSTN-VoIP-PSTN Call.

4.6. PSTN-to-PSTN Call

For the "legacy" case of a PSTN-to-PSTN call, otherwise beyond improvement, we may be able to use out-of-band IP connectivity at both the originating and terminating carrier to validate the call information.

5. Limitations of Current Solutions

From the inception of SIP, the From header field value has held an arbitrary user-supplied identity, much like the From header field value of an SMTP email message. During work on [RFC3261], efforts began to provide a secure origin for SIP requests as an extension to SIP. The so-called "short term" solution, the P-Asserted-Identity header described in [RFC3325], is deployed fairly widely, even though it is limited to closed trusted networks where end-user devices cannot alter or inspect SIP messages and offers no cryptographic validation. As P-Asserted-Identity is used increasingly across multiple networks, it cannot offer any protection against identity spoofing by intermediaries or entities that allow untrusted entities to set the P-Asserted-Identity information. An overview of

addressing spam in SIP, and explaining how it differs from similar problems with email, appeared in [RFC5039].

Subsequent efforts to prevent calling origin identity spoofing in SIP include the SIP Identity effort (the "long term" identity solution) [RFC4474] and Verification Involving PSTN Reachability (VIPR) [I-D.jennings-vipr-overview]. SIP Identity attaches a new header field to SIP requests containing a signature over the From header field value combined with other message components to prevent replay attacks. SIP Identity is meant to prevent both: (a) SIP UAs from originating calls with spoofed From headers; and (b) intermediaries, such as SIP proxies, from launching man-in-the-middle attacks by altering calls as they pass through the intermediaries. The VIPR architecture attacked a broader range of problems relating to spam, routing and identity with a new infrastructure for managing rendezvous and security, which operated alongside of SIP deployments.

As we will describe in more detail below, both SIP Identity and VIPR suffer from serious limitations that have prevented their deployment at significant scale, but they may still offer ideas and protocol building blocks for a solution.

5.1. P-Asserted-Identity

The P-Asserted-Identity header field of SIP [RFC3325] provides a way for trusted network entities to share with one another an authoritative identifier for the originator of a call. The value of P-Asserted-Identity cannot be populated by a user, though if a user wants to suggest an identity to the trusted network, a separate header (P-Preferred-Identity) enables them to do so. The features of the P-Asserted-Identity header evolved as part of a broader effort to reach parity with traditional telephone network signaling mechanisms for selectively sharing and restricting presentation of the calling party number at the user level, while still allowing core network elements to know the identity of the user for abuse prevention and accounting.

In order for P-Asserted-Identity to have these properties, it requires the existence of a trust domain as described in [RFC3324]. Any entity in the trust domain may add a P-Asserted-Identity header to a SIP message, and any entity in the trust domain may forward a message with a P-Asserted-Identity header to any other entity in the trust domain. If a trusted entity forwards a SIP request to an untrusted entity, however, the P-Asserted-Identity header must first be removed; most sorts of end user devices are outside trust domains. Sending a P-Asserted-Identity request to an untrusted entity could leak potentially private information, such as the network-asserted calling party number in a case where a caller has requested

presentation restriction. This concept of a trust domain is modeled on the trusted network of devices that operate the traditional telephone network.

P-Asserted-Identity has been very successful in telephone replacement deployments of SIP. It is an extremely simple in-band mechanism, requiring no cryptographic operations. Since it is so reminiscent of legacy mechanisms in the traditional telephone network, and it interworks so seamlessly with those protocols, it has naturally been favored by providers comfortable with these operating principles.

In practice, a trust domain exhibits many of the same merits and flaws as the traditional telephone network when it comes to securing a calling party number. Any trusted entity may provide P-Asserted-Identity, and a recipient of a SIP message has no direct assurance of who generated the P-Asserted-Identity header field value: all trust is transitive. Trust domains are dictated by business arrangements more than by security standards, and thus the level of assurance of P-Asserted-Identity is only as good as the least trustworthy member of a trust domain. Since the contents of P-Asserted-Identity are not intended for consumption by end users, end users must trust that their service provider participates in an appropriate trust domain, as there will be no direct evidence of the trust domain in SIP signaling that end user devices receive. Since the mechanism is so closely modeled on the traditional telephone network, it is unlikely to provide a higher level of security than that.

Since [RFC3325] was written, the whole notion of P- headers intended for use in private SIP domains has also been deprecated (see [RFC5727], largely because of overwhelming evidence that these headers were being used outside of private contexts and leaking into the public Internet. It is unclear how many deployments that make use of P-Asserted-Identity in fact conform with the Spec-T requirements of RFC3324.

P-Asserted-Identity also complicates the question of which URI should be presented to a user when a call is received. Per RFC3261, SIP user agents would render the contents of the From header field to a user when receiving an INVITE request, but what if the P-Asserted-Identity contains a more trustworthy URI, and presentation is not restricted? Subsequent proposals have suggested additional header fields to carry different forms of identity related to the caller, including billing identities. As the calling identities in a SIP request proliferate, the question of how to select one to render to the end user becomes more difficult to answer.

5.2. SIP Identity

The SIP Identity mechanism [RFC4474] provided two header fields for securing identity information in SIP requests: the Identity and Identity-Info header fields. Architecturally, the SIP Identity mechanism assumes a classic "SIP trapezoid" deployment in which an authentication service, acting on behalf of the originator of a SIP request, attaches identity information to the request which provides partial integrity protection; a verification service acting on behalf of the recipient validates the integrity of the request when it is received.

The Identity header field value contains a signature over a hash of selected elements of a SIP request, including several header field values (most significantly, the From header field value) and the entirety of the body of the request. The set of header field values was chosen specifically to prevent cut-and-paste attacks; it requires the verification service to retain some state to guard against replays. The signature over the body of a request has different properties for different SIP methods, but all prevent tampering by man-in-the-middle attacks. For a SIP MESSAGE request, for example, the signature over the body covers the actual message conveyed by the request: it is pointless to guarantee the source of a request if a man-in-the-middle can change the content of the message, as in that case the message content is created by an attacker. Similar threats exist against the SIP NOTIFY method. For a SIP INVITE request, a signature over the SDP body is intended to prevent a man-in-the-middle from changing properties of the media stream, including the IP address and port to which media should be sent, as this provides a means for the man-in-the-middle to direct session media to resource that the originator did not specify, and thus to impersonate an intended listener.

The Identity-Info header field value contains a URI designating the location of the certificate corresponding to the private key that signed the hash in the Identity header. That certificate could be passed by-value along with the SIP request, in which case a "cid" URI appears in Identity-Info, or by-reference, for example when the Identity-Info header field value has the URL of a service that delivers the certificate. [RFC4474] imposes further constraints governing the subject of that certificate: namely, that it must cover the domain name indicated in the domain component of the URI in the From header field value of the request.

The SIP Identity mechanism, however, has two fundamental limitations that have precluded its deployment: first, that it provides Identity only for domain names rather than other identifiers; second, that it

does not tolerate intermediaries that alter the bodies, or certain header fields, of SIP requests.

As deployed, SIP predominantly mimics the structures of the telephone network, and thus uses telephone numbers as identifiers. Telephone numbers in the From header field value of a SIP request may appear as the user part of a SIP URI, or alternatively in an independent tel URI. The certificate designated by the Identity-Info header field as specified, however, corresponds only to the domain portion of a SIP URI in the From header field. As such, [RFC4474] does not have any provision to identify the assignee of a telephone number. While it could be the case that the domain name portion of a SIP URI signifies a carrier (like "att.com") to whom numbers are assigned, the SIP Identity mechanism provides no assurance that a number is assigned to any carrier. For a tel URI, moreover, it is unclear in [RFC4474] what entity should hold a corresponding certificate. A caller may not want to reveal the identity of its service provider to the callee, and may thus prefer tel URIs in the From header field.

This lack of authority gives rise to a wholeclass of SIP identity problems when dealing with telephone numbers, as is explored in [I-D.rosenberg-sip-rfc4474-concerns]. That document shows how the Identity header of a SIP request targeting a telephone number (embedded in a SIP URI) could be dropped by an intermediate domain, which then modifies and re-signs the request, all without alerting the verification service: the verification service has no way of knowing which original domain signed the request. Provided that the local authentication service is complicit, an originator can claim virtually any telephone number, impersonating any chosen Caller ID from the perspective of the verifier. Both of these attacks are rooted in the inability of the verification service to ascertain a specific certificate that is authoritative for a telephone number.

As deployed, SIP is moreover highly mediated, and mediated in ways that [RFC3261] did not anticipate. As request routing commonly depends on policies dissimilar to [RFC3263], requests transit multiple intermediate domains to reach a destination; some forms of intermediaries in those domains may effectively re-initiate the session.

One of the main reasons that SIP deployments mimic the PSTN architecture is because the requirement for interconnection with the PSTN remains paramount: a call may originate in SIP and terminate on the PSTN, or vice versa; and worse still, a PSTN-to-PSTN call may transit a SIP network in the middle, or vice versa. This necessarily reduces SIP's feature set to the least common denominator of the telephone network, and mandates support for telephone numbers as a primary calling identifier.

Interworking with non-SIP networks makes end-to-end identity problematic. When a PSTN gateway sends a call to a SIP network, it creates the INVITE request anew, regardless of whether a previous leg of the call originated in a SIP network that later dropped the call to the PSTN. As these gateways are not necessarily operated by entities that have any relationship to the number assignee, it is unclear how they could provide an identity signature that a verifier should trust. Moreover, how could the gateway know that the calling party number it receives from the PSTN is actually authentic? And when a gateway receives a call via SIP and terminates a call to the PSTN, how can that gateway verify that a telephone number in the From header field value is authentic, before it presents that number as the calling party number in the PSTN?

Similarly, some SIP networks deploy intermediaries that act as back-to-back user agents (B2BUAs), typically in order to provide policy or interworking functions at network boundaries (hence the nickname "Session Border Controller"). These functions range from topology hiding, to alterations necessary to interoperate successfully with particular SIP implementations, to simple network address translation from private address space. To achieve these aims, these entities modify SIP INVITE requests in transit, potentially changing the From, Contact and Call-ID header field values, as well as aspects of the SDP, including especially the IP addresses and ports associated with media. Consequently, a SIP request exiting a B2BUA has no necessary relationship to the original request received by the B2BUA, much like a request exiting a PSTN gateway has no necessary relationship to any SIP request in a pre-PSTN leg of the call. An Identity signature provided for the original INVITE has no bearing on the post-B2BUA INVITE, and, were the B2BUA to preserve the original Identity header, any verification service would detect a violation of the integrity protection.

The SIP community has long been aware of these problems with [RFC4474] in practical deployments. Some have therefore proposed weakening the security constraints of [RFC4474] so that at least some deployments of B2BUAs will be compatible with integrity protection of SIP requests. However, such solutions do not address one key problem identified above: the lack of any clear authority for telephone numbers, and the fact that some INVITE requests are generated by intermediaries rather than endpoints. Removing the signature over the SDP from the Identity header will not, for example, make it any clearer how a PSTN gateway should assert identity in an INVITE request.

5.3. VIPR

Verification Involving PSTN Reachability (VIPR) directly attacks the twin problems of identifying number assignees on the Internet and coping with intermediaries that may modify signaling. To address the first problem, VIPR relies on the PSTN itself: it discovers which endpoints on the Internet are reachable via a particular PSTN number by calling the number on the PSTN to determine whom a call to that number will reach. As VIPR-enabled Internet endpoints associated with PSTN numbers are discovered, VIPR provides a rendez-vous service that allows the endpoints of a call to form an out-of-band connection over the Internet; this connection allows the endpoints to exchange information that secures future communications and permits direct, unmediated SIP connections.

VIPR provides these services within a fairly narrow scope of applicability. Its seminal use case is the enterprise IP PBX, a device that has both PSTN connectivity and Internet connectivity, which serves a set of local users with telephone numbers; after a PSTN call has connected successfully and then ended, the PBX searches a distributed hash-table to see if any VIPR-compatible devices have advertised themselves as a route for the unfamiliar number on the Internet. If advertisements exist, the originating PBX then initiates a verification process to determine whether the entity claiming to be the assignee of the unfamiliar number in fact received the successful call: this involves verifying details such as the start and stop times of the call. If the destination verifies successfully, the originating PBX provisions a local database with a route for that telephone number to the URI provided by the proven destination. The destination moreover gives a token to the originator that can be inserted in future call setup messages to authenticate the source of future communications.

Through this mechanism, the VIPR system provides a suite of properties, ones that go well beyond merely securing the origins of communications. It also provides a routing system which dynamically discovers mappings between telephone numbers and URIs, effectively building an ad hoc ENUM database in every VIPR implementation. The tokens exchanged over the out-of-band connection established by VIPR moreover provide an authorization mechanism for accepting calls over the Internet that significantly reduces the potential for spam. Because the token can act as a cookie due to the presence of this out-of-band connectivity, the VIPR token is less susceptible to cut-and-paste attacks and thus needs to cover with its signature far less of a SIP request.

Due to its narrow scope of applicability, and the details of its implementation, VIPR has some significant limitations. The most

salient for the purposes of this document is that it only has bearing on repeated communications between entities: it has no solution to the classic "robocall" problem, where the target typically receives a call from a number that has never called before. All of VIPR's strengths in establishing identity and spam prevention kick in only after an initial PSTN call has been completed, and subsequent attempts at communication begin. Every VIPR-compliant entity moreover maintains its own stateful database of previous contacts and authorizations, which lends itself to more aggregators like IP PBXs that may front for thousands of users than to individual phones. That database must be refreshed by periodic PSTN calls to determine that control over the number has not shifted to some other entity; figuring out when data has grown stale is one the challenges of the architecture. As VIPR requires compliant implementations to operate both a PSTN interface and an IP interface, it has little apparent applicability to ordinary desktop PCs or similar devices with no ability to place direct PSTN calls.

The distributed hash table also creates a new attack surface for impersonation. Attackers who want to pose as the owners of telephone numbers can advertise themselves as routes to a number in the hash table. VIPR has no inherent restriction on the number of entities that may advertise themselves as routes for a number, and thus an originator may find multiple advertisements for a number on the DHT even when an attack is not in progress. As for attackers, even if they cannot successfully verify themselves to the originators of calls (because they lack the call detail information), they may learn from those verification attempts which VIPR entities recently placed calls to the target number: it may be that this information is all the attacker hopes to glean. The fact that advertisements and verifications are public results from the public nature of the DHT that VIPR creates. The public DHT prevents any centralized control, or attempts to impede communications, but those come at the cost of apparently unavoidable privacy losses.

Because of these limitations, VIPR, much like SIP Identity, has had little impact in the marketplace. Ultimately, VIPR's utility as an identity mechanism is limited by its reliance on the PSTN, especially its need for an initial PSTN call to complete before any of VIPR's benefits can be realized, and by the drawbacks of the highly-public exchanges requires to create the out-of-band connection between VIPR entities. As such, there is no obvious solution to providing secure origin services for SIP on the Internet today.

6. Environmental Changes

6.1. Shift to Mobile Communication

In the years since [RFC4474] was conceived, there have been a number of fundamental shifts in the communications marketplace. The most transformative has been the precipitous rise of mobile smart phones, which are now arguably the dominant communications device in the developed world. Smart phones have both a PSTN and an IP interface, as well as an SMS and MMS capabilities. This suite of tools suggests that some of the techniques proposed by VIPR could be adapted to the smart phone environment. The installed base of smart phones is moreover highly upgradable, and permits rapid adoption of out-of-band rendezvous services for smart phones that circumvent the PSTN. Mobile messaging services that use telephone numbers as identities allow smart phone users to send text messages to one another over the Internet rather than over the PSTN. Like VIPR, such services create an out-of-band connection over the Internet between smart phones; unlike VIPR, the rendezvous service is provided by a trusted centralized database rather than by a DHT, and it is the centralized database that effectively verifies and asserts the telephone number of the sender of a message. While such messaging services are specific to the users of the specific service, it seems clear that similar databases could be provided by neutral third parties in a position to coordinate between endpoints.

6.2. Failure of Public ENUM

At the time [RFC4474] was written, the hopes for establishing a certificate authority for telephone numbers on the Internet largely rested on public ENUM deployment. The e164.arpa DNS tree established for ENUM could have grown to include certificates for telephone numbers or at least for number ranges. It is now clear however that public ENUM as originally envisioned has little prospect for adoption. That said, some national authorities for telephone numbers are migrating their provisioning services to the Internet, and issuing credentials that express authority for telephone numbers to secure those services. These new authorities for numbers could provide to the public Internet the necessary signatory authority for securing calling parties' numbers. While these systems are far from universal, the authors of this draft believe that a solution devised for the North American Numbering Plan could have applicability to other country codes.

6.3. Public Key Infrastructure Developments

Also, there have been a number of recent high-profile compromises of web certificate authorities. The presence of numerous (in some cases, of hundreds) of trusted certificate authorities in modern web browsers has become a significant security liability. As [RFC4474] relied on web certificate authorities, this too provides new lessons for any work on revising [RFC4474]: namely, that innovations like DANE [RFC6698] that designate a specific certificate preferred by the owner of a DNS name could greatly improve the security of a SIP identity mechanism; and moreover, that when considering new certificate authorities for telephone numbers, we should be wary of excessive pluralism. While a chain of delegation with a progressively narrowing scope of authority (e.g., from a regulatory entity to a carrier to a reseller to an end user) is needed to reflect operational practices, there is no need to have multiple roots, or peer entities that both claim authority for the same telephone number or number range.

6.4. Prevalence of B2BUA Deployments

Given the prevalence of established B2BUA deployments, we may have a further opportunity to review the elements signed by [RFC4474] and to decide on the value of alternative signature mechanisms. Separating the elements necessary for (a) securing the From header field value and preventing replays, from (b) the elements necessary to prevent men-in-the-middle from tampering with messages, may also yield a strategy for identity that will be practicable in some highly mediated networks. Solutions in this space must however remain mindful of the requirements for securing cryptographic material necessary to support DTLS-SRTP or future security mechanisms.

6.5. Stickiness of Deployed Infrastructure

One thing that has not changed, and is not likely to change in the future, is the transitive nature of trust in the PSTN. When a call from the PSTN arrives at a SIP gateway with a calling party number, the gateway will have little chance of determining whether the originator of the call was authorized to claim that calling party number. Due to roaming and countless other factors, calls on the PSTN may emerge from administrative domains that were not assigned the originating number. This use case will remain the most difficult to tackle for an identity system, and may prove beyond repair. It does however seem that with the changes in the solution space, and a better understanding of the limits of [RFC4474] and VIPR, we are today in a position to reexamine the problem space and find solutions that can have a significant impact on the secure origins problem.

6.6. Concerns about Pervasive Monitoring

While spoofing the origins of communication is a source of numerous security concerns, solutions for identifying communications must also be mindful of the security risks of pervasive monitoring (see [I-D.farrell-perpass-attack]). Identifying information, once it is attached to communications, can potentially be inspected by parties other than the intended recipient and collected for any number of reasons. As stated above, the purpose of this work is not to eliminate anonymity, but furthermore, to be viable and in the public interest, solutions should not facilitate the unauthorized collection of calling data.

6.7. Relationship with Number Assignment and Management

Currently, telephone numbers are typically managed in a loose delegation hierarchy. For example, a national regulatory agency may task a private, neutral entity with administering numbering resources, such as area codes, and a similar entity with assigning number blocks to carriers and other authorized entities, who in turn then assign numbers to customers. Resellers with looser regulatory obligations can complicate the picture, and in many cases it is difficult to distinguish the roles of enterprises from carriers. In many countries, individual numbers are portable between carriers, at least within the same technology (e.g., wireline-to-wireline). Separate databases manage the mapping of numbers to switch identifiers, companies and textual caller ID information.

As the PSTN transitions to using VoIP technologies, new assignment policies and management mechanisms are likely to emerge. For example, it has been proposed that geography could play a smaller role in number assignments, and that individual numbers are assigned to end users directly rather than only to service providers, or that the assignment of numbers does not depend on providing actual call delivery services.

Databases today already map telephone numbers to entities that have been assigned the number, e.g., through the LERG (originally, Local Exchange Routing Guide) in the United States. Thus, the transition to IP-based networks may offer an opportunity to integrate cryptographic bindings between numbers or number ranges and service providers into databases.

7. Basic Requirements

This section describes only the high level requirements of the effort, which we expected will be further articulated as work continues:

Generation: Intermediaries as well as end system must be able to generate the source identity information.

Validation: Intermediaries as well as end system must be able to validate the source identity information.

Usability: Any validation mechanism must work without human intervention, that is, without for example CAPTCHA-like mechanisms.

Deployability: Must survive transition of the call to the PSTN and the presence of B2BUAs.

Reflecting existing authority: Must stage credentials on existing national-level number delegations, without assuming the need for an international golden root on the Internet.

Accommodating current practices: Must allow number portability among carriers and must support legitimate usage of number spoofing (doctor's office and call centers)

Minimal payload overhead: Must lead to minimal expansion of SIP headers fields to avoid fragmentation in deployments that use UDP.

Efficiency: Must minimize RTTs for any network lookups and minimize any necessary cryptographic operations.

Privacy: A solution must prevent unauthorized third parties from learning what numbers have been called by a specific caller.

Some requirements specifically outside the scope of the effort include:

Display name: This effort does not consider how the display name of the caller might be validated.

Response authentication: This effort only considers the problem of providing secure telephone identity for requests, not for responses to requests; no solution is here proposed for the problem of determining to which number a call has connected.

8. Acknowledgments

We would like to thank Sanjay Mishra, Fernando Mousinho, David Frankel, Penn Pfautz, Mike Hammer, Dan York, Andrew Allen, Philippe Fouquart, Hadriel Kaplan, Richard Shockey, Russ Housley, Alissa Cooper, Bernard Aboba, Sean Turner, Brian Rosen, Eric Burger, and Eric Rescorla for their discussion input that lead to this document.

9. IANA Considerations

This memo includes no request to IANA.

10. Security Considerations

This document is about improving the security of call origin identification; security considerations for specific solutions will be discussed in solutions documents.

11. Informative References

[I-D.cooper-iab-secure-origin]

Cooper, A., Tschofenig, H., Peterson, J., and B. Aboba, "Secure Call Origin Identification", draft-cooper-iab-secure-origin-00 (work in progress), November 2012.

[I-D.farrell-perpass-attack]

Farrell, S. and H. Tschofenig, "Pervasive Monitoring is an Attack", draft-farrell-perpass-attack-06 (work in progress), February 2014.

[I-D.jennings-vipr-overview]

Barnes, M., Jennings, C., Rosenberg, J., and M. Petit-Huguenin, "Verification Involving PSTN Reachability: Requirements and Architecture Overview", draft-jennings-vipr-overview-06 (work in progress), December 2013.

[I-D.peterson-sipping-retarget]

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

[I-D.rosenberg-sip-rfc4474-concerns]

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

[RFC3261] 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.

[RFC3263] Rosenberg, J. and H. Schulzrinne, "Session Initiation Protocol (SIP): Locating SIP Servers", RFC 3263, June 2002.

- [RFC3324] Watson, M., "Short Term Requirements for Network Asserted Identity", RFC 3324, November 2002.
- [RFC3325] 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.
- [RFC3966] Schulzrinne, H., "The tel URI for Telephone Numbers", RFC 3966, December 2004.
- [RFC4474] Peterson, J. and C. Jennings, "Enhancements for Authenticated Identity Management in the Session Initiation Protocol (SIP)", RFC 4474, August 2006.
- [RFC4916] Elwell, J., "Connected Identity in the Session Initiation Protocol (SIP)", RFC 4916, June 2007.
- [RFC5039] Rosenberg, J. and C. Jennings, "The Session Initiation Protocol (SIP) and Spam", RFC 5039, January 2008.
- [RFC5727] Peterson, J., Jennings, C., and R. Sparks, "Change Process for the Session Initiation Protocol (SIP) and the Real-time Applications and Infrastructure Area", BCP 67, RFC 5727, March 2010.
- [RFC6698] Hoffman, P. and J. Schlyter, "The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA", RFC 6698, August 2012.
- [TDOS] Krebs, B., "DHS Warns of 'TDoS' Extortion Attacks on Public Emergency Networks", URL: <http://krebsonsecurity.com/2013/04/dhs-warns-of-tdos-extortion-attacks-on-public-emergency-networks/>, Apr 2013.
- [news-hack] Wikipedia, , "News International phone hacking scandal", URL: http://en.wikipedia.org/wiki/News_International_phone_hacking_scandal, Apr 2013.
- [robocall-competition] FTC, , "FTC Robocall Challenge", URL: <http://robocall.challenge.gov/>, Apr 2013.
- [robocall-fcc] FCC, , "Robocalls", URL: <http://www.fcc.gov/guides/robocalls>, Apr 2013.

[swatting]

Wikipedia, , "Don't Make the Call: The New Phenomenon of
'Swatting'", URL: [http://www.fbi.gov/news/stories/2008/
february/swatting020408](http://www.fbi.gov/news/stories/2008/february/swatting020408), Feb 2008.

Authors' Addresses

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

Email: jon.peterson@neustar.biz

Henning Schulzrinne
Columbia University
Department of Computer Science
450 Computer Science Building
New York, NY 10027
US

Phone: +1 212 939 7004
Email: hgs@cs.columbia.edu
URI: <http://www.cs.columbia.edu>

Hannes Tschofenig
Hall, Tirol 6060
Austria

Email: Hannes.Tschofenig@gmx.net
URI: <http://www.tschofenig.priv.at>

Network Working Group
Internet-Draft
Intended status: Informational
Expires: April 15, 2014

J. Peterson
NeuStar, Inc.
October 12, 2013

Secure Telephone Identity Threat Model
draft-ietf-stir-threats-00.txt

Abstract

As the Internet and the telephone network have become increasingly interconnected and interdependent, attackers can impersonate or obscure calling party numbers when orchestrating bulk commercial calling schemes, hacking voicemail boxes or even circumventing multi-factor authentication systems trusted by banks. This document analyzes threats in the resulting system, enumerating actors, reviewing the capabilities available to and used by attackers, and describing scenarios in which attacks are launched.

Status of This Memo

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

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

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

This Internet-Draft will expire on April 15, 2014.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must

include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction and Scope	2
2. Actors	3
2.1. Endpoints	4
2.2. Intermediaries	4
2.3. Attackers	5
3. Attacks	6
3.1. Voicemail Hacking via Impersonation	6
3.2. Unsolicited Commercial Calling from Impersonated Numbers	7
4. Attack Scenarios	8
4.1. TBD: Solution-Specific Attacks	8
5. Acknowledgments	9
6. IANA Considerations	9
7. Security Considerations	9
8. Informative References	9
Author's Address	11

1. Introduction and Scope

As is discussed in the STIR problem statement [9], the primary enabler of robocalling, vishing, swatting and related attacks is the capability to impersonate a calling party number. The starkest example of these attacks are cases where automated callees on the PSTN rely on the calling number as a security measure, for example to access a voicemail system. Robocallers use impersonation as a means of obscuring identity; while robocallers can, in the ordinary PSTN, block (that is, withhold) their caller identity, callees are less likely to pick up calls from blocked identities, and therefore calling from some number, any number, is preferable. Robocallers however prefer not to call from a number that can trace back to the robocaller, and therefore they impersonate numbers that are not assigned to them.

The scope of impersonation in this threat model pertains solely to the rendering of a calling telephone number to a callee (human user or automaton) at the time of call set-up. The primary attack vector is therefore one where the attacker contrives for the calling telephone number in signaling to be a specific number. In this attack, the number is one that the attacker is not authorized to use (as a caller), but gives in order for that number to be consumed or rendered on the terminating side. The threat model assumes that this attack simply cannot be prevented: there is no way to stop the attacker from creating calls that contain attacker-chosen calling

telephone numbers. The solution space therefore focuses on ways that terminating or intermediary elements might differentiate authorized from unauthorized calling party numbers, in order that policies, human or automatic, might act on that information.

Securing an authenticated calling party number at call set-up time does not entail anything about the entity or entities that will send and receive media during the call itself. In call paths with intermediaries and gateways (as described below), there may be no way to provide any assurance in the signaling about participants in the media of a call. In those end-to-end IP environments where such an assurance is possible, it is highly desirable. However, in the threat model described in this document, "impersonation" does not consider impersonating an authorized listener after a call has been established, such as a third party attempting to eavesdrop on a conversation. Attackers that could impersonate an authorized listener require capabilities that robocallers and voicemail hackers are unlikely to possess, and historically such attacks have not played a role in enabling robocalling or related problems.

In SIP and even many traditional telephone protocols, call signaling can be renegotiated after the call has been established. Using various transfer mechanisms common in telephone systems, a callee can easily be connected to, or conferenced in with, telephone numbers other than the original calling number once a call has been established. These post-setup changes to the call are outside the scope of impersonation considered in this model. Furthermore, impersonating a reached number to the originator of a call is outside the scope of this threat model.

In much of the PSTN, there exists a supplemental service that translates calling party numbers into regular names, including the proper names of people and businesses, for rendering to the called user. These services (frequently termed 'Caller ID') provide a further attack surface for impersonation. The threat model described in this document addresses only the calling party number, even though presenting a forged calling party number may cause a forged 'Caller ID' name to be rendered to the user as well. Providing a verifiable calling party number therefore improve the security of Caller ID systems, but this threat model does not consider attacks specific to Caller ID. Such attacks may be carried out against the databases consulted by the terminating side of a call to provide Caller ID, or by impersonators forging a particular calling party number in order to present a misleading Caller ID to the user.

2. Actors

2.1. Endpoints

There are two main categories of end-user terminals relevant to this discussion, a dumb device (such as a 'black phone') or a smart device:

Dumb devices comprise a simple dial pad, handset and ringer, optionally accompanied by a display that can render a limited number of characters (typically, enough for a telephone number and an accompanying name, sometimes less). Although users interface with these devices, the intelligence that drives them lives in the service provider network.

Smart devices are general purpose computers with some degree of programmability, and with the capacity to access the Internet and to render text, audio and/or images. This category includes smart phones, telephone applications on desktop and laptop computers, IP private branch exchanges, and so on.

There is a further category of automated terminals without an end user. These include systems like voicemail services, which may provide a different set of services to a caller based solely on the calling party's number, granting the mailbox owner access to a menu while giving other callers only the ability to leave a message. Though the capability of voicemail services varies widely, many today have Internet access and advanced application interfaces (to render 'visual voicemail,' to automatically transcribe voicemail to email, and so on).

There is a further category of automated terminals without an end user. These include systems like voicemail services that consume the calling party number without rendering it to a human. Though the capability of voicemail services varies widely, many today have Internet access and advanced application interfaces (to render 'visual voicemail,' to automatically transcribe voicemail to email, and so on).

2.2. Intermediaries

The endpoints of a traditional telephone call connect through numerous intermediary switches in the network. The set of intermediary devices traversed during call setup between two endpoints is referred to as a call path. The length of the call path can vary considerably: it is possible in VoIP deployments for two endpoint entities to send traffic to one another directly, but, more commonly, several intermediaries exist in a VoIP call path. One or more gateways may also appear on a call path.

Intermediaries forward call signaling to the next entity in the path. These intermediaries may also modify the signaling in order to improve interoperability, to enable proper network-layer media connections, or to enforce operator policy. This threat model assumes there are no restrictions on the modifications to signaling that an intermediary can introduce (which is consistent with the observed behavior of such devices).

Gateways translate call signaling from one protocol into another. In the process, they tend to consume any signaling specific of the original protocol (elements like transaction-matching identifiers) and may need to transcode or otherwise alter identifiers as they are rendered in the destination protocol.

This threat model assumes that intermediaries and gateways can forward and retarget calls as necessary, which can result in a call terminating at a place the originator did not expect; this is a common condition in call routing. This is significant to the solution space, because it limits the ability of the originator to anticipate what the telephone number of the respondent will be (for more on the "unanticipated respondent" problem, see [10]).

Furthermore, we assume that some intermediaries or gateways may, due to their capabilities or policies, discard calling party number information, in whole or part. Today, many IP-PSTN gateways simply ignore any information available about the caller in the IP leg of the call, and allow the telephone number of the PRI line used by the gateway to be sent as the calling party number for the PSTN leg of the call. A call might also gateway to a multifrequency network where only a limited number of digits of automatic numbering identification (ANI) data are signaled, for example. Some protocols may render telephone numbers in a way that makes it impossible for a terminating side to parse or canonicalize a number. In these cases, providing authenticated identity may be impossible. This is not however indicative of an attack or other security failure.

2.3. Attackers

We assume that an attacker has the following capabilities:

An attacker can create telephone calls at will, originating them either on the PSTN or over IP, and can supply an arbitrary calling party number.

An attacker can capture and replay signaling previously observed by it. [TBD: should this include an attacker that can capture signaling that isn't directly sent to it? Not a factor for robocalling, but perhaps for voicemail hacking, say.]

An attacker has access to the Internet, and thus the ability to inject arbitrary traffic over the Internet, to access public directories, and so on.

There are attack scenarios in which an attacker compromises intermediaries in the call path, or captures credentials that allow the attacker to impersonate a target. Those system-level attacks are not considered in this threat model, though secure design and operation of systems to prevent these sorts of attacks is necessary for envisioned countermeasures to work.

This threat model also does not consider scenarios in which the operators of intermediaries or gateways are themselves adversaries who intentionally discard valid identity information (without a user requesting anonymity) or who send falsified identity using their own credentials. The design of the credential system will however limit the scope of the credentials issued to carriers or national authorities to those numbers that fall under their purview.

3. Attacks

3.1. Voicemail Hacking via Impersonation

A voicemail service allows users calling from their mobile phones access to their voicemail boxes on the basis of the calling party number. If an attacker wants to access the voicemail of a particular target, the attacker may try to impersonate the calling party number using one of the scenarios described below.

The envisioned countermeasures for this attack involve the voicemail treating calls that supply an authenticated identity differently from other calls. In the absence of identity, for example, a voicemail service might enforce some other caller authentication policy (perhaps requiring a PIN for caller authentication). Authenticated identity alone provides a positive confirmation only when an identity is claimed legitimately; the absence of authenticated identity here may not be evidence of malice, just of uncertainty.

If the voicemail service could learn ahead of time that it should expect authenticated identity from a particular number, that would enable the voicemail service to adopt stricter policies for handling a request without authenticated identity. Since users contact a voicemail service repeatedly, the service could for example remember which users usually sign their requests and require further authentication mechanisms when signatures are absent. Alternatively, issuers of credentials or other authorities could provide a service that informs verifiers that they should expect identity signatures in calls from particular numbers.

3.2. Unsolicited Commercial Calling from Impersonated Numbers

The unsolicited commercial calling, or for short robocalling, attack is similar to the voicemail attack, except that the robocaller does not need to impersonate the particular number controlled by the target, merely some "plausible" number. A robocaller may impersonate a number that is not an assignable number (for example, in the United States, a number beginning with 0), or an unassigned number. A robocaller may change numbers every time a new call is placed, even selecting numbers randomly.

A closely related attack is sending unsolicited bulk commercial messages via text messaging services. Almost always, these messages originate on the Internet, though they may ultimately reach endpoints over traditional telephone network protocols or the Internet. While most text messaging endpoints are mobile phones, increasingly broadband residential services support text messaging as well. The originators of these messages typically impersonate a calling party number, in some cases a "short code" specific to text messaging services.

The envisioned countermeasures to robocalling are similar to those in the voicemail example, but there are significant differences. One important potential countermeasure is simply to verify that the calling party number is in fact assignable and assigned. Unlike voicemail services, end users typically have never been contacted by the number used by a robocaller before. Thus they can't rely on past association to anticipate whether or not the calling party number should supply authenticated identity. If there were a service that could inform the terminating side of that it should expect an identity signature in calls or texts from that number, however, that would also help in the robocalling case.

When a human callee is to be alerted at call setup time, the time frame for executing any countermeasures is necessarily limited. Ideally, a user would not be alerted that a call has been received until any necessary identity checks have been performed. This could however result in inordinate post-dial delay from the perspective of legitimate callers. Cryptographic operations and network operations must be minimized for these countermeasures to be practical. For text messages, a delay for executing anti-impersonation countermeasures is much less likely to degrade perceptible service.

The eventual effect of these countermeasures would be to force robocallers to either block their caller identity, in which case end users could opt not to receive their calls or messages, or to force robocallers to use authenticated identity for numbers traceable to them, which would then allow for other forms of redress.

4. Attack Scenarios

Impersonation, IP-PSTN

An attacker on the Internet uses a commercial WebRTC service to send a call to the PSTN with a chosen calling party number. The service contacts an Internet-to-PSTN gateway, which inserts the attacker's chosen calling party number into the CPN field of an IAM. When the IAM reaches the terminating telephone switch, the terminal renders the attacker's chosen calling party number as the calling identity.

Impersonation, PSTN-PSTN

An attacker with a traditional PBX (connected to the PSTN through ISDN) sends a Q.931 SETUP request with a chosen calling party number which a service provider inserts into the corresponding SS7 calling party number (CPN) field of a call setup message (IAM). When the IAM reaches the endpoint switch, the terminal renders the attacker's chosen calling party number as the calling identity.

Impersonation, IP-IP

An attacker with an IP phone sends a SIP request to an IP-enabled voicemail service. The attacker puts a chosen calling party number into the From header field value of the INVITE. When the INVITE reaches the endpoint terminal, the terminal renders the attacker's chosen calling party number as the calling identity.

Impersonation, IP-PSTN-IP

An attacker with an IP phone sends a SIP request to the telephone number of a voicemail service, perhaps without even knowing that the voicemail service is IP-based. The attacker puts a chosen calling party number into the From header field value of the INVITE. The attacker's INVITE reaches an Internet-to-PSTN gateway, which inserts the attacker's chosen calling party number into the CPN of an IAM. That IAM then traverses the PSTN until (perhaps after a call forwarding) it reaches another gateway, this time back to the IP realm, to an H.323 network. The PSTN-IP gateway puts takes the calling party number in the IAM CPN field and puts it into the SETUP request. When the SETUP reaches the endpoint terminal, the terminal renders the attacker's chosen calling party number as the calling identity.

4.1. TBD: Solution-Specific Attacks

[TBD: This is just forward-looking notes]

Attacks Against In-band

- Token replay

- Removal of in-band signaling features

Attacks Against Out-of-Band

- Provisioning Gargbage CPRs

- Data Mining

Attacks Against Either Approach

- Attack on directories/services that say whether you should expect authenticated identity or not

- Canonicalization attack

5. Acknowledgments

Stephen Kent, Brian Rosen, Alex Bobotek, Henning Schulzrinne, Hannes Tschofenig, Cullen Jennings and Eric Rescorla provided key input to the discussions leading to this document.

6. IANA Considerations

This memo includes no request to IANA.

7. Security Considerations

This document provides a threat model and is thus entirely about security.

8. Informative References

- [1] Peterson, J. and C. Jennings, "Enhancements for Authenticated Identity Management in the Session Initiation Protocol (SIP)", RFC 4474, August 2006.
- [2] 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.

- [3] 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.
- [4] 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.
- [5] Hoffman, P. and J. Schlyter, "The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA", RFC 6698, August 2012.
- [6] Elwell, J., "Connected Identity in the Session Initiation Protocol (SIP)", RFC 4916, June 2007.
- [7] Schulzrinne, H., "The tel URI for Telephone Numbers", RFC 3966, December 2004.
- [8] Cooper, A., Tschofenig, H., Peterson, J., and B. Aboba, "Secure Call Origin Identification", draft-cooper-iab-secure-origin-00 (work in progress), November 2012.
- [9] Peterson, J., Schulzrinne, H., and H. Tschofenig, "Secure Telephone Identity Problem Statement", draft-ietf-stir-problem-statement-00 (work in progress), October 2013.
- [10] Peterson, J., "Retargeting and Security in SIP: A Framework and Requirements", draft-peterson-sipping-retarget-00 (work in progress), February 2005.
- [11] Rosenberg, J., "Concerns around the Applicability of RFC 4474", draft-rosenberg-sip-rfc4474-concerns-00 (work in progress), February 2008.
- [12] Kaplan, H. and V. Pascual, "Loop Detection Mechanisms for Session Initiation Protocol (SIP) Back-to- Back User Agents (B2BUAs)", draft-ietf-straw-b2bua-loop-detection-02 (work in progress), September 2013.
- [13] Barnes, M., Jennings, C., Rosenberg, J., and M. Petit-Huguenin, "Verification Involving PSTN Reachability: Requirements and Architecture Overview", draft-jennings-vipr-overview-04 (work in progress), February 2013.

- [14] Rosenberg, J. and H. Schulzrinne, "Session Initiation Protocol (SIP): Locating SIP Servers", RFC 3263, June 2002.

Author's Address

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

Email: jon.peterson@neustar.biz

Network Working Group
Internet-Draft
Intended status: Informational
Expires: February 13, 2015

J. Peterson
NeuStar, Inc.
August 12, 2014

Secure Telephone Identity Threat Model
draft-ietf-stir-threats-04.txt

Abstract

As the Internet and the telephone network have become increasingly interconnected and interdependent, attackers can impersonate or obscure calling party numbers when orchestrating bulk commercial calling schemes, hacking voicemail boxes or even circumventing multi-factor authentication systems trusted by banks. This document analyzes threats in the resulting system, enumerating actors, reviewing the capabilities available to and used by attackers, and describing scenarios in which attacks are launched.

Status of This Memo

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

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

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

This Internet-Draft will expire on February 13, 2015.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must

include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction and Scope	2
2. Actors	4
2.1. Endpoints	4
2.2. Intermediaries	4
2.3. Attackers	5
3. Attacks	6
3.1. Voicemail Hacking via Impersonation	6
3.2. Unsolicited Commercial Calling from Impersonated Numbers	7
3.3. Telephony Denial-of-Service Attacks	8
4. Attack Scenarios	9
4.1. Solution-Specific Attacks	10
5. Acknowledgments	11
6. IANA Considerations	11
7. Security Considerations	11
8. Informative References	11
Author's Address	12

1. Introduction and Scope

As is discussed in the STIR problem statement [I-D.ietf-stir-problem-statement], the primary enabler of robocalling, vishing, swatting and related attacks is the capability to impersonate a calling party number. The starkest examples of these attacks are cases where automated callees on the PSTN rely on the calling number as a security measure, for example to access a voicemail system. Robocallers use impersonation as a means of obscuring identity; while robocallers can, in the ordinary PSTN, block (that is, withhold) their calling number from presentation, callees are less likely to pick up calls from blocked identities, and therefore appearing to calling from some number, any number, is preferable. Robocallers however prefer not to call from a number that can trace back to the robocaller, and therefore they impersonate numbers that are not assigned to them.

The scope of impersonation in this threat model pertains solely to the rendering of a calling telephone number to a callee (human user or automaton) at the time of call set-up. The primary attack vector is therefore one where the attacker contrives for the calling telephone number in signaling to be a chosen number. In this attack, the number is one that the attacker is not authorized to use (as a caller), but gives in order for that number to be consumed or rendered on the terminating side. The threat model assumes that this

attack simply cannot be prevented: there is no way to stop the attacker from creating call setup messages that contain attacker-chosen calling telephone numbers. The solution space therefore focuses on ways that terminating or intermediary elements might differentiate authorized from unauthorized calling party numbers, in order that policies, human or automatic, might act on that information.

Securing an authenticated calling party number at call set-up time does not entail any assertions about the entity or entities that will send and receive media during the call itself. In call paths with intermediaries and gateways (as described below), there may be no way to provide any assurance in the signaling about participants in the media of a call. In those end-to-end IP environments where such assurance is possible, it is highly desirable. However, in the threat model described in this document, "impersonation" does not consider impersonating an authorized listener after a call has been established (e.g., as a third party attempting to eavesdrop on a conversation). Attackers that could impersonate an authorized listener require capabilities that robocallers and voicemail hackers are unlikely to possess, and historically such attacks have not played a role in enabling robocalling or related problems.

In SIP and even many traditional telephone protocols, call signaling can be renegotiated after the call has been established. Using various transfer mechanisms common in telephone systems, a callee can easily be connected to, or conferenced in with, telephone numbers other than the original calling number once a call has been established. These post-setup changes to the call are outside the scope of impersonation considered in this model: the motivating use cases of defeating robocalling, voicemail hacking and swatting all rely on impersonation during the initial call setup. Furthermore, this threat model does not include in its scope the verification of the reached party's telephone number back to the originator of the call. There is no assurance to the originator that they are reaching the correct number, nor any indication when call forwarding has taken place. This threat model is focused only on verifying the calling party number to the callee.

In much of the PSTN, there exists a supplemental service that translates calling party numbers into names, including the proper names of people and businesses, for rendering to the called user. These services (frequently marketed as part of 'Caller ID') provide a further attack surface for impersonation. The threat model described in this document addresses only the calling party number, even though presenting a forged calling party number may cause a chosen calling party name to be rendered to the user as well. Providing a verifiable calling party number therefore improves the security of

calling party name systems, but this threat model does not consider attacks specific to names. Such attacks may be carried out against the databases consulted by the terminating side of a call to provide calling party names, or by impersonators forging a particular calling party number in order to present a misleading name to the user.

2. Actors

2.1. Endpoints

There are two main categories of end-user terminals relevant to this discussion, a dumb device (such as a 'black phone') or a smart device.

Dumb devices comprise a simple dial pad, handset and ringer, optionally accompanied by a display that can render a limited number of characters. Typically the display renders enough characters for a telephone number and an accompanying name, but sometimes fewer are rendered. Although users interface with these devices, the intelligence that drives them lives in the service provider network.

Smart devices are general purpose computers with some degree of programmability, and with the capacity to access the Internet and to render text, audio and/or images. This category includes smart phones, telephone applications on desktop and laptop computers, IP private branch exchanges, etc.

There is a further category of automated terminals without an end user. These include systems like voicemail services, which may provide a different set of services to a caller based solely on the calling party's number, for example granting the (purported) mailbox owner access to a menu while giving other callers only the ability to leave a message. Though the capability of voicemail services varies widely, many today have Internet access and advanced application interfaces (to render 'visual voicemail,' [refs.OMTP-VV] to automatically transcribe voicemail to email, etc.).

2.2. Intermediaries

The endpoints of a traditional telephone call connect through numerous intermediary devices in the network. The set of intermediary devices traversed during call setup between two endpoints is referred to as a call path. The length of the call path can vary considerably: it is possible in VoIP deployments for two endpoint entities to send traffic to one another directly, but, more commonly, several intermediaries exist in a VoIP call path. One or more gateways also may appear on a call path.

Intermediaries forward call signaling to the next device in the path. These intermediaries may also modify the signaling in order to improve interoperability, to enable proper network-layer media connections, or to enforce operator policy. This threat model assumes there are no restrictions on the modifications to signaling that an intermediary can introduce (which is consistent with the observed behavior of such devices).

A gateway is a subtype of intermediary that translates call signaling from one protocol into another. In the process, they tend to consume any signaling specific of the original protocol (elements like transaction-matching identifiers) and may need to transcode or otherwise alter identifiers as they are rendered in the destination protocol.

This threat model assumes that intermediaries and gateways can forward and retarget calls as necessary, which can result in a call terminating at a place the originator did not expect; this is a common condition in call routing. This observation is significant to the solution space, because it limits the ability of the originator to anticipate what the telephone number of the respondent will be (for more on the "unanticipated respondent" problem, see [I-D.peterson-sipping-retarget]).

Furthermore, we assume that some intermediaries or gateways may, due to their capabilities or policies, discard calling party number information, in whole or in part. Today, many IP-PSTN gateways simply ignore any information available about the caller in the IP leg of the call, and allow the telephone number of the PRI line used by the gateway to be sent as the calling party number for the PSTN leg of the call. For example, a call might also gateway to a multi-frequency network where only a limited number of digits of automatic numbering identification (ANI) data are signaled. Some protocols may render telephone numbers in a way that makes it impossible for a terminating side to parse or canonicalize a number. In these cases, providing authenticated calling number data may be impossible, but this is not indicative of an attack or other security failure.

2.3. Attackers

We assume that an attacker has the following capabilities:

An attacker can create telephone calls at will, originating them either on the PSTN or over IP, and can supply an arbitrary calling party number.

An attacker can capture and replay signaling previously observed by it.

An attacker has access to the Internet, and thus the ability to inject arbitrary traffic over the Internet, to access public directories, etc.

There are attack scenarios in which an attacker compromises intermediaries in the call path, or captures credentials that allow the attacker to impersonate a caller. Those system-level attacks are not considered in this threat model, though secure design and operation of systems to prevent these sorts of attacks are necessary for envisioned countermeasures to work. To date, robocallers and other impersonators do not resort to compromising systems, but rather exploit the intrinsic lack of secure identity in existing mechanisms: it is remedying this problem that lies within the scope of this threat model.

This threat model also does not consider scenarios in which the operators of intermediaries or gateways are themselves adversaries who intentionally discard valid identity information (without a user requesting anonymity) or who send falsified identity; see Section 4.1.

3. Attacks

The uses of impersonation described in this section are broadly divided into two categories: those where an attack will not succeed unless the attacker impersonates a specific identity, and those where an attacker impersonates an arbitrary identity in order to disguise its own. At a high level, impersonation encourages targets to answer attackers' calls and makes identifying attackers more difficult. This section shows how concrete attacks based on those different techniques might be launched.

3.1. Voicemail Hacking via Impersonation

A voicemail service may allow users calling from their phones access to their voicemail boxes on the basis of the calling party number. If an attacker wants to access the voicemail of a particular target, the attacker may try to impersonate the calling party number using one of the scenarios described in Section 4.

This attack is closely related to attacks on similar automated systems, potentially including banks, airlines, calling-card services, conferencing providers, ISPs, and other businesses that fully or partly grant access to resources on the basis of the calling party number alone (rather than any shared secret or further identity check). It is analogous to an attack in which a human is encouraged to answer a phone, or to divulge information once a call is in progress, by seeing a familiar calling party number.

The envisioned countermeasures for this attack involve the voicemail system treating calls that supply an authenticated calling number data differently from other calls. In the absence of that identity information, for example, a voicemail service might enforce some other caller authentication policy (perhaps requiring a PIN for caller authentication). Asserted caller identity alone provides an authenticated basis for granting access to a voice mailbox only when an identity is claimed legitimately; the absence of a verifiably legitimate calling identity here may not be evidence of malice, just of uncertainty or a limitation imposed by the set of intermediaries traversed for a specific call path.

If the voicemail service could learn ahead of time that it should expect authenticated calling number data from a particular number, that would enable the voicemail service to adopt stricter policies for handling a request without authentication data. Since users typically contact a voicemail service repeatedly, the service could for example remember which requests contain authenticated calling number data and require further authentication mechanisms when identity is absent. The deployment of such a feature would be facilitated in many environments by the fact that the voicemail service is often operated by an organization that would be in a position to enable or require authentication of calling party identity (for example, carriers or enterprises). Even if the voicemail service is decoupled from the number assignee, issuers of credentials or other authorities could provide a service that informs verifiers that they should expect identity in calls from particular numbers.

3.2. Unsolicited Commercial Calling from Impersonated Numbers

The unsolicited commercial calling, or for short robocalling, attack is similar to the voicemail attack, except that the robocaller does not need to impersonate the particular number controlled by the target, merely some "plausible" number. A robocaller may impersonate a number that is not an assignable number (for example, in the United States, a number beginning with 0), or an unassigned number. This behavior is seen in the wild today. A robocaller may change numbers every time a new call is placed, e.g., selecting numbers randomly.

A closely related attack is sending unsolicited bulk commercial messages via text messaging services. These messages usually originate on the Internet, though they may ultimately reach endpoints over traditional telephone network protocols or the Internet. While most text messaging endpoints are mobile phones, increasingly, broadband residential services support text messaging as well. The originators of these messages typically impersonate a calling party

number, in some cases a "short code" specific to text messaging services.

The envisioned countermeasures to robocalling are similar to those in the voicemail example, but there are significant differences. One important potential countermeasure is simply to verify that the calling party number is in fact assignable and assigned. Unlike voicemail services, end users typically have never been contacted by the number used by a robocaller before. Thus they can't rely on past association to anticipate whether or not the calling party number should supply authenticated calling number data. If there were a service that could inform the terminating side that it should expect this data for calls or texts from that number, however, that would also help in the robocalling case.

When a human callee is to be alerted at call setup time, the time frame for executing any countermeasures is necessarily limited. Ideally, a user would not be alerted that a call has been received until any necessary identity checks have been performed. This could however result in inordinate post-dial delay from the perspective of legitimate callers. Cryptographic and network operations must be minimized for these countermeasures to be practical. For text messages, a delay for executing anti-impersonation countermeasures is much less likely to degrade perceptible service.

The eventual effect of these countermeasures would be to force robocallers to either block their caller identity, in which case end users could opt not to receive such calls or messages, or to force robocallers to use authenticated calling numbers traceable to them, which would then allow for other forms of redress.

3.3. Telephony Denial-of-Service Attacks

In the case of telephony denial-of-service (or TDoS) attacks, the attack relies on impersonation in order to obscure the origin of an attack that is intended to tie up telephone resources. By placing incessant telephone calls, an attacker renders a target number unreachable by legitimate callers. These attacks might target a business, an individual or a public resource like emergency responders; the attacker may intend to extort the target. Attack calls may be placed from a single endpoint, or from multiple endpoints under the control of the attacker, and the attacker may control endpoints in different administrative domains. Impersonation in this case allows the attack to evade policies that would block based on the originating number, and furthermore prevents the victim from learning the perpetrator of the attack, or even the originating service provider of the attacker.

As is the case with robocalling, the attacker typically does not have to impersonate a specific number in order to launch a denial-of-service attack. The number simply has to vary enough to prevent simple policies from blocking the attack calls. An attacker may however have a further intention to create the appearance that a particular party is to blame for an attack, and in that case, the attacker might want to impersonate a secondary target in the attack.

The envisioned countermeasures are twofold. First, as with robocalling, ensuring that calling party numbers are assignable or assigned will help mitigate unsophisticated attacks. Second, if authenticated calling number data is supplied for legitimate calls, then Internet endpoints or intermediaries can make effective policy decisions in the middle of an attack by deprioritizing unsigned calls when congestion conditions exist; signed calls, if accepted, have the necessary accountability should it turn out they are malicious. This could extend to include, for example, an originating network observing a congestion condition for a destination number and perhaps dropping unsigned calls that are clearly part of a TDoS attack. As with robocalling, all of these countermeasures must execute in a timely manner to be effective.

There are certain flavors of TDoS attacks, including those against emergency responders, against which authenticated calling number data is unlikely to be a successful countermeasure. These entities are effectively obligated to attempt to respond to every call they receive, and the absence of authenticated calling number data in many cases will not remove that obligation.

4. Attack Scenarios

The examples that follow rely on Internet protocols including SIP [RFC3261] and WebRTC [I-D.ietf-rtcweb-overview].

Impersonation, IP-IP

An attacker with an IP phone sends a SIP request to an IP-enabled voicemail service. The attacker puts a chosen calling party number into the From header field value of the INVITE. When the INVITE reaches the endpoint terminal, the terminal renders the attacker's chosen calling party number as the calling identity.

Impersonation, PSTN-PSTN

An attacker with a traditional PBX (connected to the PSTN through ISDN) sends a Q.931 SETUP request with a chosen calling party number which a service provider inserts into the corresponding SS7 [refs.Q764] calling party number (CgPN) field of a call setup message

(IAM). When the call setup message reaches the endpoint switch, the terminal renders the attacker's chosen calling party number as the calling identity.

Impersonation, IP-PSTN

An attacker on the Internet uses a commercial WebRTC service to send a call to the PSTN with a chosen calling party number. The service contacts an Internet-to-PSTN gateway, which inserts the attacker's chosen calling party number into the SS7 [refs.Q764] call setup message (the CgPN field of an IAM). When the call setup message reaches the terminating telephone switch, the terminal renders the attacker's chosen calling party number as the calling identity.

Impersonation, IP-PSTN-IP

An attacker with an IP phone sends a SIP request to the telephone number of a voicemail service, perhaps without even knowing that the voicemail service is IP-based. The attacker puts a chosen calling party number into the From header field value of the INVITE. The attacker's INVITE reaches an Internet-to-PSTN gateway, which inserts the attacker's chosen calling party number into the CgPN of an IAM. That IAM then traverses the PSTN until (perhaps after a call forwarding) it reaches another gateway, this time back to the IP realm, to an H.323 network. The PSTN-IP gateway takes the calling party number in the IAM CgPN field and puts it into the SETUP request. When the SETUP reaches the endpoint terminal, the terminal renders the attacker's chosen calling party number as the calling identity.

4.1. Solution-Specific Attacks

Solution-specific attacks are outside the scope of this document, though two sorts of solutions are anticipated by the STIR problem statement: in-band and out-of-band solutions (see [I-D.ietf-stir-problem-statement]). There are a few points which future work on solution-specific threats must acknowledge. The design of the credential system envisioned as a solution to this threats must for example limit the scope of the credentials issued to carriers or national authorities to those numbers that fall under their purview. This will impose limits on what (verifiable) assertions can be made by intermediaries.

Some of the attacks that should be considered in the future include the following:

Attacks Against In-band Solutions

Replaying parts of messages used by the solution

Using a SIP REFER request to induce a party with access to credentials to place a call to a chosen number

Removing parts of messages used by the solution

Attacks Against Out-of-Band Solutions

Provisioning false or malformed data reflecting a placed call into any datastores that are part of the out-of-band mechanism

Mining any datastores that are part of the out-of-band mechanism

Attacks Against Either Approach

Attack on any directories/services that report whether you should expect authenticated calling number data or not

Canonicalization attacks

5. Acknowledgments

Sanjay Mishra, David Frankel, Penn Pfautz, Stephen Kent, Brian Rosen, Alex Bobotek, Henning Schulzrinne, Hannes Tschofenig, Cullen Jennings and Eric Rescorla provided key input to the discussions leading to this document.

6. IANA Considerations

This memo includes no request to IANA.

7. Security Considerations

This document provides a threat model and is thus entirely about security.

8. Informative References

[I-D.ietf-rtcweb-overview]

Alvestrand, H., "Overview: Real Time Protocols for Browser-based Applications", draft-ietf-rtcweb-overview-10 (work in progress), June 2014.

- [I-D.ietf-stir-problem-statement]
Peterson, J., Schulzrinne, H., and H. Tschofenig, "Secure Telephone Identity Problem Statement and Requirements", draft-ietf-stir-problem-statement-05 (work in progress), May 2014.
- [I-D.peterson-sipping-retarget]
Peterson, J., "Retargeting and Security in SIP: A Framework and Requirements", draft-peterson-sipping-retarget-00 (work in progress), February 2005.
- [RFC3261] 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.
- [refs.OMTP-VV]
OMTP, , "Visual Voice Mail Interface Specification", URL: http://www.gsma.com/newsroom/wp-content/uploads/2012/07/OMTP_VVM_Specification_1_3.pdf, May 1998.
- [refs.Q764]
ITU-T, , "Signaling System No. 7; ISDN User Part Signaling procedure", ITU-T URL: http://www.itu.int/rec/T-REC-Q.764/_page.print, September 1997.
- [refs.Q931]
ITU-T, , "ISDN user-network interface layer 3 specification for basic call control", ITU-T URL: <http://www.itu.int/rec/T-REC-Q.931-199805-I/en>, May 1998.

Author's Address

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

Email: jon.peterson@neustar.biz

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 24, 2014

J. Peterson
NeuStar
C. Jennings
Cisco
E. Rescorla
RTFM, Inc.
October 21, 2013

Authenticated Identity Management in the Session Initiation Protocol
(SIP)
draft-jennings-stir-rfc4474bis-00

Abstract

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

Status of This Memo

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

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

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

This Internet-Draft will expire on April 24, 2014.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents

(<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

Table of Contents

1. Introduction	3
2. Terminology	4
3. Background	4
3.1. Intermediary Authentication Services	6
4. Overview of Operations	6
5. Signature Generation and Validation	7
5.1. Authentication Service Behavior	7
5.1.1. Identity within a Dialog and Retargeting	10
5.2. Verifier Behavior	11
6. Credentials	13
6.1. Credential Use by the Authentication Service	13
6.2. Credential Use by the Verification Service	14
6.3. Handling Identity-Info URIs	14
7. Identity and Telephone Numbers	16
8. Considerations for User Agents	17
9. Considerations for Proxy Servers	18
10. Header Syntax	18
11. Compliance Tests and Examples	22
11.1. Identity-Info with a Singlepart MIME body	22
11.2. Identity for a Request with No MIME Body or Contact	25
12. Privacy Considerations	28
13. Security Considerations	28
13.1. Handling of digest-string Elements	29
13.2. Display-Names and Identity	31
13.3. Securing the Connection to the Authentication Service	32
13.4. Domain Names, Certificates and Subordination	33

13.5.	Authorization and Transitional Strategies	35
14.	IANA Considerations	36
14.1.	Header Field Names	36
14.2.	428 'Use Identity Header' Response Code	36
14.3.	436 'Bad Identity-Info' Response Code	37
14.4.	437 'Unsupported Certificate' Response Code	37
14.5.	438 'Invalid Identity Header' Response Code	37
14.6.	Identity-Info Parameters	37
14.7.	Identity-Info Algorithm Parameter Values	38
15.	Acknowledgements	38
16.	Original RFC 4474 Requirements	38
17.	Changes from RFC4474	39
17.1.	Motivation for Changes	39
17.2.	Changes to the Identity-Info Header	41
17.3.	Changes to the Identity Header	42
18.	References	43
18.1.	Normative References	43
18.2.	Informative References	43
	Authors' Addresses	45

1. Introduction

This document provides enhancements to the existing mechanisms for authenticated identity management in the Session Initiation Protocol (SIP, RFC 3261 [RFC3261]). An identity, for the purposes of this document, is defined as either a SIP URI, commonly a canonical address-of-record (AoR) employed to reach a user (such as 'sip:alice@atlanta.example.com'), or a telephone number, which can be represented as either a TEL URI or as the user portion of a SIP URI.

RFC 3261 [RFC3261] stipulates several places within a SIP request where 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 appropriately, in the absence of some sort of cryptographic authentication mechanism.

RFC 3261 [RFC3261] specifies a number of security mechanisms that can be employed by SIP user agents (UAs), including Digest, Transport Layer Security (TLS), and S/MIME (implementations may support other security schemes as well). However, few SIP user agents today support the end-user certificates necessary to authenticate themselves (via S/MIME, for example), and furthermore Digest authentication is limited by the fact that the originator and destination must share a prearranged secret. It is desirable for SIP user agents to be able to send requests to destinations with which 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 the person's displayed calling party number (and/or Caller-ID) is accurate. A cryptographic approach, like the one described in this document, can provide a much stronger and less spoofable assurance of identity than the telephone network provides today.

2. Terminology

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

3. Background

The usage of many SIP applications and services is governed by authorization policies. These policies may be automated, or they may be applied manually by humans. An example of the latter would be an Internet telephone application that displays the calling party number (and/or Caller-ID) of a caller, which a human may review (making a policy decision) before answering a call. An example of the former would be a voicemail service that compares the identity of the caller to a whitelist before determining whether it should allow the caller access to recorded messages. In both of these cases, attackers might attempt to circumvent these authorization policies through impersonation. Since the primary identifier of the sender of a SIP request, the From header field, can be populated arbitrarily by the controller of a user agent, impersonation is very simple today. The mechanism described in this document provides a strong identity system for SIP requests in which authorization policies cannot be circumvented by impersonation.

This document proposes an authentication architecture for SIP in which requests are processed by a logical authentication service that may be implemented as part of a user agent or as a proxy server. Once a message has been authenticated, the service then adds new cryptographic information to requests to communicate to other SIP entities that the sending user has been authenticated and its use of the From header field has been authorized.

But authorized by whom? Identities are issued to users by authorities. When a new user becomes associated with example.com, the administrator of the SIP service for that domain will issue them an identity in that namespace, such as alice@example.com. Alice may then send REGISTER requests to example.com that make her user agents eligible to receive requests for sip:alice@example.com. In some cases, Alice may be the owner of the domain herself, and may issue herself identities as she chooses. But ultimately, it is the

controller of the SIP service at example.com that must be responsible authorizing the use of names in the example.com domain. Therefore, the credentials needed to prove this authorization must ultimately derive from the domain owner: either a user agent gives requests to the domain name owner in order for them to be signed by the domain owner's credentials, or the user agent must possess credentials that prove in some fashion that the domain owner has given the user agent the right to a name.

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

This document specifies a means of sharing a cryptographic assurance of end-user SIP identity in an interdomain or intradomain context that is based on the authentication service adding a SIP header, the Identity header. In order to assist in the validation of this assurance, this specification also describes an Identity-Info header that can be used by the recipient of a request to recover the credentials of the signer. Note that the scope of this document is limited to providing this identity assurance for SIP requests; solving this problem for SIP responses is outside the scope of this work.

This specification allows either a user agent or a proxy server to provide identity services and to verify identities. To maximize end-to-end security, it is obviously preferable for end-users to acquire their own credentials; if they do, they can act as an authentication service. However, end-user credentials may be neither practical nor affordable, given the potentially large number of SIP user agents (phones, PCs, laptops, PDAs, gaming devices) that may be employed by a single user. In such environments, synchronizing keying material across multiple devices may be very complex and requires quite a good deal of additional endpoint behavior. Managing several credentials for the various devices could also be burdensome. This trade-off needs to be understood by implementers of this specification.

3.1. Intermediary Authentication Services

In cases where a user agent does not possess its own credentials to sign an Identity header, the user agent must send its request through an intermediary that will provide a signed Identity header based on the contents of the request. This requires, among other things, that intermediaries have some means of authenticating the user agents sending requests.

All RFC 3261 [RFC3261] compliant user agents support Digest authentication, which utilizes a shared secret, as a means for authenticating themselves to a SIP registrar. Registration allows a user agent to express that it is an appropriate entity to which requests should be sent for a particular SIP AoR URI (e.g., 'sip:alice@atlanta.example.com'). For such SIP URIs, by the definition of identity used in this document, registration proves the identity of the user to a registrar. Similar checks might be performed for telephone numbers as identities. This is of course only one manner in which a domain might determine how a particular user is authorized to populate the From header field; as an aside, for other sorts of URIs in the From (like anonymous URIs), other authorization policies would apply.

RFC 3261 [RFC3261] already describes an intermediary architecture very similar to the one proposed in this document 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.

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 [12]. 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.

4. 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 assert the identity that is populated in the From header field. This value may be Alice's AoR, or it may be some other value that the proxy server has authority over, such as a telephone number. It then computes a hash over some particular headers, including the From header field (and, optionally the body) in the message. This hash is signed with the appropriate credential (example.com, in the sip:alice@example.com case) and inserted in a new header field in the SIP message, the 'Identity' header.

The proxy, as the holder of the private key for 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) that appears in the From header field. The proxy also inserts a companion header field, Identity-Info, that tells Bob how to acquire keying material necessary to validate its credentials, 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 validate that the authority over the identity in the From header field authenticated the user, and permitted the user to assert that From header field value. This same validation operation may be performed by Bob's user agent server (UAS).

5. Signature Generation and Validation

5.1. Authentication Service Behavior

This document defines a role for SIP entities called an authentication service. The authentication service role can be instantiated by a proxy server or a user agent. Any entity that instantiates the authentication service role MUST possess the private key of one or more credentials that can be used to sign for a domain or a telephone number (see Section 6.1). Intermediaries that instantiate this role MUST be capable of authenticating one or more

SIP users who can register for that identity. Commonly, this role will be instantiated by a proxy server, since these entities are more likely to have a static hostname, hold corresponding credentials, and have access to SIP registrar capabilities that allow them to authenticate users. It is also possible that the authentication service role might be instantiated by an entity that acts as a redirect server, but that is left as a topic for future work.

SIP entities that act as an authentication service MUST add a Date header field to SIP requests if one is not already present (see Section 10 for information on how the Date header field assists verifiers). Similarly, authentication services MUST add a Content-Length header field to SIP requests if one is not already present; this can help verifiers to double-check that they are hashing exactly as many bytes of message-body as the authentication service when they verify the message.

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

Step 1:

The 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 SIP Secure (SIPS) URI, and the user portion is not a telephone number, the authentication service MUST extract the hostname portion of the identity field and compare it to the domain(s) for which it is responsible (following the procedures in RFC 3261 [RFC3261], Section 16.4), used by a proxy server to determine the domain(s) for which it is responsible). If the identity field uses the TEL URI scheme, or the identity field is a SIP or SIPS URI with a telephone number in the user portion, the authentication service determines whether or not it is responsible for this telephone number; see Section 7 for more information. If the authentication service is not authoritative for the identity in question, it SHOULD process and forward 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.

Step 2:

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

If the authentication service is instantiated by a SIP intermediary (proxy 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 [RFC3261], Section 22.3). Note that if that proxy server is maintaining a TLS connection with the client over which the client had previously authenticated itself using Digest authentication, the identity value obtained from that previous authentication step can be reused without an additional Digest challenge.

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 credential, or preferably by providing a password that unlocks said private key.

Authorization of the use of a particular username or telephone number in the user part of the From header field is a matter of local policy for the authentication service, see Section 6.1 for more information.

Note that this check is performed 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'). Authentication services MAY check and validate 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 response code. The reason phrase should indicate the nature of the problem; for example, "Inappropriate Display Name". However, the display-name is not always present, and in many environments the requisite operational procedures for display-name validation may not exist. For more information, see Section 13.2.

Step 3:

The authentication service SHOULD ensure that any preexisting Date header in the request is accurate. Local policy can dictate precisely how accurate the Date must be; a RECOMMENDED maximum discrepancy of ten minutes will ensure that the request is unlikely

to upset any verifiers. If the Date header contains a time different by more than ten minutes from the current time noted by the authentication service, the authentication service SHOULD reject the request. This behavior is not mandatory because a user agent client (UAC) could only exploit the Date header in order to cause a request to fail verification; the Identity header is not intended to provide a source of non-repudiation or a perfect record of when messages are processed. Finally, the authentication service MUST verify that the Date header falls within the validity period of its credential. For more information on the security properties associated with the Date header field value, see Section 10.

[TBD: Should consider a lower threshold than ten minutes? With the removal of other elements from the sig, that's a lot of leeway.]

Step 4:

The authentication service MAY form an identity-reliance signature and add an Identity-Reliance header to the request containing this signature. The Identity-Reliance header provides body security properties that are useful for non-INVITE transactions, and in environments where body security of INVITE transactions is necessary. Details on the generation of this header is provided in Section 10.

Step 5:

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 credential can be acquired; see Section 6.3 for more on credential acquisition. Details on the syntax of both of these headers are provided in Section 10.

Finally, the authentication service MUST forward the message normally.

5.1.1. Identity within a Dialog and Retargeting

Retargeting is broadly defined as the alteration of the Request-URI by intermediaries. More specifically, retargeting supplants the original target URI with one that corresponds to a different user, a user that is not authorized to register under the original target URI. By this definition, retargeting does not include translation of the Request-URI to a contact address of an endpoint that has registered under the original target URI, for example.

When a dialog-forming request is retargeted, this can cause a few wrinkles for the Identity mechanism when it is applied to requests sent in the backwards direction within a dialog. This section provides some non-normative considerations related to this case.

When a request is retargeted, it may reach a SIP endpoint whose user is not identified by the URI designated in the To header field value. The value in the To header field of a dialog-forming request is used as the From header field of requests sent in the backwards direction during the dialog, and is accordingly the header that would be signed by an authentication service for requests sent in the backwards direction. In retargeting cases, if the URI in the From header does not identify the sender of the request in the backwards direction, then clearly it would be inappropriate to provide an Identity signature over that From header. As specified above, if the authentication service is not responsible for the domain in the From header field of the request, it **MUST NOT** add an Identity header to the request, and it should process/forward the request normally.

Any means of anticipating retargeting, and so on, is outside the scope of this document, and likely to have equal applicability to response identity as it does to requests in the backwards direction within a dialog. Consequently, no special guidance is given for implementers here regarding the 'connected party' problem; authentication service behavior is unchanged if retargeting has occurred for a dialog-forming request. Ultimately, the authentication service provides an Identity header for requests in the backwards dialog when the user is authorized to assert the identity given in the From header field, and if they are not, an Identity header is not provided.

For further information on the problems of response identity and the potential solution spaces, see [15].

5.2. Verifier Behavior

This document introduces a new logical role for SIP entities called a verification service or verifier. When a verifier receives a SIP message containing an Identity header, it may inspect the signature to verify the identity of the sender of the message. Typically, the results of a verification are provided as input to an authorization process that is outside the scope of this document. If an Identity header is not present in a request, and one is required by local policy (for example, based on a per-sending-domain policy, or a per-sending-user policy), then a 428 'Use Identity Header' response **MUST** be sent.

In order to verify the identity of the sender of a message, an entity acting as a verifier MUST perform the following steps, in the order here specified.

Step 1:

In order to determine whether the signature for the URI in the From header field value should be over the entire URI or just a canonicalized telephone number, the verification service must follow the process described in Section 7. That section also describes the procedures the verification service must follow to determine if the signer is authoritative for a telephone number. For domains, the verifier MUST follow the process described in Section 13.4 to determine if the signer is authoritative for the URI in the From header field.

Step 2:

The verifier must first ensure that it possesses the proper keying material to validate the signature in the Identity header field. See Section 6.2 for more information on these procedures.

Step 3:

The verifier MUST verify the signature in the Identity header field, following the procedures for generating the hashed digest-string described in Section 10. If a verifier determines that the signature on the message does not correspond to the reconstructed digest-string, then a 438 'Invalid Identity Header' response MUST be returned.

Step 4:

If the request contains an Identity-Reliance header, the verifier SHOULD verify the signature in the Identity-Reliance header field, following the procedures for generating the hashed reliance-digest-string described in Section 10. If a verifier determines that the signature on the message does not correspond to the reconstructed digest-string, then a 438 'Invalid Identity Header' response SHOULD be returned.

Step 5:

The verifier MUST validate the Date header in the manner described in Section 13.1; recipients that wish to verify Identity signatures MUST support all of the operations described there. It must furthermore ensure that the value of the Date header falls within the validity period of the certificate whose corresponding private key was used to sign the Identity header.

6. Credentials

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

For more on the use of certificates for domain names as a credential system, see Section 13.4.

6.1. Credential Use by the Authentication Service

In order to act as an authentication service, a SIP entity must have access to the private keying material of one or more credentials that cover URIs, domain names or telephone numbers. These credentials may represent authority over only a single name (such as `alice@example.com`), an entire domain (such as `example.com`), or potentially a set of domains. Similarly, a credential may represent authority over a single telephone number or a range of telephone numbers. The way that the scope of a credential is expressed is specific to the credential mechanism.

Authorization of the use of a particular username or telephone number in the user part of the From header field is a matter of local policy for the authentication service, one that depends greatly on the manner in which authentication is performed. For non-telephone number user parts, one policy might be as follows: the username given in the 'username' parameter of the Proxy-Authorization header MUST correspond exactly to the username in the From header field of the SIP message. However, there are many cases in which this is too limiting or inappropriate; a realm might use 'username' parameters in Proxy-Authorization that do not correspond to the user-portion of SIP From headers, or a user might manage multiple accounts in the same administrative domain. In this latter case, a domain might maintain a mapping between the values in the 'username' parameter of Proxy-Authorization and a set of one or more SIP URIs that might legitimately be asserted for that 'username'. For example, the username can correspond to the 'private identity' as defined in Third

Generation Partnership Project (3GPP), in which case the From header field can contain any one of the public identities associated with this private identity. In this instance, another policy might be as follows: the URI in the From header field MUST correspond exactly to one of the mapped URIs associated with the 'username' given in the Proxy-Authorization header. This is a suitable approach for telephone numbers in particular. Various exceptions to such policies might arise for cases like anonymity; if the AoR asserted in the From header field uses a form like 'sip:anonymous@example.com', then the 'example.com' proxy should authenticate that the user is a valid user in the domain and insert the signature over the From header field as usual.

6.2. Credential Use by the Verification Service

In order to act as a verification service, a SIP entity must have a way to acquire and retain credentials for authorities over particular URIs, domain names and/or telephone numbers. The Identity-Info header (as described in the next section) is supported by all verification service implementations to create a baseline means of credential acquisition. Provided that the credential used to sign a message is not previously known to the verifier, SIP entities SHOULD discover this credential by dereferencing the Identity-Info header, unless they have some more efficient implementation-specific way of acquiring certificates. If the URI scheme in the Identity-Info header cannot be dereferenced, then a 436 'Bad Identity-Info' response MUST be returned.

In the case the credential is a certificate, the verifier processes this certificate in the usual ways, including checking that it has not expired, that the chain is valid back to a trusted certificate authority (CA), and that it does not appear on revocation lists. Once the certificate is acquired, it MUST be validated following the procedures in RFC 3280 [RFC3280]. If the certificate cannot be validated (it is self-signed and untrusted, or signed by an untrusted or unknown certificate authority, expired, or revoked), the verifier MUST send a 437 'Unsupported Certificate' response.

Verification service implementations supporting this specification SHOULD have some means of retaining credentials (in accordance with normal practices for credential lifetimes and revocation) in order to prevent themselves from needlessly downloading the same credential every time a request from the same identity is received. Credentials cached in this manner SHOULD be indexed by their scope, or the URI given in the Identity-Info header field value.

6.3. Handling Identity-Info URIs

A URI in an Identity-Info header MUST contain a URI which dereferences to a resource containing the credential used by the authentication service to sign a request. Much as is the case with the trust anchor(s) required for deployments of this specification, it is essential that a URI in the Identity-Info header be dereferencable by any entity that can receive the request. For common cases, this means that the URI must be dereferencable by any entity on the public Internet. In constrained deployment environments, a service private to the environment might be used instead.

Beyond providing a means of accessing credentials for an identity, the Identity-Info header further services a means of differentiating which particular credential was used to sign a request, when there are potentially multiple authorities eligible to sign. For example, imagine a case where a domain implements the authentication service role for example.com, and a user agent belonging to Alice has acquired a credential for alice@example.com. Either would be eligible to sign a SIP request from alice@example.com. Verification services however need a means to differentiate which one performed the signature. The Identity-Info header performs that function.

All implementations of this specification MUST support the use of HTTP and HTTPS URIs in the Identity-Info header. Such HTTP and HTTPS URIs MUST follow the conventions of RFC 2585 [RFC2585], and for those URIs the indicated resource MUST be of the form 'application/pkix-cert' described in that specification. Note that this introduces key lifecycle management concerns; were a domain to change the key available at the Identity-Info URI before a verifier evaluates a request signed by an authentication service, this would cause obvious verifier failures. When a rollover occurs, authentication services SHOULD thus provide new Identity-Info URIs for each new certificate, and SHOULD continue to make older key acquisition URIs available for a duration longer than the plausible lifetime of a SIP message (an hour would most likely suffice).

Beyond HTTP, implementations may support any of several alternative mechanism for acquiring credentials. When implemented as part of a user agent, for example, an authentication service might include its credential as an additional MIME body in the SIP request, and refer to the certificate with a CID URI (per [RFC2392]). Uses of SIP outside of the request transaction may be suitable for transmitting certificates in some environments, such as through a SUBSCRIBE/NOTIFY exchange. As DANE deployment increases with the widespread adoption of DNSSEC, implementations may want to rely on keying material stored in the DNS. The Identity-Info headers may use the DNS URL scheme to indicate keys in the DNS.

[TBD: Should we add some kind of hash or similar indication to the Identity-Info header to make it easier for verifiers to ascertain that they already possess a credential without dereferencing the URI?]

7. Identity and Telephone Numbers

Since many SIP applications provide a Voice over IP (VoIP) service, telephone numbers are commonly used as identities in SIP deployments. In order for telephone numbers to be used with the mechanism described in this document, authentication services must enroll with an authority that issues credentials for telephone numbers or telephone number ranges, and verification services must trust the authority employed by the authentication service that signs a request.

Given the existence of such authorities, authentication and verification services must furthermore identify when a request should be signed by an authority for a telephone number, and when it should be signed by an authority for a domain. Telephone numbers most commonly appear in SIP requests in the username portion of a SIP URI (e.g., 'sip:+17005551008@chicago.example.com/user=phone'). The user part of that URI conforms to the syntax of the TEL URI scheme (RFC 3966 [RFC3966]). It is also possible for a TEL URI to appear in the SIP To or From header field outside the context of a SIP or SIPS URI (e.g., 'tel:+17005551008'). In both of these cases, it's clear that the signer must have authority over the telephone number, not the domain name of the SIP URI. It is also possible, however, for requests to contain a URI like 'sip:7005551000@chicago.example.com'. It may be non-trivial for a service to ascertain in this case whether the URI contains a telephone number or not.

To address this problem, the authentication service and verification service both must perform the following canonicalization procedure on any SIP URI they inspect which contains a wholly numeric user part. [TBD: the algorithm] If the result of this procedure forms a complete telephone number, that number is used for the purpose of creating and signing the digest-string at the authentication service and verification service. If the result does not form a complete telephone number, the authentication service and verification service should treat the entire URI as a SIP URI, and apply a domain signature per the procedures in Section 13.4.

This specification assumes that UACs will have an appropriate means to discover an authentication service that can sign with a telephone number certificate corresponding to the UAC's telephone number. Most likely, this information will simply be provisioned in UACs.

Certificates that prove authority over telephone numbers should contain the telephone number, or number range, in the [TBD] field of the certificate. Verification services must compare the canonicalized telephone number to the contents of the [TBD] field in order to establish that the proper authority has signed the request. [TBD: This would refer to an external specification, most likely]

In the longer term, it is possible that some directory or other discovery mechanism 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.

8. Considerations for User Agents

This mechanism can be applied opportunistically to existing SIP deployments; accordingly, it requires no change to SIP user agent behavior in order for it to be effective. However, because this mechanism does not provide integrity protection between the UAC and the authentication service, a UAC SHOULD implement some means of providing this integrity. TLS would be one such mechanism, which is attractive because it MUST be supported by SIP proxy servers, but is potentially problematic because it is a hop-by-hop mechanism. See Section 13.3 for more information about securing the channel between the UAC and the authentication service.

When a UAC sends a request, it MUST accurately populate the From header field with a value corresponding to an identity that it believes it is authorized to claim. In a request, it MUST set the URI portion of its From header to match a SIP, SIPS, or TEL URI AoR that it is authorized to use in the domain (including anonymous URIs, as described in RFC 3323 [RFC3323]).

Note that this document defines a number of new 4xx response codes. If user agents support these response codes, they will be able to respond intelligently to Identity-based error conditions.

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. For a given domain, if an entity that can instantiate the authentication service role is not in the path of dialog-forming requests, identity for mid-dialog requests in the backwards direction cannot be provided.

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. Considerations for Proxy Servers

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 verification service, it MAY validate the Identity header before it makes a forwarding decision for a request. Compliant proxy servers MUST NOT remove or modify an existing Identity or Identity-Info header in a request.

10. Header Syntax

This document specifies three SIP headers: Identity, Identity-Reliance and Identity-Info. Each of these headers can appear only once in a SIP request; Identity-Reliance is OPTIONAL, while Identity and Identity-Info are REQUIRED for securing requests with this specification. The grammar for these three headers is (following the ABNF [6] in RFC 3261 [1]):

```
Identity = "Identity" HCOLON signed-identity-digest
signed-identity-digest = LDQUOTE 32LHEX RDQUOTE
```

```
Identity-Reliance = "Identity-Reliance" HCOLON signed-identity-reliance-dig
est signed-identity-reliance-digest = LDQUOTE 32LHEX RDQUOTE
```

```
Identity-Info = "Identity-Info" HCOLON ident-info
                *( SEMI ident-info-params )
ident-info = LAQUOTE absoluteURI RAQUOTE
ident-info-params = ident-info-alg / ident-info-extension
ident-info-alg = "alg" EQUAL token
ident-info-extension = generic-param
```

[TBD: The version has the Identity-Reliance header covered under the Identity signature. It is also possible to do this the other way around, where the base Identity signature is generated first, and Identity-Reliance would cover both the Identity header and the body. This is a trade-off of whether the authentication service should decide whether Identity-Reliance is needed or if the verification service should decide. These have different properties, and some investigation would be needed to decide between them.]

The signed-identity-reliance-digest is a signed hash of a canonical string generated from certain components of a SIP request. Creating this hash and the Identity-Reliance header field to contain it is OPTIONAL, and its usage is a matter of policy for authentication services. To create the contents of the signed-identity-digest, the following element of a SIP message MUST be placed in a bit-exact string:

The body content of the message with the bits exactly as they are in the message (in the ABNF for SIP, the message-body). This includes all components of multipart message bodies. 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.

[TBD: Explore alternatives to including the whole body for INVITE requests]

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 vertical line, "|" or %x7C, character:

First, the identity. If the user part of the AoR in the From header field of the request contains a telephone number, then the canonicalization of that number goes into the first slot (see Section 7). Otherwise, the first slot contains the AoR of the UA sending the message, or addr-spec of the From header field.

Second, the target. If the user part of the AoR in the To header field of the request contains a telephone number, then the canonicalization of that number goes into the second slot (see Section 7). Otherwise, the second slot contains the addr-spec component of the To header field, which is the AoR to which the request is being sent.

Third, the request method.

Fourth, 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 RFC 3261 [RFC3261]. RFC 3261 specifies that the BNF for weekday and month is a choice amongst a set of tokens. The RFC 2234 [RFC2234] rules for the BNF specify that tokens are case sensitive. However, when used to construct the canonical string defined here, the first letter of each week and month MUST be capitalized, and the remaining two letters must be lowercase. This matches the capitalization provided in the definition of each

token. All requests that use the Identity mechanism MUST contain a Date header.

Fifth, the Identity-Reliance header field value, if there is an Identity-Reliance field in the request. If the message has no body, or no Identity-Reliance header, then the fifth slot will be empty, and the final "|" will not be followed by any additional characters.

[TBD: Should there be a special case for security parameters that would appear in SDP?]

For more information on the security properties of these headers, and why their inclusion mitigates replay attacks, see Section 13 and [RFC3893]. The precise formulation of this digest-string is, therefore (following the ABNF[RFC4234] in RFC 3261 [RFC3261]):

```
digest-string = addr-spec / tn-spec "|" addr-spec / tn-spec "|"
                Method "|" SIP-date "|" [ signed-identity-reliance-digest ]
```

For the definition of 'tn-spec' see Section 7.

After the digest-string or reliance-digest-string is formed, each MUST be hashed and signed with the certificate of authority over the identity. The hashing and signing algorithm is specified by the 'alg' parameter of the Identity-Info header (see below for more information on Identity-Info header parameters). This document defines only one value for the 'alg' parameter: 'rsa-sha1'; further values MUST be defined in a Standards Track RFC, see Section 14.7 for more information. All implementations of this specification MUST support 'rsa-sha1'. When the 'rsa-sha1' algorithm is specified in the 'alg' parameter of Identity-Info, the hash and signature MUST be generated as follows: compute the results of signing this string with sha1WithRSAEncryption as described in RFC 3370 [RFC3370] and base64 encode the results as specified in RFC 3548 [RFC3548]. A 1024-bit or longer RSA key MUST be used. The result of the digest-string hash is placed in the Identity header field; the optional reliance-digest-string hash goes in the Identity-Reliance header. For detailed examples of the usage of this algorithm, see Section 11.

The 'absoluteURI' portion of the Identity-Info header MUST contain a URI; see Section 6.3 for more on choosing how to advertise credentials through Identity-Info.

The Identity-Info header field MUST contain an 'alg' parameter. No other parameters are defined for the Identity-Info header in this document. Future Standards Track RFCs may define additional Identity-Info header parameters.

This document adds the following entries to Table 2 of RFC 3261 [RFC3261] (this repeats the registrations of RFC4474):

Header field	where	proxy	ACK	BYE	CAN	INV	OPT	REG
-----	----	-----	---	---	---	---	---	---
Identity	R	a	o	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	o
			SUB	NOT	REF	INF	UPD	PRA
			---	---	---	---	---	---
			o	o	o	o	o	o
Header field	where	proxy	ACK	BYE	CAN	INV	OPT	REG
-----	----	-----	---	---	---	---	---	---
Identity-Reliance	R	a	o	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 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 Identity and Identity-Info header MUST NOT appear in CANCEL. Note as well that the use of Identity with REGISTER is consequently a subject for future study, although it is left as optional here for forward-compatibility reasons.

11. Compliance Tests and Examples

[TBD: Need to fix examples for RFC4474bis]

The examples in this section illustrate the use of the Identity header in the context of a SIP transaction. Implementers are advised to verify their compliance with the specification against the following criteria:

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.

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 it appears here only for illustrative purposes. Therefore, in compliance testing, implementations of verifiers SHOULD generate appropriate warnings about the use of self-signed certificates. Also, the example certificates in this section have placed their domain name subject in the subjectAltName field; in practice, certificate authorities may place domain names in other locations in the certificate (see Section 13.4 for more information).

Note that all examples in this section use the 'rsa-sha1' algorithm.

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' (rendered in OpenSSL format).

```
-----BEGIN RSA PRIVATE KEY-----
MIICXQIBAAKBgQDPPMBtHVOPkXV+Z6jq1LsgfTELVWpy2BVUffJMPH06LL0cJSQO
aIeVzIoJzWtpauB7Iy1ZK1Ajb5f429tRuoUiedCwMLKblWAqZt6eHWpCNZJ71ONc
IEwnmh2nAccKk83Lp/VH3tgAS/43DQoX2sndnYh+g8522Pzwg7EGWspzzwIDAQAB
AoGBAK0W3tnEFD7AjbVQAnJNXDtx59AalVu2JEXe6oi+OrkFysJjbZJwsLmKtrgtt
PXOU8t2mZpi0wK4hX4tZhntiwGKkUPC3h9Bjp+GerifP34lRMyMO+6fPgjqOzUDw
+rPjjMpwD7AkEcqDgbTrZnWv/QnCSaaF3xkUGfFkLx5OKcRAkEA7UxnsE8XaT30
```

```

tP/UUC5lgNk2KGKgxQQTHopBcew9yfeCRFhvdL7jpaGatei5iZwGGQQDVOVHUN1H
0YLpHQjRowJBAN+R2bvA/Nimq464ZgnelEDPqaEAZWad3kofhS9+vL7oqES+u5E0
J7kXb7ZkiSVUg9XU/8PxMKx/Daz0dUmOL+UCQH8C9ETUMI2uEbqHbBdVUGNk364C
DFcndSxVh+34KqJdjiYSx6VPPv26X9m7S0OydTkSgs3/4ooPxo8HaMqXm80CQB+r
xbB3UlpOohcBwFK9mTrlMB6Cs9ql66KgwnlL9ukEhHHYozGatdXeoBCyHUsogdSU
6/aSAFcVWEGtj7/vyJECQQCCS1lKgEXoNQPPqONalvYhyyMZRXFLdD4gbwRPK1uXK
Ypk3CkfFzOyfjeLcGPxXzq2qzuHzGTDdxZ9PAepwX4RSk
-----END RSA PRIVATE KEY-----

```

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

MIIC3TCCAkagAwIBAgIBADANBgkqhkiG9w0BAQUFADBZMQswCQYDVQQGEWJVUzELMAkGA1UECAwCR0ExEDAOBgNVBACMB0F0bGFudGExDTALBgNVBAoMBE1FVEYxHDAaBgNVBAMME2F0bGFudGEuZXhhbXBsZS5jb20wHhcNMMDUxMDI0MDYzNjA2WhcNMMDYxMDI0MDYzNjA2WjBZMQswCQYDVQQGEWJVUzELMAkGA1UECAwCR0ExEDAOBgNVBACMB0F0bGFudGExDTALBgNVBAoMBE1FVEYxHDAaBgNVBAMME2F0bGFudGEuZXhhbXBsZS5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAM88wG0dWg+RdX5nqUrUuyB9MQtVanLYFVR98kw8fTosvRwlJA5oh5XMiipNA2lq4HsjKVkqUCMHL1/jb2lG6hSJ50LAWspuVYCpm3p4dakI1knuU4lwgtCeaHacBxwqtZcun9Ufe2ABL/jcNChfayd2diH6Dznby/PCDsQZaynPpAgMBAAGjgbQwgbEwHQYDVR0OBBYEFNmU/MrbVYceKDr/20WISrGljlRNMIGBBgNVHSMEEjb4gBTZlPzK2lWHBCG6/9tFiEqxtY9azaFdPfsWTELMakGA1UEBhMCVVMxMzA2bG9uZGVzZS5jb20wHhcNMMDUxMDI0MDYzNjA2WhcNMMDYxMDI0MDYzNjA2WjBZMQswCQYDVQQGEWJVUzELMAkGA1UECAwCR0ExEDAOBgNVBACMB0F0bGFudGExDTALBgNVBAoMBE1FVEYxHDAaBgNVBAMME2F0bGFudGEuZXhhbXBsZS5jb20wgZ8wDQYJKoZIhvcNAQEFBQADgYEAddQYtswBDmTSTq0mt2l17alm/XGFrnb2zdbU0vorxRdOZ04qMyrIpXG1LEmnEOgcocyrXRBvq5p6WbZAcEQk0DsE3Ve0Nc8x9nmvjl7WGsMGFCnCu04ODTf/1lGdVr9DeCzcj10YUQ3MRemDMXhY2CDisLw17SX0ORcZailoU9W=

-----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:

```

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>
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|
INVITE|Thu, 21 Feb 2002 13:02:03 GMT|

```

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

```

ZYNBbHC00VMZr2kZt6VmCvPonWJMGvQTBdqghoWeLxJfzB2alpxAr3VgrB0SsSAA
ifsRdiOPoQZY0y2wrVghuhcsMbHWUSFXi6p6q5TOQXHMmz6uEo3svJsSH49thyGn
FVcnYaz++yRlBYyQTLqWzJ+KVhPKbfU/pryhVn9Yc6U=

```

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 the following:

```

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:
"ZYNBbHC00VMZr2kZt6VmCvPonWJMGvQTBdqghoWeLxJfzB2alpxAr3VgrB0SsSAA
ifsRdiOPoQZY0y2wrVghuhcsMbHWUSFXi6p6q5TOQXHMmz6uEo3svJsSH49thyGn
FVcnYaz++yRlBYyQTLqWzJ+KVhPKbfU/pryhVn9Yc6U="

```

```

Identity-Info: <https://atlanta.example.com/atlanta.cer>;alg=rsa-sha1
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 dereferences the URL in 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 dereferencing 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".

```

-----BEGIN RSA PRIVATE KEY-----
MIICXgIBAAKBgQC/obBYLRMPjskrAqWoiGPAUxI3/m2ti7ix4caqCTAuFX5cLegQ
7nmquLOHfIhxVIqT2f06UA010o2NVofK9G7MTkVbVNiyAlLYUDEj7XWLDICf3ZHL
6Fr/+CF7wrQ9r4kv7XiJKxodVCCd/DhCT9Gp+VDoe8HymqOW/KsneriyIwIDAQAB
AoGBAJ7fsFIKXKkjWgj8ksGothS3Sn19xPSCyEdBxfEm2Pj7/Nzzeli/PcOaic0k
JALBcnqN2fHEeIGK/9xUBxTufgQYVJqvyHERs6rXX/it4Ynm9t1905EiQ9ZpHsrI
/AMMUYAlQrGgAIHvZLVLzq+9KLDEZ+HQBucLJXF+6bl0Eb5BAKEA636oMANp0Qa3
mYWEQ2utmGsYxkXSfyBbl8TCOWCty0ndBR24zyOJF2NbZS98Lz+Ga25hfIGw/JHK
nD9bOE88UwJBANBRSpd4bmS+m48R/13tRESAtHgydNinX0kS/RhwHr7mkHTU3k/M
FxQtX34I3GKzaZxMn0A66KS9v/SHdnF+ePECQQCGe7QshyZ8uitLPtZDclCWHEKH
qAQHmUEZvUF2VHLrbukLLOgHUrHNa24cILv4d3yaCVUetymNcuyTwhKj24wFAkAO
z/jx1EplN3hwL+Nsl1ZoWI58uvu7/Aq2c3czqaVGBbb317sHCYgKk0bAG3kw03mi
93/LXWTlcdiYVpmBcHDBAKEAmpgkFj+XZu5gWASY5ujv+FCMP0WwaH5hTnXu+tKe
PJ3d2IJZKxGnl6itKRN7Gerh9PSK0kZSgFeVrvsJ4Nopg==
-----END RSA PRIVATE KEY-----

```

```

-----BEGIN CERTIFICATE-----
MIIC1jCCA+jgAwIBAgIBADANBgkqhkiG9w0BAQUFADBXMQswCQYDVQQGEwJVUzEL
MAKGA1UECAwCTVMxDzANBgNVBACMBKJpbG94aTENMASGA1UECgwESUVURjEjBmBkG

```



```

A1UEAwWSYmlsb3hpLmV4YWlwbGUuY29tMB4XDTA1MTAyNDA2NDAYNloXDTA2MTAy
NDA2NDAYNlowVzELMAkGA1UEBhMCVVMxCzAJBgNVBAGMAk1TMQ8wDQYDVQQHDAZC
aWxveGkxDTALBgNVBAoMBE1FVEYxGzAZBgNVBAMMEJpbG94aS5leGFtcGxlLmNv
bTCBnzANBjkqhkig9w0BAQEFAAOBjQAwGyKCGYEA6GwWC0TD47JKwK1johjwFMS
N/5trYu4seHGqgkwLhV+XC3oEO55qrizh3yIcVSKk9n901ANJTqNjVaHyvRuzE5F
WlTYsgJS2FAxI+1liwyAn92Ry+ha//ghe8K0Pa+JL+14iSsaHVQgnfw4Qk/RqflQ
6HvB8pqjlvyrJ3q4siMCAwEAAaOBsTCBrjAdBgNVHQ4EFgQU0Z+RL47W/APDtc5B
fSoQXuEFE/wwfwYDVR0jBHgwdoAU0Z+RL47W/APDtc5BfSoQXuEFE/yhW6RZMFcx
CzAJBgNVBAYTA1VTMQswCQYDVQQIDAJNUzEPMA0GA1UEBwwGQmlsb3hpMQ0wCwYD
VQQKDARJRVRGRMRswGQYDVQQDDBJiaWxveGkuZXhhbXBsZS5jb22CAQAwDAYDVR0T
BAUwAwEB/zANBjkqhkig9w0BAQUFAAOBgQBiyKHIt8TXfGNfnpJXi5jCizOxmY8Y
gln8tyPFaeyq95TGcvTCWzdoBLVpBD+fprWRX/II5sE6VHbbAPjjVmKbZwzQAtpp
P2Fauj28t94ZeDHN2vqzjfnHjCO24kG3Juf2T80ilp9YHcDwxjUFRt86UnlC+yid
yaTeusW5Gu7vlg==
-----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

[TBD: Fix example.] 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 vertical bars. 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 BYE|Thu, 21 Feb 2002 14:19:51 GMT||

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

sv5CTo05KqpSmtHt3dcEiO/1CWTSZtnG3iV+lnmurLXV/HmtynS7Ltrg9dlxkWzo
eU7d7OV8HweTTDobV3itTmgPwCFjaEmMyEI3d7SyN21yNDo2ER/Ovgtw0Lu5csIp
pPqOgluXndzHbG7mR6Rl9BnUhHufVRbp51Mn3w0gfUs=

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 the following:

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:
 "sv5CTo05KqpSmtHt3dcEiO/1CWTSZtnG3iV+lnmurLXV/HmtynS7Ltrg9dlxkWzo
 eU7d7OV8HweTTDobV3itTmgPwCFjaEmMyEI3d7SyN21yNDo2ER/Ovgtw0Lu5csIp
 pPqOgluXndzHbG7mR6Rl9BnUhHufVRbp51Mn3w0gfUs="

Identity-Info: <https://biloxi.example.org/biloxi.cer>;alg=rsa-sha1
Content-Length: 0

biloxi.example.org then forwards the request normally.

12. Privacy Considerations

The identity mechanism presented in this document is compatible with the standard SIP practices for privacy described in RFC 3323 [RFC3323]. A SIP proxy server can act both as a privacy service and as an authentication service. Since a user agent can provide any From header field value that the authentication service is willing to authorize, there is no reason why private SIP URIs that contain legitimate domains (e.g., sip:anonymous@example.com) cannot be signed by an authentication service. The construction of the Identity header is the same for private URIs as it is for any other sort of URIs.

Note, however, that for using anonymous SIP URIs, an authentication service must possess a certificate corresponding to the host portion of the addr-spec of the From header field of the request; accordingly, using domains like 'anonymous.invalid' will not be possible for privacy services that also act as authentication services. The assurance offered by the usage of anonymous URIs with a valid domain portion is "this is a known user in my domain that I have authenticated, but I am keeping its identity private". The use of the domain 'anonymous.invalid' entails that no corresponding authority for the domain can exist, and as a consequence, authentication service functions are meaningless.

RFC 3325 [RFC3325] defines the "id" priv-value token, which is specific to the P-Asserted-Identity header. The sort of assertion provided by the P-Asserted-Identity header is very different from the Identity header presented in this document. It contains additional information about the sender of a message that may go beyond what appears in the From header field; P-Asserted-Identity holds a definitive identity for the sender that is somehow known to a closed network of intermediaries 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.

Finally, note that unlike RFC 3325 [RFC3325], the mechanism described in this specification adds no information to SIP requests that has privacy implications.

13. Security Considerations

13.1. Handling of digest-string Elements

This document describes a mechanism that provides a signature over the Date header field, and either the whole or part of the To and From header fields of SIP requests, as well as optional protections for the message body. 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 RFC 3261 [RFC3261] for including headers in tunneled 'message/sip' MIME bodies (see Section 23 in particular). The following section details the individual security properties obtained by including each of these header fields within the signature; collectively, this set of header fields provides the necessary properties to prevent impersonation.

The From header field indicates the identity of the sender of the message, and the SIP address-of-record URI, or an embedded telephone number, in the From header field is the identity of a SIP user, for the purposes of this document. 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 cut-and-paste attacks in which an Identity header from legitimate request for one user is cut-and-pasted into a request for a different user; second, it preserves the starting URI scheme of the request, which helps prevent downgrade attacks against the use of SIPS.

The Date header field provides replay protection, as described in RFC 3261 [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). The result of this is that if an Identity header is replayed within the Date interval, verifiers will recognize that it is invalid; if an Identity header is replayed after the Date interval, verifiers will recognize that it is invalid because the Date is stale.

Without the method an INVITE request could be cut- and-pasted by an attacker and transformed into a MESSAGE 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.

RFC4474 had protections for the Contact, Call-ID and CSeq. These are removed from RFC4474bis. The absence of these header values creates some opportunities for determined attackers to impersonate based on

cut-and-paste attacks; however, the absence of these headers does not seem impactful to preventing against the simple unauthorized claiming of a From header field value.

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

This mechanism also provides an optional signature over the bodies of SIP requests. This can help to protect non-INVITE transactions such as MESSAGE or NOTIFY, as well as INVITES in those environments where intermediaries do not change SDP. While this is not strictly necessary to prevent the impersonation attacks, there is little purpose in establishing the identity of the user that originated a SIP request if this assurance is not coupled with a comparable assurance over the contents of the message. There are furthermore some baiting attacks (where the attacker receives a request from the target and reoriginates it to a third party) that might not be prevented by only a signature over the From, To and Date, but could be prevented by securing SDP. Note, however, that this is not perfect end-to-end security. The authentication service itself, when instantiated at an intermediary, could conceivably change the body (and SIP headers, for that matter) before providing a signature. Thus, while this mechanism reduces the chance that a replayer or man-in-the-middle will modify bodies, it does not eliminate it entirely. Since it is a foundational assumption of this mechanism that the users trust 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.

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

One additional respect in which the Identity-Info header cannot protect itself is the 'alg' parameter. The 'alg' parameter is not included in the digest-string, and accordingly, a man-in-the-middle might attempt to modify the 'alg' parameter. However, it is important to note that preventing men-in-the-middle is not the primary impetus for this mechanism. Moreover, changing the 'alg'

would at worst result in some sort of bid-down attack, and at best cause a failure in the verifier. Note that only one valid 'alg' parameter is defined in this document and that thus there is currently no weaker algorithm to which the mechanism can be bid down. 'alg' has been incorporated into this mechanism for forward-compatibility reasons in case the current algorithm exhibits weaknesses, and requires swift replacement, in the future.

13.2. Display-Names and Identity

As a matter of interface design, SIP user agents might render the display-name portion of the From header field of a caller as the identity of the caller; there is a significant precedent in email user interfaces for this practice. As such, it might seem that the lack of a signature over the display-name is a significant omission.

However, there are several important senses in which a signature over the display-name does not prevent impersonation. In the first place, a particular display-name, like "Jon Peterson", is not unique in the world; many users in different administrative domains might legitimately claim that name. Furthermore, enrollment practices for SIP-based services might have a difficult time discerning the legitimate display-name for a user; it is safe to assume that impersonators will be capable of creating SIP accounts with arbitrary display-names. The same situation prevails in email today. Note

that an impersonator who attempted to replay a message with an Identity header, changing only the display-name in the From header field, would be detected by the other replay protection mechanisms described in Section 13.1.

Of course, an authentication service can enforce policies about the display-name even if the display-name is not signed. The exact mechanics for creating and operationalizing such policies is outside the scope of this document. The effect of this policy would not be to prevent impersonation of a particular unique identifier like a SIP URI (since display-names are not unique identifiers), but to allow a domain to manage the claims made by its users. If such policies are enforced, users would not be free to claim any display-name of their choosing. In the absence of a signature, man-in-the-middle attackers could conceivably alter the display-names in a request with impunity. Note that the scope of this specification is impersonation attacks, however, and that a man-in-the-middle might also strip the Identity and Identity-Info headers from a message.

There are many environments in which policies regarding the display-name aren't feasible. Distributing bit-exact and internationalizable display-names to end-users as part of the enrollment or registration process would require mechanisms that are not explored in this document. In the absence of policy enforcement regarding domain names, there are conceivably attacks that an adversary could mount against SIP systems that rely too heavily on the display-name in their user interface, but this argues for intelligent interface design, not changes to the mechanisms. Relying on a non-unique identifier for identity would ultimately result in a weak mechanism.

13.3. Securing the Connection to the Authentication Service

The assurance provided by this mechanism is strongest when a user agent forms a direct connection, preferably one secured by TLS, to an intermediary-based authentication service. The reasons for this are twofold:

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

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

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

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

13.4. Domain Names, Certificates and Subordination

When a verifier processes a request containing an Identity-Info header with a domain signature, it must compare the domain portion of the URI in the From header field of the request with the domain name that is the subject of the certificate acquired from the Identity-Info header. While it might seem that this should be a straightforward process, it is complicated by two deployment realities. In the first place, certificates have varying ways of describing their subjects, and may indeed have multiple subjects, especially in 'virtual hosting' cases where multiple domains are managed by a single application. Secondly, some SIP services may delegate SIP functions to a subordinate domain and utilize the procedures in RFC 3263 [RFC3263] that allow requests for, say, 'example.com' to be routed to 'sip.example.com'. As a result, a user with the AoR 'sip:jon@example.com' may process requests through a host like 'sip.example.com', and it may be that latter host that acts as an authentication service.

To meet the second of these problems, a domain that deploys an authentication service on a subordinate host MUST be willing to supply that host with the private keying material associated with a certificate whose subject is a domain name that corresponds to the domain portion of the AoRs that the domain distributes to users. Note that this corresponds to the comparable case of routing inbound SIP requests to a domain. When the NAPTR and SRV procedures of RFC 3263 are used to direct requests to a domain name other than the domain in the original Request-URI (e.g., for 'sip:jon@example.com', the corresponding SRV records point to the service 'sip1.example.org'), the client expects that the certificate passed back in any TLS exchange with that host will correspond exactly with the domain of the original Request-URI, not the domain name of the host. Consequently, in order to make inbound routing to such SIP services work, a domain administrator must similarly be willing to share the domain's private key with the service. This design decision was made to compensate for the insecurity of the DNS, and it makes certain potential approaches to DNS-based 'virtual hosting' unsecurable for SIP in environments where domain administrators are unwilling to share keys with hosting services.

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

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 to 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. It is strongly RECOMMENDED that self-signed domain certificates should not be trusted by verifiers, unless some previous key exchange has justified such trust.

[TBD: DANE?]

For further information on certificate security and practices, see RFC 3280 [RFC3280]. The Security Considerations of RFC 3280 [RFC3280] are applicable to this document.

13.5. Authorization and Transitional Strategies

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 used its administrative practices to authenticate its users. However, it is possible that some domains will implement policies that effectively make users unaccountable (e.g., ones that accept unauthenticated registrations from arbitrary users). The value of an Identity header from such 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.

One cannot expect the Identity and Identity-Info headers to be supported by every SIP entity overnight. This leaves the verifier in a compromising position; when it receives a request from a given SIP user, how can it know whether or not the sender's domain supports Identity? In the absence of ubiquitous support for identity, some transitional strategies are necessary.

A verifier could remember when it receives a request from a domain that uses Identity, and in the future, view messages received from that domain without Identity headers with skepticism.

A verifier could query the domain through some sort of callback system to determine whether or not it is running an authentication service. There are a number of potential ways in which this could be implemented; use of the SIP OPTIONS method is one possibility. This is left as a subject for future work.

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

Finally, it is worth noting that the presence or absence of the Identity headers cannot be the sole factor in making an authorization decision. Permissions might be granted to a message on the basis of

the specific verified Identity or really on any other aspect of a SIP request. Authorization policies are outside the scope of this specification, but this specification advises any future authorization work not to assume that messages with valid Identity headers are always good.

14. IANA Considerations

[TBD: update for rfc4474bis or remove?]

This document requests changes to the header and response-code sub-registries of the SIP parameters IANA registry, and requests the creation of two new registries for parameters for the Identity-Info header.

14.1. 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 has been 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: n

14.2. 428 'Use Identity Header' Response Code

This document registers a new SIP response code, which is described in Section 5.2. It is sent when a verifier receives a SIP request that lacks an Identity header in order to indicate that the request should be re-sent with an Identity header. This response code is defined by the following information, which has been 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

14.3. 436 'Bad Identity-Info' Response Code

This document registers a new SIP response code, which is described in Section 5.2. It is used when the Identity-Info header contains a URI that cannot be dereferenced by the verifier (either the URI scheme is unsupported by the verifier, or the resource designated by the URI is otherwise unavailable). This response code is defined by the following information, which has been added to the method and response-code sub-registry under <http://www.iana.org/assignments/sip-parameters>

Response Code Number: 436
Default Reason Phrase: Bad Identity-Info

14.4. 437 'Unsupported Certificate' Response Code

This document registers a new SIP response code, which is described in Section 5.2. It is used when the verifier cannot validate the certificate referenced by the URI of the Identity-Info header, because, for example, the certificate is self-signed, or signed by a root certificate authority for whom the verifier does not possess a root certificate. This response code is defined by the following information, which has been added to the method and response-code sub-registry under <http://www.iana.org/assignments/sip-parameters>

Response Code Number: 437
Default Reason Phrase: Unsupported Certificate

14.5. 438 'Invalid Identity Header' Response Code

This document registers a new SIP response code, which is described in Section 5.2. It is used when the verifier receives a message with an Identity signature that does not correspond to the digest-string calculated by the verifier. This response code is defined by the following information, which has been added to the method and response-code sub-registry under <http://www.iana.org/assignments/sip-parameters>

Response Code Number: 438
Default Reason Phrase: Invalid Identity Header

14.6. Identity-Info Parameters

The IANA has created a new registry for Identity-Info headers. This registry is to be prepopulated with a single entry for a parameter

called 'alg', which describes the algorithm used to create the signature that appears in the Identity header. Registry entries must contain the name of the parameter and the specification in which the parameter is defined. New parameters for the Identity-Info header may be defined only in Standards Track RFCs.

14.7. Identity-Info Algorithm Parameter Values

The IANA has created a new registry for Identity-Info 'alg' parameter values. This registry is to be prepopulated with a single entry for a value called 'rsa-sha1', which describes the algorithm used to create the signature that appears in the Identity header. Registry entries must contain the name of the 'alg' parameter value and the specification in which the value is described. New values for the 'alg' parameter may be defined only in Standards Track RFCs.

15. Acknowledgements

The authors would like to thank the many commentators on this document.

16. Original RFC 4474 Requirements

The following requirements were crafted throughout the development of the mechanism described in this document. They are preserved here for historical reasons.

The mechanism must allow a UAC or a proxy server to provide a strong cryptographic identity assurance in a request that can be verified by a proxy server or UAS.

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

User agents that hold certificates on behalf of their user must be capable of adding this identity assurance to requests.

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 this mechanism in order for an identity assurance to be added to a request in this fashion.

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

In order to provide full replay protection, the mechanism must be capable of protecting the integrity of SIP message bodies (to ensure that media offers and answers are linked to the signaling identity).

It must be possible for a user to have multiple AoRs (i.e., accounts or aliases) that it is authorized to use within 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.

17. Changes from RFC4474

17.1. Motivation for Changes

The original sip-identity drafts that lead to RFC 4474 [RFC4474] were first published in 2002. Since that point many things have changed that impact the design.

- o The DNS root has been signed.
- o SPAM continues to be a problem.
- o It has become clear that B2BUAs will continue to be a major factor in SIP deployments.
- o Multipart MIME has failed as a SIP extension mechanism.
- o Widespread identity providers such as Facebook have emerged.
- o Techniques for non-carrier entities to verify phone numbers and then use them for addressing (such as Apple's iMessage) have been shown to be commercially feasible.
- o Substantial portions of commercial, government, and personal voice communications rely on SIP at some stage in the communications.
- o The cost of operating large databases has fallen and outsourced versions of these databases have become cheaply available.
- o Extensive experience and user research has improved our understanding of how to present security information to users.
- o The world is in the middle of a huge transition to mobile devices. Even the most limited modern mobile devices have user interface and computational capabilities that greatly exceed a 2002-era SIP phone.

The authors believe that the confluence of changing technology, the evolution of mobile devices and internet, and a political will to change make this the right time to consider an change of the scope of 4474 to solve the following problems:

- o Assert strong identity for E.164 numbers such as +1 408 555-1212
- o Continue to assert strong identity for domain scoped names such as alice@example.com
- o Work for calls crossing even the most adverse networks such as the PSTN.
- o Provide reliable information about who is calling before the call is answered to help stop SPAM.
- o Provide reliable information about who you are talking to.
- o Work with evolving non SIP based communications systems such as WebRTC.
- o Potentially, as future work explore organization attributes (e.g., "this is a Bank").

We believe it is possible to solve all of these in a way that is commercially viable, deployable, and provides a delightful user experience.

The core problem in a global identity system with delegated names is understanding who is authorized to make assertions about a given name. The proposal is to solve that problem with a two pronged approach. The design of such a system is outside the scope of this draft, and perhaps of the IETF, but we believe it will have a twofold character:

First, it will delegate responsibility for a number down from a root in a series of delegation sub delegation towards the user. For example, the North American Numbering Plan Administrator assigns a portion of the +1 space to a service provider. That service provider may assign a sub space to a company and that company may assign a number to a user. At each level of delegation, cryptographic credentials could be provided that allow the user to prove the space was delegated to them given some common trust root. This approach is referred to as "delegation" and effectively works from the top down.

The other prong to solving the problem is called "claims" and works via a bottom up approach. The end user of a number basically claims it and some trusted system validates this claim. The validation may

be as simple as sending a SMS to the number or more complicated such as the VIPR system.

The delegation approach creates an easier user experience but is harder to deploy from a business incentive point of view so our approach is to do both and work down from the top and up from the bottom with a meet in the middle approach to coverage of the full name space. For the purposes of the current work, it is envisioned that a certificate authority could encompass both approaches.

Authentication services that possess a credential (whether of the delegation or claim variety) for a telephone number or domain name can, in this mechanism, create one of two types of assertions: basic assertions and reliance assertions. The basic assertion provides replay protection, whereas the reliance assertion provides a broader body protection. Some networks might modify the signaling in ways that impact the reliance assertions but not the other, and thus the reliance assertion is optional.

As in RFC4474, identity assertions are passed in-band in SIP from the caller to the callee for verification. There are however some cases where in-band signaling cannot survive the call path, such as when the call passes through a gateway to the PSTN. This specification assumes that other, out-of-band mechanisms may be used in cases where in-band identity is not carried end-to-end, but those mechanisms are outside the scope of this document.

17.2. Changes to the Identity-Info Header

RFC4474 restricted the subject of the certificate to a domain name, and accordingly the RFC4474 Identity-Info header contains a URI which designates a certificate whose subject (more precisely, subjectAltName) must correspond to the domain of the URI in the From header field value of the SIP request. Per the analysis in [I-D.peterson-secure-origin-ps], this document relaxes that constraint to allow designating an alternative authority for telephone numbers, when telephone numbers appear in the From header field value.

These changes will allow the Identity-Info URI to point to the certificate with authority over the calling telephone number. A verification service will therefore authorize a SIP request when the telephone number in the From header field value agrees with the subject of the certificate. Verification services must of course trust the certificate authority that issued the certificate in question. To implement this change to the Identity-Info header, we must allow for two possibilities for the conveyance of a telephone number in a request: appearing within a tel URI or appearing as the

user portion of a SIP URI. Therefore, we must prescribe the verification service behind in the case where the From header field value URI contains a telephone user part followed by a domain -- which should the verification service expect to find in a certificate?

Future version of this document may explore alternate ways of acquiring credentials, including the use of credentials other than certificates. This might include implementing enough flexibility in the URI to allow a model more like the IdP model described in [I-D.rescorla-rtcweb-generic-idp]; this could be useful as RTCWeb sees increasing deployment. We also should consider any implications of the signing of the DNSSEC root and the DANE specifications to the existing Identity-Info uses with domain name. At a high level, it is not expected that the proposed changes will radically alter the semantics of Identity-Info.

17.3. Changes to the Identity Header

Per the analysis in [I-D.peterson-secure-origin-ps], this document changes the signature mechanism that RFC44474 specified for the Identity header: in particular, to replace this signature mechanism with one that is more likely to survive end-to-end in SIP networks where intermediaries act as back-to-back user agents rather than proxy servers.

To accomplish this, we here create two distinct signatures within SIP requests: a basic assurance and a reliance assurance. The basic assurance prevents impersonation attacks by providing a signature over the From header field value and certain other headers which will allow a verification service to detect some cut-and-paste attacks. The reliance assurance protects against attackers changing other parameters of the call: these include the entirety of the messaging body, including the target IP address and ports in SDP which, if unprotected, can allow an attacker to succeed with more sophisticated cut-and-paste attacks. Authentication services behavior would change to allow them to decide, based on their policy in a deployment environment, whether only the basic assurance can realistically survive network transit, or if the reliance assurance should be available. There are several similar design choices in this space to consider, and some analysis will be required to identify the best option.

In cases where the From header field value of a SIP request contains a SIP URI with a telephone number user part, we will also consider replay assurance canonicalizations that do not cover the domain portion of the URI.

[TBD: in order to preserve critical security parameters even in adverse network conditions, should the basic assurance integrity protection must always cover security parameters of the SDP required to negotiate media-level security? There may be other exception cases, or extensibility mechanisms, worth considering here.]

18. References

18.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2818] Rescorla, E., "HTTP Over TLS", RFC 2818, May 2000.
- [RFC3261] 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.
- [RFC3280] Housley, R., Polk, W., Ford, W., and D. Solo, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 3280, April 2002.
- [RFC3323] Peterson, J., "A Privacy Mechanism for the Session Initiation Protocol (SIP)", RFC 3323, November 2002.
- [RFC3370] Housley, R., "Cryptographic Message Syntax (CMS) Algorithms", RFC 3370, August 2002.
- [RFC3548] Josefsson, S., "The Base16, Base32, and Base64 Data Encodings", RFC 3548, July 2003.
- [RFC3893] Peterson, J., "Session Initiation Protocol (SIP) Authenticated Identity Body (AIB) Format", RFC 3893, September 2004.
- [RFC4234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", RFC 4234, October 2005.

18.2. Informative References

- [I-D.cooper-iab-secure-origin-00]
Cooper, A., Tschofenig, H., Peterson, J., and B. Aboba, "Secure Call Origin Identification", draft-cooper-iab-secure-origin-00 (work in progress), November 2012.

- [I-D.peterson-secure-origin-ps]
Peterson, J., Schulzrinne, H., and H. Tschofenig, "Secure Origin Identification: Problem Statement, Requirements, and Roadmap", draft-peterson-secure-origin-ps-00 (work in progress), May 2013.
- [I-D.peterson-sipping-retarget]
Peterson, J., "Retargeting and Security in SIP: A Framework and Requirements", draft-peterson-sipping-retarget-00 (work in progress), February 2005.
- [I-D.rescorla-callerid-fallback]
Rescorla, E., "Secure Caller-ID Fallback Mode", draft-rescorla-callerid-fallback-00 (work in progress), May 2013.
- [I-D.rescorla-rtcweb-generic-idp]
Rescorla, E., "RTCWEB Generic Identity Provider Interface", draft-rescorla-rtcweb-generic-idp-01 (work in progress), March 2012.
- [RFC2234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", RFC 2234, November 1997.
- [RFC2585] Housley, R. and P. Hoffman, "Internet X.509 Public Key Infrastructure Operational Protocols: FTP and HTTP", RFC 2585, May 1999.
- [RFC3263] Rosenberg, J. and H. Schulzrinne, "Session Initiation Protocol (SIP): Locating SIP Servers", RFC 3263, June 2002.
- [RFC3325] 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.
- [RFC3761] Faltstrom, P. and M. Mealling, "The E.164 to Uniform Resource Identifiers (URI) Dynamic Delegation Discovery System (DDDS) Application (ENUM)", RFC 3761, April 2004.
- [RFC3966] Schulzrinne, H., "The tel URI for Telephone Numbers", RFC 3966, December 2004.
- [RFC4474] Peterson, J. and C. Jennings, "Enhancements for Authenticated Identity Management in the Session Initiation Protocol (SIP)", RFC 4474, August 2006.

- [RFC4475] Sparks, R., Hawrylyshen, A., Johnston, A., Rosenberg, J., and H. Schulzrinne, "Session Initiation Protocol (SIP) Torture Test Messages", RFC 4475, May 2006.
- [RFC6919] Barnes, R., Kent, S., and E. Rescorla, "Further Key Words for Use in RFCs to Indicate Requirement Levels", RFC 6919, April 1 2013.

Authors' Addresses

Jon Peterson
NeuStar

Email: jon.peterson@neustar.biz

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

Email: fluffy@iii.ca

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

Phone: +1 650 678 2350
Email: ekr@rtfm.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 18, 2014

J. Peterson
NeuStar
C. Jennings
Cisco
E. Rescorla
RTFM, Inc.
February 14, 2014

Authenticated Identity Management in the Session Initiation Protocol
(SIP)
draft-jennings-stir-rfc4474bis-01.txt

Abstract

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

Status of This Memo

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

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

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

This Internet-Draft will expire on August 18, 2014.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents

(<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
2. Background	3
3. Overview of Operations	5
4. Signature Generation and Validation	6
4.1. Authentication Service Behavior	6
4.1.1. Intermediary Authentication Services	9
4.2. Verifier Behavior	10
4.3. Identity within a Dialog and Retargeting	11
5. Credentials	12
5.1. Credential Use by the Authentication Service	12
5.2. Credential Use by the Verification Service	13
5.3. Handling Identity-Info URIs	14
5.4. Credential Systems	14
6. Identity Types	15
6.1. Telephone Numbers	15
6.2. Usernames with Domain Names	16
7. Header Syntax	17
8. Examples	21
9. Privacy Considerations	21
10. Security Considerations	22
10.1. Handling of digest-string Elements	22
10.2. Securing the Connection to the Authentication Service	24
10.3. Authorization and Transitional Strategies	25
10.4. Display-Names and Identity	26
11. IANA Considerations	27
11.1. Header Field Names	27
11.2. 428 'Use Identity Header' Response Code	27
11.3. 436 'Bad Identity-Info' Response Code	27
11.4. 437 'Unsupported Credential' Response Code	28
11.5. 438 'Invalid Identity Header' Response Code	28
11.6. Identity-Info Parameters	28
11.7. Identity-Info Algorithm Parameter Values	28
12. Acknowledgments	29
13. Changes from RFC4474	29
14. Informative References	29
Authors' Addresses	31

1. Introduction

This document provides enhancements to the existing mechanisms for authenticated identity management in the Session Initiation Protocol (SIP, RFC 3261 [1]). An identity, for the purposes of this document, is defined as either a SIP URI, commonly a canonical address-of-record (AoR) employed to reach a user (such as 'sip:alice@atlanta.example.com'), or a telephone number, which can be represented as either a TEL URI or as the user portion of a SIP URI.

RFC 3261 [1] stipulates several places within a SIP request where 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 appropriately, in the absence of some sort of cryptographic authentication mechanism.

RFC 3261 [1] specifies a number of security mechanisms that can be employed by SIP user agents (UAs), including Digest, Transport Layer Security (TLS), and S/MIME (implementations may support other security schemes as well). However, few SIP user agents today support the end-user certificates necessary to authenticate themselves (via S/MIME, for example), and furthermore Digest authentication is limited by the fact that the originator and destination must share a prearranged secret. It is desirable for SIP user agents to be able to send requests to destinations with which 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 the person's displayed calling party number (and/or Caller-ID) is accurate. A cryptographic approach, like the one described in this document, can provide a much stronger and less spoofable assurance of identity than the telephone network provides today.

2. Background

The usage of many SIP applications and services is governed by authorization policies. These policies may be automated, or they may be applied manually by humans. An example of the latter would be an Internet telephone application that displays the calling party number (and/or Caller-ID) of a caller, which a human may review to make a policy decision before answering a call. An example of the former would be a voicemail service that compares the identity of the caller to a whitelist before determining whether it should allow the caller access to recorded messages. In both of these cases, attackers might attempt to circumvent these authorization policies through impersonation. Since the primary identifier of the sender of a SIP request, the From header field, can be populated arbitrarily by the

controller of a user agent, impersonation is very simple today. The mechanism described in this document provides a strong identity system for SIP requests in which authorization policies cannot be circumvented by impersonation.

This document proposes an authentication architecture for SIP in which requests are processed by a logical authentication service that may be implemented as part of a user agent or as a proxy server. Once a message has been authenticated, the service then adds new cryptographic information to requests to communicate to other SIP entities that the sending user has been authenticated and its use of the From header field has been authorized.

But authorized by whom? Identities are issued to users by authorities. When a new user becomes associated with example.com, the administrator of the SIP service for that domain will issue them an identity in that namespace, such as alice@example.com. Alice may then send REGISTER requests to example.com that make her user agents eligible to receive requests for sip:alice@example.com. In some cases, Alice may be the owner of the domain herself, and may issue herself identities as she chooses. But ultimately, it is the controller of the SIP service at example.com that must be responsible authorizing the use of names in the example.com domain. Therefore, the credentials needed to prove this authorization must ultimately derive from the domain owner: either a user agent gives requests to the domain name owner in order for them to be signed by the domain owner's credentials, or the user agent must possess credentials that prove in some fashion that the domain owner has given the user agent the right to a name.

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

This document specifies a means of sharing a cryptographic assurance of end-user SIP identity in an interdomain or intradomain context that is based on the authentication service adding a SIP header, the Identity header. In order to assist in the validation of this

assurance, this specification also describes an Identity-Info header that can be used by the recipient of a request to recover the credentials of the signer. Note that the scope of this document is limited to providing this identity assurance for SIP requests; solving this problem for SIP responses is outside the scope of this work.

This specification allows either a user agent or a proxy server to provide identity services and to verify identities. To maximize end-to-end security, it is obviously preferable for end-users to acquire their own credentials; if they do, their user agents can act as an authentication service. However, end-user credentials may be neither practical nor affordable, given the potentially large number of SIP user agents (phones, PCs, laptops, PDAs, gaming devices) that may be employed by a single user. In such environments, synchronizing keying material across multiple devices may be very complex and requires quite a good deal of additional endpoint behavior. Managing several credentials for the various devices could also be burdensome. This trade-off needs to be understood by implementers of this specification.

3. 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 assert the identity that is populated in the From header field. This value may be Alice's AoR, or in other cases it may be some different value that the proxy server has authority over, such as a telephone number. It then computes a hash over some particular headers, including the From header field (and optionally the body) of the message. This hash is signed with the appropriate credential (example.com, in the sip:alice@example.com case) and inserted in a new header field in the SIP message, the 'Identity' header.

The proxy, as the holder of the private key for 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) that appears in the From header field. The proxy also inserts a companion header field, Identity-Info, that tells Bob how to acquire keying material necessary to validate its credentials, 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 validate that the authority over the identity in the From header field authenticated the user, and permitted the user to assert that From header field value. This same validation operation may be performed by Bob's user agent server (UAS).

4. Signature Generation and Validation

4.1. Authentication Service Behavior

This document specifies a role for SIP entities called an authentication service. The authentication service role can be instantiated by an intermediary such as a proxy server or a user agent. Any entity that instantiates the authentication service role MUST possess the private key of one or more credentials that can be used to sign for a domain or a telephone number (see Section 5.1). Intermediaries that instantiate this role MUST be capable of authenticating one or more SIP users who can register for that identity. Commonly, this role will be instantiated by a proxy server, since these entities are more likely to have a static hostname, hold corresponding credentials, and have access to SIP registrar capabilities that allow them to authenticate users. It is also possible that the authentication service role might be instantiated by an entity that acts as a redirect server, but that is left as a topic for future work.

SIP entities that act as an authentication service MUST add a Date header field to SIP requests if one is not already present (see Section 7 for information on how the Date header field assists verifiers).

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

Step 1:

The 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 SIP Secure (SIPS) URI, and the user portion is not a telephone number, the

authentication service MUST extract the hostname portion of the identity field and compare it to the domain(s) for which it is responsible (following the procedures in RFC 3261 [1], Section 16.4), used by a proxy server to determine the domain(s) for which it is responsible). If the identity field uses the TEL URI scheme, or the identity field is a SIP or SIPS URI with a telephone number in the user portion, the authentication service determines whether or not it is responsible for this telephone number; see Section 6.1 for more information. If the authentication service is not authoritative for the identity in question, it SHOULD process and forward the request normally, but it MUST NOT following the steps below to add an Identity header; see below for more information on authentication service handling of an existing Identity header. [where?]

Step 2:

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

If the authentication service is instantiated by a SIP intermediary (proxy server), it may 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 [1], Section 22.3). Note that if that proxy server is maintaining a TLS connection with the client over which the client had previously authenticated itself using Digest authentication, the identity value obtained from that previous authentication step can be reused without an additional Digest challenge.

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 credential, or preferably by providing a password that unlocks said private key.

Authorization of the use of a particular username or telephone number in the user part of the From header field is a matter of local policy for the authentication service, see Section 5.1 for more information.

Note that this check is performed only 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').

Authentication services MAY check and validate 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 response code. The reason phrase should indicate the nature of the problem; for example, "Inappropriate Display Name". However, the display-name is not always present, and in many environments the requisite operational procedures for display-name validation may not exist. For more information, see Section 10.4.

Step 3:

The authentication service SHOULD ensure that any preexisting Date header in the request is accurate. Local policy can dictate precisely how accurate the Date must be; a RECOMMENDED maximum discrepancy of ten minutes will ensure that the request is unlikely to upset any verifiers. If the Date header contains a time different by more than ten minutes from the current time noted by the authentication service, the authentication service SHOULD reject the request. This behavior is not mandatory because a user agent client (UAC) could only exploit the Date header in order to cause a request to fail verification; the Identity header is not intended to provide a source of non-repudiation or a perfect record of when messages are processed. Finally, the authentication service MUST verify that the Date header falls within the validity period of its credential. For more information on the security properties associated with the Date header field value, see Section 7.

[TBD: Should consider a lower threshold than ten minutes? With the removal of other elements from the sig, that's a lot of leeway.]

Step 4:

The authentication service MAY form an identity-reliance signature and add an Identity-Reliance header to the request containing this signature. The Identity-Reliance header provides body security properties that are useful for non-INVITE transactions, and in environments where body security of INVITE transactions is necessary. Details on the generation of this header is provided in Section 7. If the authentication service is adding an Identity-Reliance header, it MUST also add a Content-Length header field to SIP requests if one is not already present; this can help verifiers to double-check that they are hashing exactly as many bytes of message-body as the authentication service when they verify the message.

Step 5:

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 credential can be acquired; see Section 5.3 for more on credential acquisition. Details on the syntax of both of these headers are provided in Section 7.

Finally, the authentication service MUST forward the message normally.

4.1.1.1. Intermediary Authentication Services

In cases where a user agent does not possess its own credentials to sign an Identity header, the user agent can send its request through an intermediary that will provide a signed Identity header based on the contents of the request. This requires, among other things, that intermediaries have some means of authenticating the user agents sending requests.

All RFC 3261 [1] compliant user agents support Digest authentication, which utilizes a shared secret, as a means for authenticating themselves to a SIP registrar. Registration allows a user agent to express that it is an appropriate entity to which requests should be sent for a particular SIP AoR URI (e.g., 'sip:alice@atlanta.example.com'). For such SIP URIs, by the definition of identity used in this document, registration proves the identity of the user to a registrar. Similar checks might be performed for telephone numbers as identities. This is of course only one manner in which a domain might determine how a particular user is authorized to populate the From header field; as an aside, for other sorts of URIs in the From (like anonymous URIs), other authorization policies would apply.

RFC 3261 [1] already describes an intermediary architecture very similar to the one proposed in this document 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.

This specification assumes that UACs will have an appropriate means to discover an authentication service that can sign with a credential

corresponding to the UAC's identity. Most likely, this information will simply be provisioned in UACs.

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 RFC 3325 [9]. 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.

4.2. Verifier Behavior

This document specifies a logical role for SIP entities called a verification service, or verifier. When a verifier receives a SIP message containing an Identity header, it inspects the signature to verify the identity of the sender of the message. Typically, the results of a verification are provided as input to an authorization process that is outside the scope of this document. If an Identity header is not present in a request, and one is required by local policy (for example, based on a per-sending-domain policy, or a per-sending-user policy), then a 428 'Use Identity Header' response MUST be sent.

In order to verify the identity of the sender of a message, an entity acting as a verifier MUST perform the following steps, in the order here specified.

Step 1:

In order to determine whether the signature for the URI in the From header field value should be over the entire URI or just a canonicalized telephone number, the verification service must follow the process described in Section 6.1. That section also describes the procedures the verification service must follow to determine if the signer is authoritative for a telephone number. For domains, the verifier MUST follow the process described in Section 6.2 to determine if the signer is authoritative for the URI in the From header field.

Step 2:

The verifier must first ensure that it possesses the proper keying material to validate the signature in the Identity header field. See Section 5.2 for more information on these procedures.

Step 3:

The verifier MUST verify the signature in the Identity header field, following the procedures for generating the hashed digest-string described in Section 7. If a verifier determines that the signature on the message does not correspond to the reconstructed digest-string, then a 438 'Invalid Identity Header' response MUST be returned.

Step 4:

If the request contains an Identity-Reliance header, the verifier SHOULD verify the signature in the Identity-Reliance header field, following the procedures for generating the hashed reliance-digest-string described in Section 7. If a verifier determines that the signature on the message does not correspond to the reconstructed digest-string, then a 438 'Invalid Identity Header' response SHOULD be returned.

Step 5:

The verifier MUST validate the Date header in the manner described in Section 10.1; recipients that wish to verify Identity signatures MUST support all of the operations described there. It must furthermore ensure that the value of the Date header falls within the validity period of the credential used to sign the Identity header.

4.3. Identity within a Dialog and Retargeting

The mechanism in this document provides a signature over the URI in the To header field value. The recipient of a request must compare that value to their own identity in order to determine whether or not the identity information in this call might have been replayed. Retargeting, however, complicates this evaluation.

Retargeting is broadly defined as the alteration of the Request-URI by intermediaries. More specifically, retargeting supplants the original target URI with one that corresponds to a different user, potentially a user that is not authorized to register under the original target URI. By this definition, retargeting does not include translation of the Request-URI to a contact address of an endpoint that has registered under the original target URI.

When a request is retargeted, it may reach a SIP endpoint whose user is not identified by the URI designated in the To header field value. Moreover, the value in the To header field of a dialog-forming request is used as the From header field of requests sent in the backwards direction during the dialog, and is accordingly the header that would be signed by an authentication service for requests sent in the backwards direction. But in retargeting cases, if the URI in

the From header does not identify the sender of the request in the backwards direction, then clearly it would be inappropriate to provide an Identity signature over that From header. As specified above, if the authentication service is not responsible for the domain in the From header field of the request, it MUST NOT add an Identity header to the request, and it should process/forward the request normally.

Any means of anticipating retargeting, and so on, is outside the scope of this document, and likely to have equal applicability to response identity as it does to requests in the backwards direction within a dialog. Consequently, no special guidance is given for implementers here regarding the 'connected party' problem; authentication service behavior is unchanged if retargeting has occurred for a dialog-forming request. Ultimately, the authentication service provides an Identity header for requests in the backwards dialog when the user is authorized to assert the identity given in the From header field, and if they are not, an Identity header is not provided.

For further information on the problems of response identity see [17].

5. Credentials

5.1. Credential Use by the Authentication Service

In order to act as an authentication service, a SIP entity must have access to the private keying material of one or more credentials that cover URIs, domain names or telephone numbers. These credentials may represent authority over only a single name (such as `alice@example.com`), an entire domain (such as `example.com`), or potentially a set of domains. Similarly, a credential may represent authority over a single telephone number or a range of telephone numbers. The way that the scope of a credential is expressed is specific to the credential mechanism.

Authorization of the use of a particular username or telephone number in the user part of the From header field is a matter of local policy for the authentication service, one that depends greatly on the manner in which authentication is performed. For non-telephone number user parts, one policy might be as follows: the username given in the 'username' parameter of the Proxy-Authorization header MUST correspond exactly to the username in the From header field of the SIP message. However, there are many cases in which this is too limiting or inappropriate; a realm might use 'username' parameters in Proxy-Authorization that do not correspond to the user-portion of SIP From headers, or a user might manage multiple accounts in the same

administrative domain. In this latter case, a domain might maintain a mapping between the values in the 'username' parameter of Proxy-Authorization and a set of one or more SIP URIs that might legitimately be asserted for that 'username'. For example, the username can correspond to the 'private identity' as defined in Third Generation Partnership Project (3GPP), in which case the From header field can contain any one of the public identities associated with this private identity. In this instance, another policy might be as follows: the URI in the From header field MUST correspond exactly to one of the mapped URIs associated with the 'username' given in the Proxy-Authorization header. This is a suitable approach for telephone numbers in particular. Various exceptions to such policies might arise for cases like anonymity; if the AoR asserted in the From header field uses a form like 'sip:anonymous@example.com', then the 'example.com' proxy should authenticate that the user is a valid user in the domain and insert the signature over the From header field as usual.

5.2. Credential Use by the Verification Service

In order to act as a verification service, a SIP entity must have a way to acquire and retain credentials for authorities over particular URIs, domain names and/or telephone numbers. The Identity-Info header (as described in the next section) is supported by all verification service implementations to create a baseline means of credential acquisition. Provided that the credential used to sign a message is not previously known to the verifier, SIP entities SHOULD discover this credential by dereferencing the Identity-Info header, unless they have some more efficient implementation-specific way of acquiring certificates. If the URI scheme in the Identity-Info header cannot be dereferenced, then a 436 'Bad Identity-Info' response MUST be returned.

Verification service implementations supporting this specification SHOULD have some means of retaining credentials (in accordance with normal practices for credential lifetimes and revocation) in order to prevent themselves from needlessly downloading the same credential every time a request from the same identity is received. Credentials cached in this manner may be indexed in accordance with local policy: for example, by their scope, or the URI given in the Identity-Info header field value.

[TBD: Should we add some kind of hash or similar indication to the Identity-Info header to make it easier for verifiers to ascertain that they already possess a credential without dereferencing the URI?]

5.3. Handling Identity-Info URIs

An Identity-Info header MUST contain a URI which dereferences to a resource which contains the public key components of the credential used by the authentication service to sign a request. Much as is the case with the trust anchor(s) required for deployments of this specification, it is essential that a URI in the Identity-Info header be dereferencable by any entity that could plausibly receive the request. For common cases, this means that the URI must be dereferencable by any entity on the public Internet. In constrained deployment environments, a service private to the environment might be used instead.

Beyond providing a means of accessing credentials for an identity, the Identity-Info header further services a means of differentiating which particular credential was used to sign a request, when there are potentially multiple authorities eligible to sign. For example, imagine a case where a domain implements the authentication service role for example.com, and a user agent belonging to Alice has acquired a credential for alice@example.com. Either would be eligible to sign a SIP request from alice@example.com. Verification services however need a means to differentiate which one performed the signature. The Identity-Info header performs that function.

5.4. Credential Systems

This document makes no specific recommendation for the use of any credential system. Today, there are two primary credential systems in place for proving ownership of domain names: certificates (e.g., X.509 v3, see [8]) and the domain name system itself (e.g., DANE, see [10]). It is envisioned that either could be used in the SIP context: an Identity-Info header could for example give an HTTP URL of the form 'application/pkix-cert' pointing to a certificate (following the conventions of [3]). The Identity-Info headers may use the DNS URL scheme (see [11]) to indicate keys in the DNS.

While no comparable public credentials exist for telephone numbers, either approach could be applied to telephone numbers. A credential system based on certificates is given in draft-peterson-stir-certificates [TBD - fix after submitting]. One based on the domain name system is given in [18].

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

which URIs schemes the credential will use in the Identity-Info header, and any special procedures required to dereference the URIs

how the verifier can learn the scope of the credential.

any special procedures required to extract keying material from the resources designated by the URI

any algorithms that would appear in the Identity-Info "alg" parameter other than 'rsa-sha256.' Note that per the IANA Considerations of this document (Section 11.7), new algorithms can only be specified by Standards Action.

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

Note that credential systems must address key lifecycle management concerns: were a domain to change the credential available at the Identity-Info URI before a verifier evaluates a request signed by an authentication service, this would cause obvious verifier failures. When a rollover occurs, authentication services SHOULD thus provide new Identity-Info URIs for each new credential, and SHOULD continue to make older key acquisition URIs available for a duration longer than the plausible lifetime of a SIP message (an hour would most likely suffice).

[TBD: What will the normative language here be? Support for which mechanisms?]

6. Identity Types

6.1. Telephone Numbers

Since many SIP applications provide a Voice over IP (VoIP) service, telephone numbers are commonly used as identities in SIP deployments. In order for telephone numbers to be used with the mechanism described in this document, authentication services must enroll with an authority that issues credentials for telephone numbers or telephone number ranges, and verification services must trust the authority employed by the authentication service that signs a request. Enrollment procedures and credential management are outside the scope of this document.

Given the existence of such authorities, authentication and verification services must identify when a request should be signed by an authority for a telephone number, and when it should be signed

by an authority for a domain. Telephone numbers most commonly appear in SIP header field values in the username portion of a SIP URI (e.g., 'sip:+17005551008@chicago.example.com;user=phone'). The user part of that URI conforms to the syntax of the TEL URI scheme (RFC 3966 [5]). 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 both of these cases, it's clear that the signer must have authority over the telephone number, not the domain name of the SIP URI. It is also possible, however, for requests to contain a URI like 'sip:7005551000@chicago.example.com'. It may be non-trivial for a service to ascertain in this case whether the URI contains a telephone number or not.

To address this problem, the authentication service and verification service both must perform the following canonicalization procedure on any SIP URI they inspect which contains a wholly numeric user part.

[TBD canonicalization algorithm - drop the characters, +'s, assess if its a valid local number (if so, append country code), etc]

[TBD define tn-spec here for ABNF purposes]

If the result of this procedure forms a complete telephone number, that number is used for the purpose of creating and signing the digest-string by both the authentication service and verification service. If the result does not form a complete telephone number, the authentication service and verification service should treat the entire URI as a SIP URI, and apply a domain signature per the procedures in Section 6.2.

In the longer term, it is possible that some directory or other discovery mechanism 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.

6.2. Usernames with Domain Names

When a verifier processes a request containing an Identity-Info header with a domain signature, it must compare the domain portion of the URI in the From header field of the request with the domain name that is the subject of the credential acquired from the Identity-Info header. While this might seem that this should be a straightforward process, it is complicated by two deployment realities. In the first place, credentials have varying ways of describing their subjects, and may indeed have multiple subjects, especially in 'virtual hosting' cases where multiple domains are managed by a single application. Secondly, some SIP services may delegate SIP functions

to a subordinate domain and utilize the procedures in RFC 3263 [2] that allow requests for, say, 'example.com' to be routed to 'sip.example.com'. As a result, a user with the AoR 'sip:jon@example.com' may process requests through a host like 'sip.example.com', and it may be that latter host that acts as an authentication service.

To meet the second of these problems, a domain that deploys an authentication service on a subordinate host **MUST** be willing to supply that host with the private keying material associated with a credential whose subject is a domain name that corresponds to the domain portion of the AoRs that the domain distributes to users. Note that this corresponds to the comparable case of routing inbound SIP requests to a domain. When the NAPTR and SRV procedures of RFC 3263 are used to direct requests to a domain name other than the domain in the original Request-URI (e.g., for 'sip:jon@example.com', the corresponding SRV records point to the service 'sip1.example.org'), the client expects that the certificate passed back in any TLS exchange with that host will correspond exactly with the domain of the original Request-URI, not the domain name of the host. Consequently, in order to make inbound routing to such SIP services work, a domain administrator must similarly be willing to share the domain's private key with the service. This design decision was made to compensate for the insecurity of the DNS, and it makes certain potential approaches to DNS-based 'virtual hosting' unsecurable for SIP in environments where domain administrators are unwilling to share keys with hosting services.

A verifier **MUST** evaluate the correspondence between the user's identity and the signing credential by following the procedures defined in RFC 2818 [7], Section 3.1. While RFC 2818 [7] deals with the use of HTTP in TLS and is specific to certificates, the procedures described are applicable to verifying identity if one substitutes the "hostname of the server" in HTTP for the domain portion of the user's identity in the From header field of a SIP request with an Identity header.

7. Header Syntax

This document specifies three SIP headers: Identity, Identity-Reliance and Identity-Info. Each of these headers can appear only once in a SIP request; Identity-Reliance is **OPTIONAL**, while Identity and Identity-Info are **REQUIRED** for securing requests with this specification. The grammar for these three headers is (following the ABNF [12] in RFC 3261 [1]):

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

```
Identity-Reliance = "Identity-Reliance" HCOLON signed-identity-reliance-diges
t signed-identity-reliance-digest = LDQUOT 32LHEX RDQUOT
```

```
Identity-Info = "Identity-Info" HCOLON ident-info
                *( SEMI ident-info-params )
ident-info = LAQUOT absoluteURI RAQUOT
ident-info-params = ident-info-alg / ident-info-extension
ident-info-alg = "alg" EQUAL token
ident-info-extension = generic-param
```

[TBD: The version has the Identity-Reliance header covered under the Identity signature. It is also possible to do this the other way around, where the base Identity signature is generated first, and Identity-Reliance would cover both the Identity header and the body. This is a trade-off of whether the authentication service should decide whether Identity-Reliance is needed or if the verification service should decide. These have different properties, and some investigation would be needed to decide between them.]

The signed-identity-reliance-digest is a signed hash of a canonical string generated from certain components of a SIP request. Creating this hash and the Identity-Reliance header field to contain it is OPTIONAL, and its usage is a matter of local policy for authentication services. To create the contents of the signed-identity-reliance-digest, the following element of a SIP message MUST be placed in a bit-exact string:

The body content of the message with the bits exactly as they are in the message (in the ABNF for SIP, the message-body). This includes all components of multipart message bodies. 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.

[TBD: Explore alternatives to including the whole body for INVITE requests; should there be a special case for security parameters that would appear in SDP?]

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 vertical line, "|" or %x7C, character:

First, the identity. If the user part of the AoR in the From header field of the request contains a telephone number, then the canonicalization of that number goes into the first slot (see Section 6.1). Otherwise, the first slot contains the AoR of the UA sending the message, or addr-spec of the From header field.

Second, the target. If the user part of the AoR in the To header field of the request contains a telephone number, then the canonicalization of that number goes into the second slot (see Section 6.1). Otherwise, the second slot contains the addr-spec component of the To header field, which is the AoR to which the request is being sent.

Third, the request method.

Fourth, the Date header field, with exactly one space each for each SP and the weekday and month items case set as shown in the BNF of RFC 3261 [1]. RFC 3261 specifies that the BNF for weekday and month is a choice amongst a set of tokens. The RFC 4234 [12] rules for the BNF specify that tokens are case sensitive. However, when used to construct the canonical string defined here, the first letter of each week and month MUST be capitalized, and the remaining two letters must be lowercase. This matches the capitalization provided in the definition of each token. All requests that use the Identity mechanism MUST contain a Date header.

Fifth, the Identity-Reliance header field value, if there is an Identity-Reliance field in the request. If the message has no body, or no Identity-Reliance header, then the fifth slot will be empty, and the final "|" 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 10 and [4]. The precise formulation of this digest-string is, therefore (following the ABNF[12] in RFC 3261 [1]):

```
digest-string = addr-spec / tn-spec "|" addr-spec / tn-spec "|"
                Method "|" SIP-date "|" [ signed-identity-reliance-digest ]
```

For the definition of 'tn-spec' see Section 6.1.

After the digest-string or reliance-digest-string is formed, each MUST be hashed and signed with the certificate of authority over the identity. The hashing and signing algorithm is specified by the 'alg' parameter of the Identity-Info header (see below for more information on Identity-Info header parameters). This document

defines only one value for the 'alg' parameter: 'rsa-sha256'; further values MUST be defined in a Standards Track RFC, see Section 14.7 for more information. All implementations of this specification MUST support 'rsa-sha256'. When the 'rsa-sha256' algorithm is specified in the 'alg' parameter of Identity-Info, the hash and signature MUST be generated as follows: compute the results of signing this string with sha1WithRSAEncryption as described in RFC 3370 [13] and base64 encode the results as specified in RFC 3548 [14]. A 2048-bit or longer RSA key MUST be used. The result of the digest-string hash is placed in the Identity header field; the optional reliance-digest-string hash goes in the Identity-Reliance header. For detailed examples of the usage of this algorithm, see Section 8.

The 'absoluteURI' portion of the Identity-Info header MUST contain a URI; see Section 5.3 for more on choosing how to advertise credentials through Identity-Info.

This document adds (or amends) the following entries to Table 2 of RFC 3261 [1] (this repeats the registrations of RFC4474):

Header field	where	proxy	ACK	BYE	CAN	INV	OPT	REG
-----	----	-----	----	----	----	----	----	----
Identity	R	a	o	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	o
			SUB	NOT	REF	INF	UPD	PRA
			---	---	---	---	---	---
			o	o	o	o	o	o
Header field	where	proxy	ACK	BYE	CAN	INV	OPT	REG
-----	----	-----	----	----	----	----	----	----
Identity-Reliance	R	a	o	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 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 Identity and Identity-Info header MUST NOT appear in CANCEL. Note as well that the use of Identity with REGISTER is consequently a subject for future study, although it is left as optional here for forward-compatibility reasons.

8. Examples

9. Privacy Considerations

The identity mechanism presented in this document is compatible with the standard SIP practices for privacy described in RFC 3323 [15]. A SIP proxy server can act both as a privacy service and as an authentication service. Since a user agent can provide any From header field value that the authentication service is willing to authorize, there is no reason why private SIP URIs that contain legitimate domains (e.g., sip:anonymous@example.com) cannot be signed by an authentication service. The construction of the Identity header is the same for private URIs as it is for any other sort of URIs.

Note, however, that for using anonymous SIP URIs, an authentication service must possess a certificate corresponding to the host portion of the addr-spec of the From header field of the request; accordingly, using domains like 'anonymous.invalid' will not be possible for privacy services that also act as authentication services. The assurance offered by the usage of anonymous URIs with a valid domain portion is "this is a known user in my domain that I have authenticated, but I am keeping its identity private". The use of the domain 'anonymous.invalid' entails that no corresponding authority for the domain can exist, and as a consequence, authentication service functions are meaningless.

RFC 3325 [9] defines the "id" priv-value token, which is specific to the P-Asserted-Identity header. The sort of assertion provided by the P-Asserted-Identity header is very different from the Identity header presented in this document. It contains additional information about the sender of a message that may go beyond what appears in the From header field; P-Asserted-Identity holds a definitive identity for the sender that is somehow known to a closed network of intermediaries 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.

Finally, note that unlike RFC 3325 [9], the mechanism described in this specification adds no information to SIP requests that has privacy implications.

10. Security Considerations

10.1. Handling of digest-string Elements

This document describes a mechanism that provides a signature over the Date header field, and either the whole or part of the To and From header fields of SIP requests, as well as optional protections for the message body. 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 replayed in cut-and-paste attacks. In general, the considerations related to the security of these headers are the same as those given in RFC 3261 [1] for including headers in tunneled 'message/sip' MIME bodies (see Section 23 in particular). The following section details the individual security properties obtained by including each of these header fields within the signature; collectively, this set of header fields provides the necessary properties to prevent impersonation.

The From header field indicates the identity of the sender of the message, and the SIP address-of-record URI, or an embedded telephone number, in the From header field is the identity of a SIP user, for the purposes of this document. 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 cut-and-paste attacks in which an Identity header from legitimate request for one user is cut-and-pasted into a request for a different user; second, it preserves the starting URI scheme of the request, which helps prevent downgrade attacks against the use of SIPS.

The Date header field provides replay protection, as described in RFC 3261 [1], 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). The result of this is that if an Identity header is replayed within the Date interval, verifiers will recognize that it is invalid; if an Identity header is replayed after the Date interval, verifiers will recognize that it is invalid because the Date is stale.

Without the method, an INVITE request could be cut- and-pasted by an attacker and transformed into a MESSAGE 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.

RFC4474 originally had protections for the Contact, Call-ID and CSeq. These are removed from RFC4474bis. The absence of these header values creates some opportunities for determined attackers to impersonate based on cut-and-paste attacks; however, the absence of these headers does not seem impactful to preventing against the simple unauthorized claiming of a From header field value, which is the primary scope of the current document.

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

This mechanism also provides an optional signature over the bodies of SIP requests. This can help to protect non-INVITE transactions such as MESSAGE or NOTIFY, as well as INVITEs in those environments where intermediaries do not change SDP. While this is not strictly necessary to prevent the impersonation attacks, there is little purpose in establishing the identity of the user that originated a SIP request if this assurance is not coupled with a comparable assurance over the contents of the message. There are furthermore some baiting attacks (where the attacker receives a request from the target and reoriginates it to a third party) that might not be prevented by only a signature over the From, To and Date, but could be prevented by securing SDP. Note, however, that this is not perfect end-to-end security. The authentication service itself, when instantiated at an intermediary, could conceivably change the body (and SIP headers, for that matter) before providing a signature.

Thus, while this mechanism reduces the chance that a replayer or man-in-the-middle will modify bodies, it does not eliminate it entirely. Since it is a foundational assumption of this mechanism that the users trust 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.

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

One additional respect in which the Identity-Info header cannot protect itself is the 'alg' parameter. The 'alg' parameter is not included in the digest-string, and accordingly, a man-in-the-middle might attempt to modify the 'alg' parameter. Once again, it is important to note that preventing men-in-the-middle is not the primary impetus for this mechanism. Moreover, changing the 'alg' would at worst result in some sort of bid-down attack, and at best cause a failure in the verifier. Note that only one valid 'alg' parameter is defined in this document and that thus there is currently no weaker algorithm to which the mechanism can be bid down. 'alg' has been incorporated into this mechanism for forward-compatibility reasons in case the current algorithm exhibits weaknesses, and requires swift replacement, in the future.

10.2. Securing the Connection to the Authentication Service

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

If a user does not receive a certificate from the authentication service over this TLS connection that corresponds to the expected domain (especially when the user receives a challenge via a

mechanism such as Digest), then it is possible that a rogue server is attempting to pose as an authentication service for a domain that it does not control, possibly in an attempt to collect shared secrets for that domain. A similar practice could be used for telephone numbers, though the application of certificates for telephone numbers to TLS is left as a matter for future study.

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

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

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

10.3. Authorization and Transitional Strategies

Ultimately, the worth of an assurance provided by an Identity header is limited by the security practices of the authentication service that issues the assurance. Relying on an Identity header generated by a remote administrative domain assumes that the issuing domain uses recommended administrative practices to authenticate its users. However, it is possible that some authentication services will implement policies that effectively make users unaccountable (e.g., ones that accept unauthenticated registrations from arbitrary users). The value of an Identity header from such authentication services is questionable. While there is no magic way for a verifier to distinguish "good" from "bad" signers by inspecting a SIP request, it

is expected that further work in authorization practices could be built on top of this identity solution; without such an identity solution, many promising approaches to authorization policy are impossible. That much said, it is RECOMMENDED that authentication services based on proxy servers employ strong authentication practices.

One cannot expect the Identity and Identity-Info headers to be supported by every SIP entity overnight. This leaves the verifier in a compromising position; when it receives a request from a given SIP user, how can it know whether or not the sender's domain supports Identity? In the absence of ubiquitous support for identity, some transitional strategies are necessary.

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

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

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

Finally, it is worth noting that the presence or absence of the Identity headers cannot be the sole factor in making an authorization decision. Permissions might be granted to a message on the basis of the specific verified Identity or really on any other aspect of a SIP request. Authorization policies are outside the scope of this specification, but this specification advises any future authorization work not to assume that messages with valid Identity headers are always good.

10.4. Display-Names and Identity

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

11. IANA Considerations

[TBD: update for rfc4474bis or remove?]

This document requests changes to the header and response-code sub-registries of the SIP parameters IANA registry, and requests the creation of two new registries for parameters for the Identity-Info header.

11.1. Header Field Names

This document specifies three SIP headers: Identity, Identity-Reliance and Identity- Info. Their syntax is given in Section 7. These headers are defined by the following information, which has been 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: n
Header Name: Identity-Reliance
Compact Form:

11.2. 428 'Use Identity Header' Response Code

This document registers a SIP response code, which is described in Section 4.2. It is sent when a verifier receives a SIP request that lacks an Identity header in order to indicate that the request should be re-sent with an Identity header. This response code is defined by the following information, which has been 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

11.3. 436 'Bad Identity-Info' Response Code

This document registers a SIP response code, which is described in Section 4.2. It is used when the Identity-Info header contains a URI that cannot be dereferenced by the verifier (either the URI scheme is unsupported by the verifier, or the resource designated by the URI is otherwise unavailable). This response code is defined by the following information, which has been added to the method and response-code sub-registry under <http://www.iana.org/assignments/sip-parameters>

Response Code Number: 436
Default Reason Phrase: Bad Identity-Info

11.4. 437 'Unsupported Credential' Response Code

This document registers a SIP response code, which is described in Section 4.2. It is used when the verifier cannot validate the credential referenced by the URI of the Identity-Info header, because, for example, the credential is self-signed, or signed by an authority for whom the verifier does not trust. This response code is defined by the following information, which has been added to the method and response-code sub-registry under <http://www.iana.org/assignments/sip-parameters>

Response Code Number: 437
Default Reason Phrase: Unsupported Credential

11.5. 438 'Invalid Identity Header' Response Code

This document registers a SIP response code, which is described in Section 4.2. It is used when the verifier receives a message with an Identity signature that does not correspond to the digest-string calculated by the verifier. This response code is defined by the following information, which has been added to the method and response-code sub-registry under <http://www.iana.org/assignments/sip-parameters>

Response Code Number: 438
Default Reason Phrase: Invalid Identity Header

11.6. Identity-Info Parameters

The IANA has created a registry for Identity-Info headers. This registry is to be prepopulated with a single entry for a parameter called 'alg', which describes the algorithm used to create the signature that appears in the Identity header. Registry entries must contain the name of the parameter and the specification in which the parameter is defined. New parameters for the Identity-Info header may be defined only in Standards Track RFCs.

11.7. Identity-Info Algorithm Parameter Values

The IANA has created a registry for Identity-Info 'alg' parameter values. This registry is to be prepopulated with a single entry for a value called 'rsa-sha256', which describes the algorithm used to create the signature that appears in the Identity header. Registry entries must contain the name of the 'alg' parameter value and the

specification in which the value is described. New values for the 'alg' parameter may be defined only in Standards Track RFCs.

A previous version of this specification defined the 'rsa-sha1' value for this registry. That value is hereby deprecated, and should be removed. It is not believed that any implementations are making use of this value.

[TBD - consider EC for smaller credential sizes?]

12. Acknowledgments

Lots of people made significant contributions to this document.

13. Changes from RFC4474

Lots of people made significant contributions to this document.

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

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

Added the Identity-Reliance header

Deprecated 'rsa-sha1' in favor of new baseline signing algorithm

[TBD - more]

14. Informative 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] Rosenberg, J. and H. Schulzrinne, "Session Initiation Protocol (SIP): Locating SIP Servers", RFC 3263, June 2002.
- [3] Housley, R. and P. Hoffman, "Internet X.509 Public Key Infrastructure Operational Protocols: FTP and HTTP", RFC 2585, May 1999.
- [4] Peterson, J., "Session Initiation Protocol (SIP) Authenticated Identity Body (AIB) Format", RFC 3893, September 2004.

- [5] Schulzrinne, H., "The tel URI for Telephone Numbers", RFC 3966, December 2004.
- [6] Housley, R., Polk, W., Ford, W., and D. Solo, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 3280, April 2002.
- [7] Rescorla, E., "HTTP Over TLS", RFC 2818, May 2000.
- [8] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, May 2008.
- [9] 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.
- [10] Hoffman, P. and J. Schlyter, "The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA", RFC 6698, August 2012.
- [11] Josefsson, S., "Domain Name System Uniform Resource Identifiers", RFC 4501, May 2006.
- [12] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", RFC 4234, October 2005.
- [13] Housley, R., "Cryptographic Message Syntax (CMS) Algorithms", RFC 3370, August 2002.
- [14] Josefsson, S., "The Base16, Base32, and Base64 Data Encodings", RFC 3548, July 2003.
- [15] Peterson, J., "A Privacy Mechanism for the Session Initiation Protocol (SIP)", RFC 3323, November 2002.
- [16] Peterson, J., Schulzrinne, H., and H. Tschofenig, "Secure Telephone Identity Problem Statement", draft-ietf-stir-problem-statement-03 (work in progress), January 2014.
- [17] Peterson, J., "Retargeting and Security in SIP: A Framework and Requirements", draft-peterson-sipping-retarget-00 (work in progress), February 2005.

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

Authors' Addresses

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

Email: jon.peterson@neustar.biz

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

Email: fluffy@iii.ca

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

Phone: +1 650 678 2350
Email: ekr@rtfm.com

STIR BOF Group
Internet Draft
Intended status: Standards Track
Expires: January 30, 2013

H. Kaplan
July 15, 2013

A proposal for
Caller Identity in a DNS-based Entrusted Registry (CIDER)
draft-kaplan-stir-cider-00

Abstract

This document describes a proposal for providing a database service for authentication information for Caller-ID E.164 numbers, nationally-specific number codes, and email-style names used in communication requests (such as call setup, instant messages). The model proposed uses a DNS service as a Registry for cryptographic public-keys. The database service solution is called CIDER.

Status of this Memo

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

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 January 15, 2013.

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction.....	3
2. CIDER Overview.....	4
3. Terminology.....	6
3.1. New Terminology.....	6
4. Background Information.....	8
4.1. Benefits and Drawbacks to Using DNS.....	8
4.2. Relation to ITU and National Number Authorities.....	9
4.3. Open Numbering Plans.....	10
5. Caller-ID vs. DNS Delegation and Authority Models.....	11
5.1. Email-style Registries.....	12
5.2. E.164 and Number Code Registries.....	12
6. Assignee Roles and Actions.....	13
6.1. Key Agents and Third-Party Private Agents.....	14
7. Using CIDER for Caller-ID Verification.....	15
7.1. CIDER DNS Client Requirements.....	15
7.2. Information Needed by the Caller-ID Verifier.....	16
7.3. Generating the CIDER DNS query.....	16
7.3.1 Query for Email-style Identity	16
7.3.2 Query for E.164-based Identity	16
7.3.3 Query for Number Code-based Identity	17
7.4. Processing the DNS Answer.....	17
8. DNS Binding.....	18
8.1. Subdomain Namespace.....	18
8.2. Resource Record Type for Key Storage.....	18
8.3. TXT Record Format.....	18
9. Security Considerations.....	19
9.1. Privacy of Assignee.....	19
10. Open Issues.....	19
11. IANA Considerations.....	20
12. Acknowledgments.....	20
13. References.....	20
13.1. Normative References.....	20

13.2. Informative References.....	20
Author's Address.....	21
Appendix A. Possible Deployment Model.....	21
Appendix B. Requirements for a CAPP Mechanism.....	23

1. Introduction

For many years the identity of the calling party (i.e., Caller-ID) of voice communications has been made available to the callee, and has been assumed to be generally accurate/reliable. Not only do end users expect this to be the case, but also applications such as calling-name (CNAM) services, call-back services, and voice-mail account access, have depended on the validity of the Caller-ID. Even some forms of call rate-control and Denial-of-Service (DoS) attack prevention between service providers depend on valid Caller-IDs. The ability to spoof Caller-IDs enables numerous fraud and abuse scenarios.

Unfortunately, this problem already exists and is exacerbated by the presence of Internet-based calling services. While a small volume of Caller-ID spoofing has occurred for many years on the PSTN, it was infrequent enough to handle through manual investigation, and could be largely ignored as being merely isolated cases. Lately, however, the frequency has increased to a point that national regulators are becoming concerned (e.g., [fcc-doc]), the media have been reporting on it, and service providers themselves have been DoS attacked using invalid Caller-IDs.

There are several causes for the increase in Caller-ID spoofing: the decrease in cost for making calls, the large number of inexpensive products capable of generating spoofed Caller-IDs, and the growing number of entry points/paths into the trust network upon which Caller-ID reputability has always depended.

Spoofing a Caller-ID is possible because to-date there has been no means to validate a Caller-ID; instead the assumption has been that the received Caller-ID came from trustworthy upstream providers, in a chain-of-trust based on the PSTN model. Some PBXs are also allowed to generate whatever Caller-IDs they wish across PRI circuits, based on the belief that they can be trusted due to the relative cost, complexity, and physical hurdle of getting a PRI trunk.

The original assumption of Caller-ID reputability that was based on the PSTN trust model no longer holds. Therefore, in order to keep Caller-IDs valid in a less trustworthy interconnection model, a means of verifying Caller-IDs must be deployed that works with the

model. Replacing the current PSTN-style interconnection model itself is not realistic.

One approach for verifying Caller-IDs relies on public-key cryptography, whereby the originator signs some information in the call setup message using a private key, and the receiver verifies the signature using the public key. For example the solutions described in [RFC4474], [draft-4474bis], and [draft-ikes]. These approaches are conceptually similar to those employed by [DKIM] and [DMARC], which have been created to address a similar issue for email.

Regardless of the solution used, there needs to be a way for:

- (1) the originator to have a private/public key pair that is trusted by everyone else to prove the originator can claim the Caller-ID
- (2) any receiver of the originator's message to be able to retrieve the public key the originator used, and
- (3) the receiver to verify the private key was used to sign for the given Caller-ID

This document describes a model for achieving those needs. It does so by using the Domain Name System [DNS], as a database for mapping source identities to public keys with an authority structure. The DNS tree nodes have no direct identifying information - they do not identify the carrier or end-user that was assigned each E.164 number, for example. The general model and architecture is named "CIDER".

2. CIDER Overview

At a high-level, CIDER provides a database infrastructure using DNS for storing and retrieving public keys to authenticate source identities in communication messages, for example SIP or XMPP requests. Instead of being told by the message originator where the verifier should get a full certificate and then having the verifier check that the certificate is signed by a common trusted third-party for the specific identity being claimed, CIDER retrieves the public key directly from DNS and relies on the DNS authority model to control authorization of the public keys for source identities.

CIDER currently covers three types of identities: international E.164 numbers, nationally-specific number codes, and email-style names. Each type uses a different anchor to its domain name, and thus can be deployed in different ways. The DNS infrastructure used for each may be the public DNS infrastructure, or local private DNS instances populated with the same data. They can even be used in a restricted "federation" model, whereby only specific entities have access to the CIDER data: the public keys and other associated data.

For domain-based identities, CIDER follows the [DKIM] model of using each identity domain's DNS zone with a defined node name and key selector to hold the public key. The DKIM "key selector" is called a CIDER "key index" in this document, because it is syntactically different. The subdomain node name prefixed to the source's domain name is also different ("_cidkey" vs. "_domainkey"), and the TXT Resource Record format is more constrained than DKIM's.

For E.164 numbers and number codes, CIDER uses a reverse-dotted notation similar to ENUM for the DNS structure. The top of the domain tree hierarchy (the anchor) for the E.164 and number code entries is still TBD - it could be one single root anchor for all country-codes, or it could be a different anchor per country-code or geopolitical region or whatever.

In order to simplify discussion and explanation in this document, the domain 'cid.example.org' is used to describe a common top-level anchor domain for both E.164-based and number code-based identity entries. In practice they could be different, with E.164 using 'cid.example.org' and number codes using 'cid.example.net', to have separate authorities for E.164-based numbering vs. number codes.

The structure of CIDER's DNS hierarchy for the E.164-based and number-code-based domains follows a reverse-dotted notation of the E.164 numbering format, so that for example the +1 "country-code" would be the subdomain '.1.cid.example.org', whereas that for the +49 German country-code would be '.9.4.cid.example.org'. Nationally-specific number codes would be prefixed with their country-code, so that for example "911" in the North American Numbering Plan country-code 1 would be the domain "1.1.9.1.example.org".

The public keys in CIDER's DNS tree nodes have no direct identifying information - they do not identify the carrier or end-user that was assigned each E.164 number.

The public keys could be unique per E.164 number entry, or they could be the same public key for all E.164 entries belonging to the same end entity. This is purely an administrative choice of the owner. For example a carrier that is assigned millions of E.164 entries might re-use the same public-key in all of them. Or it can use one public key for every thousand E.164 numbers, or whatever. It's up to the individual end-entities that were assigned the E.164 numbers to decide what to populate their assignee DNS identity entry nodes with.

If an organization explicitly desires to make itself known, it can put its name into some to-be-defined DNS Resource Record for its

E.164 number identity entry node; such may be the case for corporate 1-800 numbers, for example. The organization can decide, on a number-by-number basis, whether to make itself known or not for that E.164 number in the CIDER DNS tree.

It should be noted that although the public key entries installed in the CIDER DNS tree are unique for every E.164 number (or group of numbers), they do not need to be maintained/stored by the organizations that uploaded them. Unlike credentials used with SSL/TLS, for example, the public keys are not transmitted by their servers to client hosts using certificates; rather, the keys are only retrieved by the verifiers when validating the Caller-ID signatures, and that's done through DNS. The signers themselves only need to know the private key, to which the public key is paired to for any given E.164 number. Furthermore, the originators can use the exactly the same private key for every E.164 number they sign for, by uploading the same public key into every E.164 node they are assigned in the CIDER DNS.

3. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119. The terminology in this document conforms to RFC 2828, "Internet Security Glossary".

It is assumed the reader is already generally familiar with E.164 numbers, Public-key cryptography concepts, Domain Name System (DNS), and DNS Security (DNSSEC).

This document uses the term "identity" instead of "identifier" with regards to verifying source identity. The reason for this is that it is not the syntactic encoding of an E.164 digit string, number code, or email-style name that are being verified - it's the canonical E.164 phone-number, number code, or email-style name that's being verified. In other words CIDER is used for verifying a logical entity, not a specific representation; and the logical entity is not a specific human user or SIP agent/device, but rather a phone-number or number code or email-style name.

3.1. New Terminology

Caller-ID: the identity of the originator of a communications request; for example the From header field in a SIP INVITE call setup request or MESSAGE instant message request.

E.164 number: a phone number in international E.164 format, which is understood by the originating and receiving entities to represent a globally unique number. This definition includes numbers that are not technically "E.164" numbers, such as toll-free 1-800 numbers in North America.

Number code: a nationally-specific number which is not representable as an E.164 number. Examples of number codes include two or three digit emergency service numbers, N11-type numbers in NANP, and inter-carrier common short codes.

Email-style name: a 'user@domain' format identifier, for which the user portion is scoped to the domain portion, and the domain portion is a classic, public domain name; removing or changing the domain portion would fundamentally change the identity of the user.

Source identity: the E.164 number, number code, or email-style name used for identifying the originator of a message to the receiving user; i.e., the identity used for "Caller-ID".

Identity entry: the DNS name entry representing an E.164-based number, nationally-specific number code, or email-style domain name.

CIDER Registry: an instance of a DNS hierarchy used for storage and retrieval of public keys for source identities. A CIDER Registry may be for an entire E.164-based country-code tree, or just for portions of one, or just for a single domain name.

CIDER Registrar: the organization that owns, manages, and is authoritative for a CIDER Registry.

CIDER Assignee: the organization/entity that the CIDER Registrar allows to populate specific identity entries with public key data, can grant/rescind Key Agents, and permanently transfer ("donate") the identity entry to another Assignee. This could be a carrier or service provider for E.164 numbers, for example.

Key Agent: an organization/entity that the CIDER Assignee grants access to upload public key data only, for one or more of the Assignee's identity entry nodes. This could be an Enterprise or service provider customer of a carrier, for example.

Third-Party Private Agent: an organization/entity that can upload public key data for an identity, but not directly to the Registry - only via a true CIDER Assignee, and thus the Registrar knows nothing about the Third-Party Private Agent. This could be an Enterprise or end-user, for example.

4. Background Information

4.1. Benefits and Drawbacks to Using DNS

A database model to hold public keys used for caller-id verification could be accessed using a protocol other than DNS. Examples include HTTP, LDAP, or even DIAMETER. CIDER uses DNS for the following reasons:

- o The DNS architecture has demonstrated massive intrinsic scalability, by allowing branches of the database tree to be run on separate physical servers and separate, independent administration.
- o The DNS protocol is extremely efficient, with no extra round-trip delays creating connections or resolving hostnames.
- o The DNS protocol allows tight control over retransmission timers and timeout behavior from the application layer, because it runs on UDP.
- o The DNS protocol has well-established practice for highly effective geographic redundancy techniques, such as through anycasting because it runs over UDP.
- o The DNS protocol has well-established and effective caching in local servers, and techniques are available to have local copies of a DNS tree.
- o The DNS protocol and architecture provide a seamless, global service, but have well-established and highly effective practice for delegation of authority, which is useful for country-code level separation of authority for E.164 numbers and nationally-specific number codes.
- o The DNS protocol already defines the encoding syntax and semantics for many of the functions needed.
- o DNS is already used very widely by services employing DKIM for email signing/verification for a similar purpose as would be needed for email-style domain-based caller-id identities.
- o Some service providers already use DNS for Private ENUM for CNAM, number portability, and communication request routing purposes.
- o Virtually all clients/hosts have DNS querying capabilities today; and there are many DNS server vendors, including a widely popular open-source implementation (BIND).

There are some drawbacks to using DNS, however:

- o DNS typically uses UDP, so if the query or response are larger than the MTU size between the client and server, fragmentation will occur. The base DNS maximum message size is 512 bytes, but CIDER requires [EDNS0] be used, which allows larger message sizes; so the real issue is for message sizes over ~1460 bytes. The current CIDER entries would not typically be that large, even if DNSSEC is being used.

- o Deploying a private registry using DNS is more complicated because it runs over UDP and has no defined mechanism to enforce access control on the queries. Private and federated DNS has been deployed, but it usually requires using private IP Addresses and VPN-style access to the servers.
- o The DNS query model does not support providing a different resource record(s) for different querying agents. In other words, it does not give a different answer depending on who asks the question. There are widely-used work-around behaviors for this today, but they are not standardized.
- o No changes to the underlying DNS protocol are envisioned. Should they be needed, some members of the IETF treat the DNS protocol, architecture, and its instantiation in the public Internet for domain name resolution, all as one monolithic, inter-dependent usage. Therefore, any changes required of the protocol have to also work for the Internet domain name resolution infrastructure as rooted-in/controlled-by IANA/ICANN. Even if a DNS extension were created for purely private use, the IAB has essentially stated they fear the public domain name infrastructure is so brittle that it would collapse if the extensions were accidentally sent to a public DNS server.

[Note: If the STIR Working Group decides future extensions to the DNS protocol would be needed, there is a way to achieve this: we could copy the DNS RFCs into new documents, give the protocol a new name ("NDNS" for "Not DNS"?) and a new default UDP port number other than 53. Such a concept seems silly, but it would give us the same benefits without the drawback of having to worry about "The DNS" collapsing due to a few unexpected bytes in a query packet.]

4.2. Relation to ITU and National Number Authorities

A concern has been raised that using E.164 numbers in a DNS hierarchy would be problematic because the ITU should be responsible for delegating E.164 country-codes, and national authorities should be responsible for managing their country-code numbers. The fear is that deploying CIDER without the ITU and national authorities approving and managing it would lead to claims of inappropriate subverting of the E.164 numbering system; or fears that for CIDER to work correctly would require such approval and management.

This document section explains why such concerns are either unfounded, or apply equally to any database model used for caller-id verification of E.164 numbers, whether it be DNS or otherwise.

With regards to approval, there is already precedent for the IETF to use names defined by other organizations in DNS: the top-level

country domain names are based on a list defined by ISO, for example.

With regards to subverting the E.164 numbering system, this document makes no claim that a specific CIDER registry, or top domain root for it, is in fact authoritative for the E.164 number space. In fact, although this document uses the term "E.164" to describe the tree structure and assignment model, all that is truly described is how a DNS domain may create subdomain names with Resource Records that could be used by others to retrieve public keys. It is up to the users of the registry/domain to decide whether to believe it represents E.164 numbers, and to use it for identities in call-control scenarios. The same CIDER model could be used, for example, to deploy a public key storage/retrieval mechanism for a private phone-numbering plan; or even digit-based names that have nothing to do with phone numbers, such as Autonomous System numbers or Enterprise OIDs.

The CIDER service described in this document does not dictate who will be registrars, although existing authorities and operators might be reasonable candidates. All CIDER needs, however, is for some organization to manage the domain tree for a given E.164-based namespace, to decide what entities get to populate the public key entries for its branch nodes, and for verifiers to use that organization's domain tree for retrieving the public keys and trust them to be correct. In other words, any organization can be the Registrar and create its own CIDER DNS registry with its own domain as the anchor of the tree, and so long as both signers and verifiers use that organization's registry and it is accurate, then CIDER works.

This is no different than what occurs for the web-pki model of third-party Certificate Authorities (CAs). If you trust a CA, then you trust that the certificates it signs are for the domain/host names contained in the certificate, even though CAs are not authoritative for DNS domain name assignments. The IETF has a mechanism for tying web-pki certificates to the actual authority of DNS names: DANE is that mechanism; but without DANE verification, the web-pki model is purely based on trusting the CAs to be accurate with regards to domain/host name assignments. Therefore, any caller-id verification model that is not ultimately controlled by the numbering authorities for the E.164 number space would have the same issues as CIDER.

4.3. Open Numbering Plans

The E.164 number model is technically an open numbering plan, meaning it does not specify a fixed number of digits. In some countries, the national numbering plan has a fixed number (e.g.,

North America does), but in others it is left open (e.g., Germany). For CIDER, this has implications if the source identity could be more digits than the CIDER Registrar knows about and has granted upload access to a CIDER Assignee for.

For example, in Germany not only is the numbering plan open, but the number assignment model is as well: an Enterprise is given a single phone number, and the Enterprise can then add a variable number of digits at the end for their specific lines/users. For routing purposes, the carriers only need to route on the assigned number portion, and the Enterprise's PBX can route the final leg to the user based on the whole number.

If the Enterprise can also claim the longer number sequence as a caller-id, but the CIDER Registrar only has entries for the assigned portion, then verification will fail.

This document does not specify a solution to this problem, but leaves it as an open issue to be decided upon. There are at least two potential solutions we can choose from:

- o CIDER could specify that Assignees can upload information to create subdomains within their assigned E.164 identity entry node. For example, let the Enterprise notify the carrier of the extra digits and their public keys, and the carrier in turn uploads it to the Registry.
- o We could require the SIP/XMPP/SS7 message information include the number of significant digits to use, so that CIDER is only queried for the assigned number portion; and have the TXT RR for the assigned number indicate that it's an open number.

5. Caller-ID vs. DNS Delegation and Authority Models

The concept of delegation and authority is somewhat confusing when referring to CIDER, because the terms are used in two different ways: one is the delegation and authority model of DNS subdomains/zones, and the other is delegation and authority model for caller-id identities and their public keys.

CIDER makes a clear distinction between which entities are allowed to provision public keys into specific entries in the CIDER Registry, versus which entities own, control, administer and are authoritative for the DNS domains/zones in the hierarchy of the Registry.

In other words, the CIDER Registry is split into two distinct aspects: the entities that own the Registry and thus the DNS domains/zones used to access those public keys, and the entities that are assigned rights to upload public key information for their

assigned identity entries in the DNS-based Registry. The former are called "CIDER Registrars", and the latter are "CIDER Assignees".

5.1. Email-style Registries

For email-style domain-based identities, CIDER uses the DKIM model of storing and retrieving the public keys from the identity domain's DNS zone. For example, if the source identity is "alice@example.com", then the DNS zone "example.com" is used, and thus uses the supplied domain name, to follow the typical DNS delegation and authority model for its contents. Each domain is thus its own CIDER Registrar as well as its own CIDER Assignee, and there is no distinction between the delegation and authority model for DNS versus the caller-id identities and public keys.

If a domain owner wishes to allow another entity to use its domain name for caller-id verification, however, the owner can follow the same model as that defined in the next section for E.164 and number code registries. For example, if "example.com" wishes to allow a contracted third-party company to generate calls using "example.com", it can continue to be the CIDER Registrar for "example.com" and make the other company a CIDER Assignee for it as well.

5.2. E.164 and Number Code Registries

For E.164 and number codes, the details for delegation and authority are separated. There is a CIDER Registrar which is the DNS authority for domains/zones/nodes, and the Registrar allows CIDER Assignees to upload public keys into specific identity entry nodes representing the E.164/number-codes.

While it is tempting to consider using the DNS subdomain delegation model for delegating DNS zones for specific E.164-based ranges or specific numbered entries to the carriers or end users to which the numbers are assigned. CIDER does not depend on such a DNS delegation model, and in fact it is expected such a DNS delegation model will not be used in practice; this is due to both the need to maintain privacy of who the carrier or end-user is for each number, and because the number portability behavior in some national numbering plans would make such a model untenable.

For example, from a DNS and DNSSEC perspective, the DNS zone authority for each of the subdomains within 'cid.example.org' (the country-code CIDER Registrars) could be the national numbering plan administrator for each country-code, possibly determined by the ITU to be authoritative for the anchor 'cid.example.org'.

Below the country-code level domain, each nation and national number authority can decide how they choose to delegate DNS zone authority, or not to. There is no requirement to delegate out DNS zone authority below the national country-code level whatsoever, nor any prohibition from doing so; the country-code authority can be the DNS authority for the entire E.164 number space of DNS domains/nodes under their country-code level. [Note: for North America, the DNS delegation could be separated by area-codes; to give Canada's area-codes a different authority from those in the USA, for example]

In fact, having a single DNS authority might have some advantages: it prevents people from learning how many numbers each carrier has, and it reduces the time for caller-id verification because the DNSSEC authority signing chain is shorter. Since most calls are national calls, the verifying systems can use a local cache of the national numbering administrator's certificate to verify the certificate of most E.164 caller-ids.

The important distinction is that it does not matter who manages the CIDER DNS registry for a given country-code - what's important is that the CIDER Registrar allows the entities which are assigned the E.164 numbers (the CIDER Assignees) to upload their public keys into their respective E.164-based nodes.

For example the CIDER Registrar can allow a SIP service provider to be the Assignee for a set of the Registry's E.164 entries, so that the service provider can upload the public keys it wishes to use for its caller-ids. From a DNS perspective, however, the Registrar is still the singular authority of the DNS zones/nodes for those E.164-based nodes. The domain tree and its zones/nodes "belong" to the Registrar from a DNS perspective.

6. Assignee Roles and Actions

As explained previously, CIDER creates a distinction between the DNS authority (the CIDER Registrar) vs. an entity allowed to upload public keys (the CIDER Assignee).

For every identity entry in the DNS tree (i.e., leaf node), the Registry has one and only one CIDER Assignee. The Assignee may be the Registrar itself, as would usually be the case for email-style domain-based identity CIDER Registries for example. For E.164-based identities, the Assignee probably is the service provider/carrier assigned the number by the national numbering authority for the given country-code.

The Assignee has controlled access to the Registry for performing the actions it can take. This may be through a web portal, or even via email or fax; if the STIR Working Group decides to continue with

CIDER, however, the author strongly recommends that a specific protocol be defined for this purpose in another document: a CIDER Assignee Publishing Protocol (CAPP). For example, CAPP could be either an HTTP/SOAP/XML or HTTP/REST type protocol. CAPP would be useful for DKIM and DNS in general, and as an alternative for Dynamic DNS [DDNS]. Requirements for CAPP are given in Appendix B.

An Assignee can add, modify, or remove public key entries from its identity node(s) in the Registry, and thereby affect the available TXT RR entries. Depending on how open numbering assignment is handled (currently an open issue in this document), the Assignee might be able to add, modify, or remove subdomain/additional-digit identities below the one the Registrar granted it access to, if the Registrar allows it.

If the Registrar allows it, an Assignee can permanently transfer control of its identity node to another Assignee, which is called "donating" its identity in this document. Such would be the case for number porting in certain countries, for example.

An Assignee can also grant/rescind access to Key Agents for its identity entries(s), if the Registrar supports such a model, as described in the next section.

6.1. Key Agents and Third-Party Private Agents

One of the use-cases a Caller-ID verification mechanism needs to support is that of enabling a third-party to assert someone else's Caller-ID for outbound calls/communications. An example of this need is outsourced call-centers making calls on behalf of a company, or even a doctor using her personal mobile phone to make calls with her medical office's number as her Caller-ID.

CIDER supports this in two ways: by either having the CIDER Registrar support an Assignee and one or more designated Key Agents for the same identity entry, or by having the Registrar only know about the CIDER Assignee and having the third-parties upload their public keys through the Assignee as Third-Party Private Agents.

The difference between a Key Agents and Third-Party Private Agents is one of involvement with the Registrar and Registry. If a Registrar supports Key Agents, then the Assignee has to grant this role, and the Registrar has to know about them, have credentials for them, etc. With Third-Party Private Agents, however, the Registrar knows nothing about it - the third-party uploads public keys to the Assignee directly (e.g., using CAPP), which then uploads them to the Registry in the same manner as its own public keys (again using CAPP).

From a Caller-ID verifier's perspective - from the data it can view in the retrieved Resource Records - there is no difference between Assignee, Key Agents or Third-Party Private Agents. In fact the verifier can't even discern who the Assignee is.

7. Using CIDER for Caller-ID Verification

CIDER specifies a key retrieval mechanism. The mechanics of Caller-ID verification that use the key is left for definition by a separate document. One example of such a mechanism is defined in [draft-ikes]. This section defines the information a verifier CIDER client needs in order to retrieve the appropriate public key for a verification mechanism, and how the retrieval is performed.

7.1. CIDER DNS Client Requirements

A Caller-ID verifier acting as a CIDER DNS client MUST implement DNS over UDP using [EDNS0] in order to handle message sizes larger than 512 bytes. The client MUST be prepared to receive DNSSEC responses, unknown RRs, etc.

If the verifier supports email-style identities, it MUST be able to query DNS servers which resolve public Internet DNS names.

If the verifier supports E.164-base identities, it is RECOMMENDED that the client be configurable to use different DNS server(s) for this purpose, separate from the one(s) used for other identity types. The client SHOULD also support being configurable for using a different anchor domain name for this purpose, separately from the one used for number code identities.

If the verifier supports number code-based identities, it is RECOMMENDED that the client be configurable to use a different DNS server(s) for this purpose, separate from the one(s) used for other identity types. The client SHOULD also support being configurable for using a different anchor domain name for this purpose, separately from the one used for E.164-based identities.

The client MAY be a recursive resolver, but it is strongly RECOMMENDED that it be a stub resolver and allow the DNS servers to resolve on its behalf. Otherwise, it will be difficult to use such a client in access-restricted CIDER deployments, such as private or federated ones. For example if the CIDER Registry is a private one for one or more nations. (see Appendix A)

7.2. Information Needed by the Caller-ID Verifier

For CIDER to function properly, a Caller-ID verifier needs to know the following information:

- o The source identity value
- o The source identity type: domain-based, E.164-based, or number code-based
- o The public key index value

The public key index value identifies which of several possible public keys for a given source identity should be used for verifying the message.

The source identity value need not be the whole source identity as a URI or 'user@domain' string - it only needs to be the information used for the CIDER query as defined in the next section.

7.3. Generating the CIDER DNS query

In order to generate a CIDER DNS query, the CIDER verifier performs slightly different actions depending on the source identity type, as defined in these sections.

7.3.1 Query for Email-style Identity

For an email-style source identity, the verifier CIDER client takes the 'user@domain' string and ignores the "user@" portion. The remaining domain name becomes the base of the DNS query key. The verifier prepends the CIDER public key index value as a subdomain, followed by the subdomain "_cidkey", in front of the domain name.

For example, if the source identity being verified is "alice@example.com" with a public key index value of 3, the DNS query key becomes "3._cidkey.example.com".

The verifier then issues a DNS query with the above query key to the public Internet DNS.

7.3.2 Query for E.164-based Identity

For an E.164-based source identity, the verifier CIDER client takes the international E.164 digit string, removes any leading '+' or visual separators, puts dots (".") between each digit, and reverses the order of digits. The verifier then appends the E.164-based CIDER Registrar domain name to the end, and prepends the CIDER public key index value as a subdomain, followed by the subdomain "_cidkey".

For example, if the source identity being verified is "+16035551010" with a public key index value of 2, and a CIDER Registrar domain for the country-code 1 of "1.cid.example.org", then the DNS query key becomes "2._cidkey.0.1.0.1.5.5.5.3.0.6.1.cid.example.org".

The verifier then issues a DNS query with the above query key to either the public Internet DNS, or to the DNS server provisioned for such a purpose.

7.3.3 Query for Number Code-based Identity

For a nationally-specific number code-based source identity, the verifier CIDER client takes the number code digit string, prepends the country-code the number code is nationally-specific for, puts dots (".") between each digit, and reverses the order of digits. The verifier then appends the Number-code CIDER Registrar domain name to the end, and prepends the CIDER public key index value as a subdomain, followed by the subdomain "_cidkey".

For example, if the source identity being verified is "911" for the country-code 1, with a public key index value of 2, and a Number-code CIDER Registrar domain of "cid.example.org", then the DNS query key becomes "2._cidkey.1.1.9.1.cid.example.org".

The verifier then issues a DNS query with the above query key to either the public Internet DNS, or to the DNS server provisioned for such a purpose.

7.4. Processing the DNS Answer

Based on the DNS binding format defined in Section 8, a successful CIDER DNS query will produce a DNS answer with a TXT RR as defined in Section 8.3. The verifier needs to parse the TXT RR, to verify the version is "CIDER1", the key-type is "rsa" or some token the verifier understands, and to base64 decode the public key data.

If the version is not "CIDER1", or the key-type is not one the verifier understands, or the public key cannot be decoded or is not of a key size the verifier supports, then the verifier should treat it as a failure. If the public key is empty, the verifier should treat it as a failure. If the DNS answer is 'not found', the verifier should treat it as a failure. If the DNS query times out, the client should try any alternate servers it is provisioned for; if there are no more, it should treat it as a failure.

The action to take for a CIDER query failure is dependent on local policy and not defined in this document.

8. DNS Binding

A binding using DNS TXT records as a key service is hereby defined. All implementations MUST support this binding.

8.1. Subdomain Namespace

All CIDER keys are stored in a subdomain named "_cidkey", with a node name of the key index value. Given an email-style source identity of "alice@example.com" and a key index of "3", the DNS query will be for "3._cidkey.example.com". Given an E.164 source identity of "16035551010" and a key index of "1", the DNS query will be for "1._cideky.0.1.0.1.5.5.5.3.0.6.1.cid.example.org". Given a nationally-specific number code for "911" in the North-American Numbering Plan region (country-code 1), and a key index of "6", the DNS query will be for "6._cidkey.1.1.9.1.cid.example.org".

8.2. Resource Record Type for Key Storage

The DNS Resource Record type used in this specification is a TXT Resource Record (RR). A later extension of this standard may define another RR type.

Strings in a TXT RR MUST be concatenated together before use with no intervening whitespace. TXT RRs MUST be unique for a particular key index value; that is, if there are multiple records in an RRset, the results are undefined.

8.3. TXT Record Format

The TXT RR used for the CIDER key MUST follow the format specified in this section, in order to provide future extension capability. No whitespace is allowed in this format, and all values are case-sensitive. The format is based on the following ABNF:

```
txt-record      = version-param SEMI key-param [ SEMI txt-param ]
version-param   = "v=" "CIDER1"
key-param       = key-type SEMI key-data
key-type        = "k=" ("rsa" / hyphenated-word)
key-data        = "p=" DQUOTE [ base64 ] DQUOTE
```

The version identifies the TXT record content syntax and semantics, which for this document is defined as "CIDER1".

The key-type identifies the CIDER public key's (the "p=" value) type, which for this specification uses the "rsa" key type to indicate that an ASN.1 DER-encoded [ITU.X660.1997] RSAPublicKey [RFC3447] is being used in the key-data's value. Future

specifications may define other key types by assigning this field a different value, but still using the "CIDER1" version.

The key-data identifies the CIDER public key value, in base64 encoded form. An empty value (the literal string 'p=")'), means the public key for this index is not usable or has been explicitly revoked as opposed to simply removed. This might be useful if the CIDER DNS authority wishes to prevent use of a specific key index node entry for some time period of time by having an essentially empty TXT RR, as opposed to deleting the entry and re-using it when the next public key is uploaded by the assigned party.

9. Security Considerations

The same security considerations as described in [DKIM] apply to CIDER; in particular Sections 8.2, 8.3, 8.4, 8.6, and 8.7 of [DKIM].

9.1. Privacy of Assignee

For E.164-based CIDER Registries, privacy of assignee is a concern. The concern stems from the need to keep the assignee of an E.164 number unknown in general - to prevent simple scripts from being used to walk the CIDER DNS tree and learn what E.164 numbers are assigned to which carrier, for example. An example of why this needs to remain private is that using such knowledge one could learn how many subscribers a publicly-traded mobile provider has gained or lost in a quarter, and be able to buy or sell stock based on such knowledge in advance of the provider's quarterly-reported statements.

Such information could be easily learned if only one or a few public keys are used by a carrier for all of its numbers in the CIDER Registry. To defend against such abuse, it is strongly RECOMMENDED that assignees only re-use the same public key for a limited number of CIDER entries. For example a large assignee might use the same public key for a thousand or ten thousand of its E.164 numbers.

It should be noted that this security concern is not specific to using DNS - any open-access database protocol would be vulnerable to a script querying all entries. Controlled-access databases would be of less concern, but CIDER can also be used in a controlled-access model.

10. Open Issues

- How to handle open numbering plan assignment country-codes.

11. IANA Considerations

This document makes no request of IANA yet.

12. Acknowledgments

The general concept of using DNS in an ENUM model for caller-id verification has been discussed in the IETF for many years. Thanks to Dave Crocker for pointing out the DKIM similarities and usage, and for reviewing the draft in detail.

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).

13. References

13.1. Normative References

- [EDNS0] Vixie, P., "Extension Mechanisms for DNS (EDNS0)", RFC 2671, August 1999.
- [DDNS] Wellington, B., "Secure Domain Name System (DNS) Dynamic Update", RFC 3007, November 2000.
- [ITU.X660.1997] "Information Technology - ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)", ITU-T Recommendation X.660, 1997.
- [RFC3447] Jonsson, J., and Kaliski, B., "Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1", RFC 3447, February 2003.

13.2. Informative References

- [fcc-doc] http://hraunfoss.fcc.gov/edocs_public/attachmatch/DA-11-1089A1.doc
- [draft-ikes] Kaplan, H., "An Identity Key-based and Effective Signature for Origin-Unknown Types", draft-kaplan-stir-ikes-out-00, July 2013.
- [RFC4474] Peterson, J., and Jennings, C., "Enhancements for Authenticated Identity Management in the Session Initiation Protocol (SIP)", RFC 4474, August 2006.
- [draft-4474bis] Peterson, J., Jennings, C., and Rescorla, E., "Authenticated Identity Management in the Session Initiation

[DKIM] Allman, E., et al, "DomainKeys Identified Mail (DKIM) Signatures", RFC 4871, May 2007.

Author's Address

Hadriel Kaplan
Email: hadrielk@yahoo.com

Appendix A. Possible Deployment Model

In order to understand how CIDER could work in the real world, this section provides an example of how CIDER could be deployed in the North American Numbering Plan (NANP). This was chosen for the example because the NANP is one of the most complicated numbering models in the World: it has 24 member countries and territories, full number portability for both fixed and mobile line numbers in certain countries, separate numbering authorities (e.g., CRTC/CNA and NANC/NANPA), and multi-tiered numbering assignment/delegation behavior.

Today, the entire +1 NANP is administered by the North American Numbering Plan Administrator (NANPA), but certain functions are handled by separate organizations: the Canadian Number Administrator (CNA) handles certain functions for Canadian area-codes, for example, and the USA and Canada each have a separate administrator for their respective number portability databases. Within each country, numbers are officially assigned in blocks to legally-authorized carriers, who then (unofficially) assign them to their customers: non-carrier service providers, enterprises, and consumers. For the countries in NANP that support number portability, the original carrier "donates" the ported number to another carrier, and the number portability database is updated with a mapping of the ported number to an assigned switch number: the Location Routing Number (LRN). Numbers are not portable across countries within the NANP, and are not portable in certain cases even within a country.

One way CIDER could be deployed in the NANP is by having NANPA be the CIDER Registrar for the country-code 1 top domain, and delegate DNS domains for country-specific area-codes to their respective country administrators. For example, NANPA could be the CIDER Registrar for '.1.cid.example.org', but then delegate authority of the DNS subdomains '.4.0.2.1.cid.example.org',

'6.3.2.1.cid.example.org', etc., (representing the Canadian area codes 204, 236, etc.) to the CRTC/CNA if they wish it. Thus the CRTC/CNA would be the CIDER Registrar for those area-code branches of the number tree.

An alternative would be to have a private organization be the CIDER Registrar for the +1 country-code using its own domain for the Registry, such as '.1.example.org'. This would only be useful if carriers/service-providers agreed to use this Registry, of course, but this might make sense as a way to get the effort started and eventually transition it over to NANPA.

One specific model for who the Registrar could be would be to use the same administrator as that used for number portability. Currently the US-based FCC selects a company to run the NPAC (Number Portability Administration Center) for the US, and the CRTC selects one for Canada; they could contract the CIDER Registrar role to the same companies they contract number portability administration to. This has some obvious benefits, but also some drawbacks with regard to federal regulatory involvement and monopoly-type power/position for the contracted vendor(s).

Regardless of whom the Registrar is and what the Registry top domain name is, the officially assigned carriers for numbers would be designated as the Assignees of their respective number identity nodes. They can upload public keys for those number nodes, in order to perform the role of caller-id signers. When they assign numbers to their customers, they could either add them as Key Agents in the Registry, or handle them as Private Agents, as described in Section 6.1. In practice, it's likely they would only add their customers as Key Agents if the "customers" were service providers, but otherwise handle customers as Third-Party Private Agents.

When a number is ported from one carrier to another, the Assignee carrier would transfer assignee control by "donating" their identity to the other carrier, as described in Section 6. This would need to occur at the same general time as porting the number in the number portability databases, and would need to take effect within 15 minutes.

Having a defined protocol between the Assignee and the Registry, for publishing the keys into identity nodes, would help this process greatly. This would be implemented in the same carrier back-office systems that are used today for number porting, for example.

From a DNS query access perspective, it is likely that the CIDER Registry for NANP begin as a private database model. Only authorized entities would be able to query the CIDER Registry,

either using IPSEC/VPN-style access, or by having each carrier have a local read-only copy of the CIDER Registry.

Most carriers would likely have such local copies of the Registry, in servers they deploy within their core call control network, to reduce verification time and control availability/reliability. All 500 million current NANP numbers instantiated in a CIDER Registry would fit in ~500GB, which even laptops have sufficient storage for. It would only take two or three modern high-end servers to fit it all in *RAM*, let alone hard-drive/flash.

For carriers to have local copies of the Registry, they could use [AXFR], [IXFR], and [DNS-NOTIFY]; or a more-efficient protocol could be defined for this purpose. Modern DNS servers offer more than just the legacy DNS-based zone transfer/update protocols, and the newer protocols could be investigated for re-use for CIDER local copying.

If each national CIDER Registry is not publicly accessible for DNS querying on the public Internet, this causes some issues for international call scenarios. The goal of CIDER is to enable caller-id verification even for international calls/communications. This is still achievable, however, because there aren't that many country-codes and national numbering plans - there are approximately ~160 such in the World.

Therefore, it is reasonable for each national CIDER Registry to have a private, controlled connection to every other national Registry, creating a full mesh of connections. The private connections could be used to either provide server resolution of private DNS queries on behalf of the local nation's carriers' verification clients, or for the local nation's Registrar to itself have a local copy of the CIDER Registry of other nations, using the same protocol mechanics as the carriers use for their local Registry copies. Even 10 Billion number entries in a CIDER Registry read-only database would only consume ~10 Terabytes of storage, which is achievable for reasonable cost today. In practice, each national Registry would likely use a hybrid model: performing DNS queries to nations that are infrequent callers, while having local copies of Registries of frequent calling nations.

Appendix B. Requirements for a CAPP Mechanism

In order for CIDER to be usable in large scale across many carriers, there needs to be a defined protocol for how CIDER Assignees perform their allowed actions to the Registry. This document describes such a protocol as CIDER Assignee Publishing Protocol (CAPP), and this section gives the requirements for such a protocol, as input to

another document to specify the CAPP protocol itself. Some of the requirements are based on the actions described in Section 6.

Requirements for CAPP:

- o It must support the add, modify, and delete actions for CIDER identity node data.
- o It may only support handling the data in an opaque/blob manner. For example CAPP may only support uploading a TXT RR text string, instead of explicitly handling the public key value, version, and key type as discrete elements. This way it's not specific to CIDER usage.
- o It should support the add, modify, or delete actions for other DNS Resource Record types, or at least be extensible to do so in the future. This would allow non-CIDER usage, as well as allow extending CIDER for other number mapping use-cases: such as CNAM, number portability, or request routing purposes.
- o When adding new public key data (or a new TXT RR), it should be possible for the Assignee not to specify the key index (i.e., subnode) value for the new record, and instead for the server to return the new value it allocates. This is useful for Third-Party Private Agent model, where the Third-Party may not know the next available index value to use.
- o It should support a means of adding new data and returning the new key index value in separate transactions - or at least in some manner that allows for multiple minutes of time to pass between the addition of data and returning of new index value.
- o When adding new data, it must allow the Assignee to define an expiration time for the data. This is useful for the Third-Party Agent model described in Section 6.1, for example.
- o It must allow the Assignee to permanently transfer the Assignee role to another party, called "donate" in this document.
- o It should allow the Assignee to grant/rescind another entity the role of Key Agent for a given identity entry node, permanently or with an expiration time.
- o It must support an action/transaction model that allows for database atomicity, consistency, isolation, and durability (ACID).
- o It should provide a means for the Assignee to add or modify multiple identity node entries using one TXT RR (or one public key) in one transaction. This is a useful optimization for carriers that have millions of numbers, and re-use public keys across them. To support ACID more easily, however, this might be done using an indirection model.
- o It must specify distinct failure and error results, in a machine-consumable fashion.
- o It must support a means of logging each transaction (action/result) on both the client and server side such that both sides can easily reference the same event; for example by

encoding transaction identifiers and timestamps in the CAPP protocol messages.

- o It must support a means for the Registry server to authenticate a client Assignee; for example using credentials of a username/password digest-challenge model.
- o It must support a means for the client Assignee to authenticate the Registry server; for example by using TLS server-side certificates.
- o It must support a means of preventing eavesdropping and repudiation, for example by using TLS.

TIR BOF Group
Internet Draft
Intended status: Standards Track
Expires: January 30, 2014

H. Kaplan
Oracle
July 12, 2013

An Identity Key-based and Effective Signature
for Origin-Unknown Types
draft-kaplan-stir-ikes-out-00

Status of this Memo

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

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 January 12, 2013.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in

Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Abstract

This document describes a mechanism and format for signing source identity information of communication requests, in a manner capable of crossing multiple communication protocol types - even if the origin's protocol type is unknown. This is useful for providing E.164 and other forms of Caller-ID reputability for various communication protocols, such as SIP, XMPP, WebRTC, H.323, and SS7/ISUP.

Table of Contents

1. Terminology.....	2
2. Introduction.....	3
3. Overview of Operations.....	4
4. Background.....	5
4.1. Identity Types: E.164 vs. Number Codes vs. Email-style...	8
4.2. Determining Canonical E.164 Numbers and Number Codes.....	9
4.3. Determining Canonical Email-style Names.....	10
4.4. Call-forwarding Issues.....	11
5. IKES Generator Behavior.....	12
6. IKES Verifier Behavior.....	14
7. IKES Information Field.....	17
8. Usage in SIP.....	19
9. Usage in XMPP.....	21
10. Usage in SS7/ISUP.....	22
10.1. SIP-SS7 Interworking.....	24
11. Usage in H.323 and ISDN.....	24
12. Open Issues.....	25
13. Security Considerations.....	25
14. IANA Considerations.....	26
15. Acknowledgments.....	27
16. References.....	27
16.1. Normative References.....	27
16.2. Informative References.....	27
Author's Address.....	28

1. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119. The terminology in this document conforms to RFC 2828, "Internet Security Glossary".

Caller-ID: the identity of the originator of a communications request, such as a SIP INVITE call setup request or MESSAGE instant message request. Technically the Caller-ID is only what is displayed to receiving users, and does not necessarily have to be the source of the call. For the purposes of this document this distinction is not important.

E.164 number: a phone number in international E.164 format, which is understood by the originating and receiving entities to represent a globally unique number, regardless of any syntactic encoding for a domain portion. Changing the domain portion would not fundamentally change the identity of the resource.

Number code: a nationally-specific number which is not representable as an E.164 number. Examples of number codes include two or three digit emergency service numbers, N11-type numbers in NANP, and inter-carrier common short codes.

Email-style name: a 'user@domain' format identity, for which the user portion is scoped to the domain portion; removing or changing the domain portion would fundamentally change the identity of the user.

Source identity: the E.164 number, number code, or email-style name used for identifying the originator of a message to the receiving user; i.e., the identity used for "Caller-ID".

Destination identity: the E.164 number, number code, or email-style name of the original destination of a message; i.e., the original called party.

NANP: North American Numbering Plan. This document is not specific to North America, but the term "NANP" is used in some cases.

It is assumed the reader is generally familiar with E.164 numbers, asymmetric key cryptography concepts, and the SIP-Identity mechanism defined in [RFC4474].

2. Introduction

In order to provide source identity assurance (i.e., authentic Caller-ID), the SIP-Identity mechanism defined in [RFC4474] cryptographically signs certain SIP request header fields and the message body, stores the resulting signature in a SIP 'Identity' header, and defines the process by which a receiving node can verify the signature and thereby validate the source identity.

Unfortunately, the [RFC4474] signature is not usable if the request is forwarded/routed by many common types of SIP Back-to-back User-Agents (B2BUAs), nor if the request crosses from SIP to another protocol such as XMPP, H.323 or SS7/ISUP. Even in pure SIP scenarios, B2BUAs are likely to be in the request path, making [RFC4474] unusable in practice.

Furthermore, [RFC4474] is not sufficient for use in validating source identities that are treated as E.164 numbers. Unlike 'user@domain' email-style identities, an E.164 number is a national or global identity and not scoped within a domain. Authenticating the domain that sent the identity is not sufficient to determine if that domain can actually represent the E.164 identity. Without such proof, any domain could claim any E.164 number.

This document proposes a new signature-based mechanism that is usable in more complex scenarios, including those crossing B2BUAs and disparate protocol types. This document does *not* define how the public keys or Certificates used for validation are stored nor retrieved, nor how the public-key/certificate is determined to be authoritative for the identity. That portion of the overall solution is still being discussed in the IETF, and could be documented separately. This document only proposes how the identity signature itself could be created and encoded, such that it is usable once a certificate authority model and means of retrieving certificates are defined.

[Note: apologies about the term 'IKES' - this was not meant to cause confusion with Internet Key Exchange (IKE), but rather to provide something more important: a cute name for the draft, at least for baseball fans (Go Red Sox!)]

3. 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 a US National number of 212-555-1010, wants to communicate with Bob in example.org.uk, who has the international E.164 number +443069991010.

Alice is using SIP, so she generates an INVITE and puts her Address of Record (AoR) in the From header field of the request by using 'sip:2125551010@example.com' as the From URI. She then sends the INVITE to a proxy for her domain, which authenticates the request as well as Alice's ability to use the identity that is populated in the From header field.

The proxy, or some other system of the domain with the same knowledge, knows that Alice's SIP AoR is logically the international E.164 number '12125551010', and generates a string based on the source and destination identity numbers, with additional information to prevent replay attacks. This IKES information string is then encoded into a new SIP header field, along with a cryptographic signature of the hash of the information string. The key used for the signature is a private key, for which a corresponding public key is used by Bob's domain to verify the Caller-ID.

The proxy, as the holder of the private key for Alice's E.164 number, is asserting that the originator of this request has been authenticated and that she is authorized to claim the identity that appears in the From header field. How the public key is retrieved or determined by Bob's domain to be authoritative for the E.164 phone number is beyond the scope of this document.

The INVITE request from Alice's domain might be inter-worked to other protocols, such as SS7 or H.323, and Bob's domain might not even use SIP as its communication protocol. So long as Bob's domain receives the same IKES information and signature, it can determine if the originating Caller-ID is valid for the given message action type.

When Bob's domain receives the request, it verifies the IKES information and signature provided in whatever protocol field Bob's domain uses, and thus can validate that the Caller-ID is valid for the request.

For phone-number identities, Alice and Bob's domains need to have a means of providing and retrieving public keys in such a way that Bob's domain can know the holder of the private key is authorized to assert the E.164 number, regardless of what domain it originated from. For email-style identities, the same must be known for an assertion of Alice's domain name but not her specific user identity; instead, so long as Bob knows example.com signed the message, he can believe it is from Alice in example.com.

4. Background

The general concept of IKES is to generate a cryptographic signature similar to that in [RFC4474], except only for information that is necessary and sufficient to provide Caller-ID reputability. The information being signed is constrained so as to be protocol-agnostic, work through intermediaries, and with a resulting value that can work through and fit in the least-extensible protocol: SS7/ISUP.

Similar to [RFC4474], IKES concatenates specific information from the request into a string which is hashed, and the resulting hash is signed using the signer's private key; a verifier generates the same information from the message, generates the hash and verifies the signature using the public key of the signer. As noted previously, this document does not specify how the public key is retrieved, nor how it is authenticated to legitimately represent the identity it claims to be authoritative for. For example the public key might be retrieved from DNS, with the DNS query key representing the international E.164 number or email-style domain name and the DNS authority being trusted as the identity authority; or it might be retrieved in a certificate using HTTP, with a web-style PKI of a common trusted root Certificate Authority.

The IKES information and signature is exchanged between parties, by encoding it into defined fields of the various protocols and transporting it in their messages. For example by encoding it in a defined SIP header, XMPP XML element, or SS7/ISUP parameter. Gateway devices that interconnect the supported protocols need to copy the IKES information and signature from one protocol's field to another, although in the SS7/ISUP case this can be achieved by some intermediaries that are not the actual gateways, as described later.

Naturally, this mechanism only works if the IKES information successfully transits from the signer to the verifier. It might be possible to use an alternative means of exchanging the IKES information other than in the messages themselves, but that is beyond the scope of this document.

To provide protocol agnosticism, this document maps message types and fields from each of the supported protocols into an abstract type number space. This is necessary for consistent hash creation in both the signer and verifier roles. For example, a dialog-creating SIP INVITE request, XMPP session-initiation iq stanza, H.323 Call Setup, and ISUP IAM are all considered a new session initiation request with a message type of single character value 'I' when used in the hash input-string creation. Other protocols wishing to support IKES need to define their own mapping to the values defined in this document.

The information necessary to provide identity assurance, replay protection, and prevent cut-paste attacks is encoded into an 'IKES Information Field' (IKES-IF) string, and included in the message of the various protocols. This IKES-IF is what is actually hashed and signed. The receiving verifier generates some of the same information from protocol-specific headers (e.g., from the SIP From URI) and performs a string comparison with the received IKES-IF before calculating the hash and signature verification.

The protocol agnostic information in the IKES-IF covered by the IKES signature includes: the message type, source identity and type, the destination identity and type, a timestamp, sequence number, and a public-key index. Unlike [RFC4474], bodies in the message are not covered by the signature; IKES is used to provide caller-id reputability, and nothing more.

The source and destination identities are determined by generating either a "canonical" international E.164 or "canonical" email-style name from the given protocol's relevant fields. For example in SIP, the From and To URIs are used for this process. As explained later, however, it is the canonical form of the identities that are used for signing - not their literal string format encoded in the message. Thus even though a SIP message may contain a SIP URI of "sip:(212)555-1212@example.com", the signer internally generates a canonical form of an E.164 number as "12125551212", and signs that canonical form without actually changing the SIP URI in the message itself.

When the identities being signed are E.164 numbers, a copy of the canonical form of the source and destination identities are also encoded in the IKES-IF. The verifier still has to generate the information from the protocol's normal fields (e.g., SIP From/To URIs), but having them also encoded in the IKES-IF helps detect failures quickly without having to perform public key cryptography, and aids in troubleshooting.

The IKES Verifier performs validation of the message in a similar fashion as [RFC4474], by validating the signature for the message given the values received. Instead of using the received SIP Call-ID and Cseq pair to check a local cache for a unique signature, an IKES Verifier uses the IKES-provided sequence number value for such a cache instead. The IKES-provided timestamp is used to detect stale signatures, similar to the use of the SIP Date header field value in [RFC4474]. The combination of identities, sequence number, timestamp and message type fields provide protection against replay and cut-and-paste attacks, as described in the security section.

The public-key index field is used to indicate which of multiple public-keys a signer used to sign the message with, and thus which one the verifier should use to verify with. This is necessary to allow updating public-keys without Internet-wide synchronization, as well as to allow multiple public-keys to be useable for a given identity so that there can be multiple signers for a given E.164 number.

4.1. Identity Types: E.164 vs. Number Codes vs. Email-style

Most modern IP-based real-time communications protocols support two forms of source 'user' identities that IKES also provides a signature for: phone numbers, and "email-style" names. Due to confusion in common SIP usage, and to provide clarity on the differences involved and how IKES treats them, this section provides a background on the differences between the two identity forms.

IKES handles phone numbers in two ways: as international E.164 numbers, and as nationally-specific number codes. For the former case, IKES requires the phone number to be converted into a canonical international E.164 number format. For example, in SIP even if the From or To URI encode a phone number only using a regional or national format/length-of-digits, the IKES process requires adding the necessary country-code and regional prefix, to form the source or destination identity used for IKES. IKES also considers toll-free 1-8xx type numbers to be "E.164" numbers, even though technically they are not.

Nationally-specific number codes are numbers that are not E.164 numbers, nor are private numbering plan numbers, but are instead number codes used for specific purposes within national numbering plans. Examples of these are N11 codes in the NANP, two or three digit emergency numbers such as 112, and inter-carrier common short codes. Even if they cannot be used as caller-id numbers, they are destination numbers and thus IKES needs to support them. The important point is that they, like E.164 numbers, are not scoped to the domain name portion of a URI.

E.164 numbers are defined by the ITU as a sequence of up to 15 digits, including the leading country-code digit(s). They are of a global scope, meaning any two different E.164 numbers are globally unique and identify distinct logical entities. Any two equal E.164 numbers identify the same logical entity no matter where they are received from; they may not be from the same human, or phone/device, or even same the service provider, but in general users understand them to represent the same logical calling entity/organization.

In this document, the term "email-style names" are identifiers of the form 'user@domain', where the domain defines and limits the scope of authority of the user portion. In other words, for an email-style name the authority of the user portion is the domain after the '@' sign, and there is no confusion that the user 'alice@example.com' is the same as the user 'alice@invalid.com'. They may happen to be from the same Alice the human, but in general users understand they are distinct communication identities.

In theory, the protocol scheme of the 'user@domain' should also be considered a scoping element - i.e., 'sip:alice@example.com' and 'xmpp:alice@example.com' should be distinct identities - in practice, however, this is never the case except for some unique situations such as the username 'admin' or 'help'. IKES ignores the protocol scheme currently, and only deals with the 'user@domain' portion.

[Open issue: should the scheme matter? If so, there will be no way to go across different protocols]

Complications arise in SIP, because although E.164 numbers are processed correctly as globally unique identities, they are often encoded in email-style URI format; they are received in email-style format but are processed as E.164 numbers regardless. While there is a defined encoding for E.164 numbers in SIP, using the 'tel' URI scheme format, it is rarely used for SIP request source or destination identifiers. Instead, the SIP URI format is used with a 'sip' scheme, with or without a 'user=phone' parameter; for example 'sip:+1212551212@example.com' is processed by SIP devices as the E.164 number '1212551212', ignoring the domain portion.

For the purposes of this document, the term "email-style" name does **NOT** apply to E.164 numbers or nationally-specific number codes encoded in SIP URI form; instead, those are still termed and considered E.164 numbers or nationally-specific number codes. If the verifying domain/node processes the received message's source identity as an E.164 number, or displays it to their end users as such - no matter what encoding format it takes on-the-wire - then the verifying domain/node *MUST* use the verification rules defined for E.164 numbers.

4.2. Determining Canonical E.164 Numbers and Number Codes

In [RFC4474] the SIP From and To header fields are covered by the signature; if the From or To URI change by the time they reach the verifier, the signature would become invalid. Unfortunately, From and To URIs are changed quite frequently by middleboxes, as described in [draft-fromto-change]. This document proposes that the important components of the To and From URI that need to be protected are the source and target identity, not the whole header fields, and not even the whole URIs; just the E.164 number, number code, or email-style name.

One of the difficulties with signing the URI username portion for E.164 numbers, however, is that even the username number changes frequently along the SIP routing path. It may be prefixed with leading digits used for local routing purposes, or it may translated to/from a local or national numbers rather than the full E.164

number, or it may have visual separators encoded in the username portion. Therefore, if the originating domain simply signs the From/To username as a string of digits, and the digits get changed by the time the request reaches the verifying domain, the verification will fail.

To prevent this from happening, part of the IKES signing and verification process involves determining a 'canonical' source and/or destination E.164 number or number code, if the request's source or destination identities are of an E.164 type or number code type, instead of email-style names.

The process for determining a canonical E.164 number cannot be fully specified in this document, because it will likely depend on the specific policies used within the local domain. For example one domain may only use local number formatting and need to convert all To/From user portions to E.164 by prepending country-code and region code digits; another domain might prefix usernames with trunk-routing codes and need to remove the prefix.

Regardless of the process used, the resulting canonical E.164 numbers used by both the signing and verifying systems MUST result in an ASCII string of only digits without whitespace or visual-separators, starting with the country-code digit(s). This canonical representation of the E.164 number is used as input to the hash calculation during signing and verifying process.

Likewise for nationally-specific number codes, the IKES process generates a canonical representation of the number code. A leading country-code is prepended to the number code, to indicate the national numbering plan they are for.

Although the canonical representations are implicitly generated from the received message, rather than being simply explicitly copied from the URI, a copy of each canonical source/destination number is inserted in the message for certain protocols (e.g., SIP and XMPP), to detect mismatches quickly and aid troubleshooting.

4.3. Determining Canonical Email-style Names

If the originating domain intends the source or destination identities to be "email-style" names rather than phone numbers, then it MUST generate a canonical form of the name when it generates the IKES information that is signed. The process for this results in a simple user@domain formatted string, without whitespace, without a scheme or parameter or additional resource information.

4.4. Call-forwarding Issues

Many communication services offer a "call forwarding" feature, whereby a user can have communication requests that were originally destined to them, forwarded on to other destination identities. This can cause issues for caller-id authentication mechanisms such as IKES, because IKES needs to know the destination identity in order to prevent cut-paste attacks. The destination identity is thus included in the string that is signed by the IKES Generator. Therefore, if the destination identity encoded in the protocol message changes, some means of determining the original one needs to be available for IKES to succeed.

In SS7/ISUP, the process of forwarding/redirecting the call changes the Called Party Number value to the new destination identity, and copies the original destination identity into an Original Called Number parameter, along with generating other redirection information. Thus for SS7 it is not difficult to determine the identity for IKES to use in signature validation.

For SIP, in theory the process of call forwarding leaves the To header field URI unchanged, while only the Request-URI changes to the new destination identity. In practice, however, most systems follow the SS7 model and change the To URI to be the new destination identity. SIP has a means of recording the original destination identity: either in History-Info header fields per [RFC4244], or in Diversion header fields per [RFC5806]. In order to help determine the original destination identity, IKES encodes the one it used for signing in the new SIP header field used for IKES.

Regardless of the specific protocol mechanics, however, call-forwarding introduces a significant weakness in any caller-id verification mechanism. Imagine if a malicious Bob could re-use the IKES information from a call he received from a bank or airline, for example, to then generate new calls to Charlie and other random users. At a protocol layer, Bob could simply behave as if the call was being forwarded. It would appear to Charlie that there is a valid caller-id from the bank or airline, when it's really Bob abusing the system.

Ultimately the receiver of the message that verifies the IKES information needs to determine if the IKES-protected identities are valid for the final user that receives the call. In other words, if Alice calls Bob, and Bob forwards all his calls to Charlie, then Charlie's verification system has to determine if Charlie trusts calls being forwarded from Bob (i.e., whether Charlie allows Bob to forward calls to him).

This document does not define a specific means of performing this forwarding-party authorization, but the author believes some guidelines should be given for how to display the caller-id properly in such scenarios, or perhaps how to use XCAP or other means for end users to provision trusted call-forwarders. This is left as an open issue for now.

5. IKES Generator Behavior

This document defines a mechanism by which the sender of communications messages can cryptographically assert the authenticity of the originator's E.164, number code, or email-style identity; this role is called an IKES Generator. The IKES generator may be an end host such as a PC or mobile phone, or it may be the originating Enterprise or Service Provider. Whoever has an appropriate private key for a given identity, and the means to authenticate the message originated from the indicated identity, can be an IKES Generator for the message.

Any entity which performs the role of IKES Generator MUST possess the private encryption key of an asymmetric key pair that can be used to sign for a given E.164 number, number code, or email-style domain name, depending on the source identity type. The public key half of this pair must be available to any receiver of the message, such that the receiver will be able to verify the sender sent the message and can claim the source identity. This may involve using certificates signed by a trusted third-party, identifying the E.164 number, number code, or domain name; or it may involve some other means of retrieving the public key for the given E.164 number, number code, or domain name. The exact means of retrieving and authenticating the public key is beyond the scope of this document.

The IKES mechanism relies on the node or domain asserting the source identity to perform some form of authentication for the identity it asserts. For example, in SIP it may digest-challenge a request before signing. If the IKES Generator does not perform sufficient authentication of the source identity, then it only impacts the legitimacy of E.164 numbers it is responsible for, or email-style names for its own domain - it does not impact the legitimacy of E.164 numbers it cannot claim to represent.

The role of the IKES Generator is to perform the following steps, in order, and sign the resulting information. The specific steps it MUST perform are:

Step 1:

The IKES Generator MUST map the protocol-specific message type into a generic IKES 'message type' value. For example a SIP INVITE is the message type 'I'. This message type value is encoded into the message and signed, to prevent cut-paste attacks.

Step 2:

The IKES Generator MUST extract the canonical identity of the sender from the appropriate message field, into a new string referred to as the 'source identity'. For example in SIP, this would be the logical international E.164, number code, or email-style source identity in the From header field URI, canonicalized into a new string used for signing. The type of the identity MUST also be determined: either an E.164, number code, or email-style type. This type information will be encoded in the message and signed, to prevent obfuscation attacks.

If the IKES Generator cannot verify the source identity claim, it MUST NOT generate or insert IKES fields. Doing otherwise is not in the best interest of the IKES Generator, as it would be signing an assertion for an identity it does not know to be accurate, and thus may lead to impacting the reputation of its assertions.

Step 3:

The IKES Generator MUST extract the canonical identity of the target identity from the appropriate message field, into a new string referred to as the 'destination identity field'. For example in SIP, this would be the international E.164 number, number code, or email-style source identity in the To header field URI, canonicalized into a new string used for signing. The canonical destination identity is also encoded into the message, useful for both troubleshooting and call-forwarding scenarios as described later.

The type of the destination identity MUST also be determined: either an international E.164 number, number code, or email-style type. This type information will be encoded in the message and signed, to prevent obfuscation attacks.

Step 4:

The IKES Generator MUST generate a new sequence number, which is encoded into the message and signed. The sequence number-space size is 24-bit, and the value increases by one every time an IKES Generator creates a new signature for any message of any message type. The sequence number is encoded into the message and signed, in order to prevent replay attacks within the timestamp's validity window.

Step 5:

The IKES Generator MUST generate a value for the current time based on UTC, referred to as the 'timestamp'. The timestamp value is encoded into the message itself and included in the signature calculation, and is used by the IKES Verifier to detect stale signatures and prevent replay/cut-paste attacks.

Step 6:

The IKES Generator MUST form a string of the information generated in previous steps as well as the private key's index, create a hash of the string, and generate the IKES signature using its appropriate private key for the source identity. The signature is encoded into the message, and used by the IKES Verifier to validate the source identity for the given message type.

Finally, the IKES Generator MUST forward the message normally.

6. IKES Verifier Behavior

This document introduces a new logical role for communication entities called an IKES Verifier. When an IKES Verifier receives a message containing an IKES signature, it may inspect the signature to verify the source identity for the message. Typically, the results of the verification are provided as input to an authorization process that is outside the scope of this document.

If an IKES signature is not present in a message or is invalid, it is up to local policy to dictate what action should occur, such as forwarding a call request to an attendant or IVR, or anonymizing the source identity and blocking it from being displayed or used, or even rejecting the request.

In order to verify the identity of the sender of a message, an entity acting as a verifier MUST perform the following steps, in the order here specified.

Step 1:

The verifier MUST map the protocol-specific message type into a generic IKES 'message type' value. For example a SIP INVITE is the message type 'I'. This forms the initial value in the IKES-IF string.

If the specific protocol being used has the IKES-IF string available, the verifier MAY immediately check that the IKES-IF 'msg-type' value matches the verifier's generated one. This is an optimization step, to detect mismatches quickly.

Step 2:

The verifier MUST extract the canonical identity of the sender from the appropriate message field. For example in SIP, this would be the E.164, number code, or email-style source identity in the From header field URI, internally converted into a canonical form. The type of the identity MUST also be determined: either an E.164 or email-style type. The identity type and canonical value are the next set of values for the IKES-IF string.

If the specific protocol being used has the IKES-IF string available, the verifier MAY immediately check that the IKES-IF 'source-id' value matches the verifier's generated one. This is an optimization step, to detect mismatches quickly.

Step 3:

The verifier MUST extract the canonical identity of the destination identity from the appropriate message field. For example in SIP, this would be the E.164, number code, or email-style identity in the To header field URI, internally converted into a canonical form. The type of the destination identity MUST also be determined: either an E.164 or email-style type. The identity type and canonical value are the next set of values for the IKES-IF string.

If call-forwarding/redirection has occurred, then the original target number/name is used for the destination identity. Determining whether redirection has occurred, and where to get the destination identity in such a case, is protocol-specific and covered in later sections for the specific protocol usage.

The verifier MUST validate that the destination identity encoded in the message either identifies the resource it will forward the message to, or that the resource it will forward the message to is willing to accept messages addressed for that identity. How the verifier determines the destination identity is of such a type is beyond the scope of this document. One example would be using a

list of URI's the forwarded-to user has populated on the verifier through a service/account portal.

If the specific protocol being used has the IKES-IF string available, the verifier MAY immediately check that the IKES-IF 'dest-id' value matches the verifier's generated one. This is an optimization step, to detect mismatches quickly as well as to detect redirections in some cases.

Step 4:

The verifier MUST validate the received IKES-IF timestamp falls within ten minutes of local time: either 10 minutes earlier or later than local system time in UTC. This avoids the need to synchronize clocks, and allows time for messages to be sent through multiple intermediaries/domains. If the public key obtained in Step 6 is from a certificate, the verifier must furthermore ensure that the value of the timestamp falls within the validity period of the certificate.

Step 5:

The verifier generates the full IKES-IF string from the fields determined above in previous steps, along with the received IKES-IF sequence number and public key index value. The verifier MUST verify it has not received this same IKES-IF string in the past 20 minutes (1200 seconds).

The purpose of this step is to prevent replay attacks. Since the IKES-IF string contains the sequence number, as well as the identities and other fields, this check will prevent the same message from being replayed. If the IKES-IF had been received previously, then it is either a replay attack or forked messages that have merged at the verifier. In either case the redundant/repeated message(s) can be rejected.

Step 6:

The verifier MUST acquire the public key for the source identity. The process for doing this is beyond the scope of this document.

Step 7:

The verifier MUST verify the IKES signature, following the procedures for generating the hashed digest-string described in Section 7.

If a verifier determines that the signature on the message does not correspond to the reconstructed IKES-IF string, then it must reject

the message or perform the actions local policy dictates for invalid source identities.

Step 8:

If all of the above checks pass/succeed, then the verifier MUST remember the valid IKES-IF string for the next 20 minutes, in order to be able to perform Step 5 for future messages. The IKES-IF string MUST NOT be remembered if it was not valid (i.e., if this Step 8 was not reached). Otherwise an attacker could generate invalid IKES-IF messages to prevent legitimate calls.

7. IKES Information Field

This document specifies an 'IKES Information Field' (IKES-IF), which is a UTF-8 string formatted in a specific manner. The IKES-IF is the canonical string created by the IKES Generator that is then hashed and signed; and the IKES-IF is re-created by the Verifier and the signature verifies its hash.

The IKES-IF string's 'match-fields' portion contains information that is determined from other message fields, while the 'valid-fields' portion contains information added by the IKES Generator for the purposes of IKES.

For SIP and XMPP protocol usages defined in this document, the entire IKES-IF literal string is itself encoded into the message in a new SIP header or XMPP XML element. This is done to provide quick mismatch detection and for troubleshooting purposes.

For SS7, H.323, and native ISDN, the IKES-IF information is encoded in a different manner in the message fields, due to the constraints of SS7/ISUP parameter sizes. In particular, the 'match-fields' portion is only auto-generated by both the IKES Generator and Verifier, while the 'valid-fields' portion is encoded in specific protocol fields. However the same full IKES-IF string is used for the hashing and signing, regardless of the protocol.

Note that this section only defines the field syntax, not the protocol-specific encoding; those are defined in the protocol-specific sections later in this document. The ABNF grammar for the IKES-IF is:

```
ikes-if      = match-fields "=" valid-fields
match-fields = msg-type "=" source-id "=" dest-id
valid-fields = sequence "=" key-index "=" timestamp
msg-type     = CHAR
source-id    = identity
dest-id      = identity
identity     = domain-based / global-e164 / number-code / token
domain-based = "D:" user "@" host
global-e164  = "G:" *DIGIT
number-code  = "C:" *DIGIT
sequence     = 1*8 DIGIT ;values 1 - 16777215
key-index    = 1*4 DIGIT  ;values 1 - 1023
timestamp    = date-time ;from [RFC3339]
```

For example, a SIP INVITE from the E.164 number '12125551212' to the E.164 number '443069991010', of sequence number 1216, using public-key index number 4, with an IKES signature generated at 1:15:30PM UTC on July 16, 2013, would result in the following IKES-IF string:

```
I=G:12125551212=G:443069991010=1216=4=2013-07-16T13:15:30Z
```

This string would then be hashed, and the resulting hash would be signed using the appropriate private key. The IKES-IF string is also encoded in its literal form in a new SIP header field defined later, as well as in a new XMPP XML element defined later.

The same example, but to the US-based number code '911', would result in the following IKES-IF string:

```
I=G:12125551212=C:1911=1216=4=2013-07-16T13:15:30Z
```

When an email-style domain-based identity is used for the source or destination identities, the canonical form of the user@domain identity is also included in the IKES-IF in the 'domain-based' field, so that the signature covers the identity.

For example, a SIP INVITE from the canonical name 'alice@foo.com' to 'bob@bar.co.uk', of sequence number 1216, using public-key index number 4, with an IKES signature generated at 1:15:30PM UTC on July 16, 2013, would result in the following full IKES-IF string:

```
I=D:alice@foo.com=D:bob@bar.co.uk=1216=4=2013-07-16T13:15:30Z
```

The above string would be then hashed and signed, using a private key valid for asserting the domain foo.com.

Having a source identity of one type and a destination identity of a different type is completely valid. Therefore an example such as this is possible:

```
I=G:12125551212=D:bob@bar.co.uk=1216=4=2013-07-16T13:15:30Z
```

The 'msg-type' field is a single character representing the message action type: for example an 'I' for call/session initiation request, 'U' for mid-session change request, 'X' for session transfer request, or 'M' for instant-message request. The appropriate message type field value to use for specific protocol messages is defined in later sections, for each protocol.

The 'source-id' and 'dest-id' fields represent the source identity and destination identity of the message. For E.164-based identities, the 'global-el64' syntax is used; for email-style identities, the 'domain-based' syntax is used.

The 'sequence' field represents a sequence number. The sequence number MUST be guaranteed to be unique for each message, for at least 10 minutes (600 seconds) after the time field value. Message retransmission at the protocol transport layer MUST NOT generate a new sequence number, and thus the same signature value will be used for retransmissions. In other words, IKES authentication and validation occur at a layer above the transport layer. The sequence number prevents replay attacks within the valid duration of the timestamp.

The 'key-index' field identifies which specific public-key index to use. Regardless of whether DNS or HTTP or some other mechanism is used for retrieving the public key, there will likely be more than one active public-key for a given domain name or E.164 phone number.

The 'timestamp' field identifies the UTC time at which the IKES-IF was generated. A verifier will only accept a time value of up to 10 minutes early or 10 minutes late. Since this field is also signed, it cannot be changed and thus helps prevent replay attacks.

8. Usage in SIP

This document defines a new SIP header field for carrying the IKES-IF data and signature: the 'Likes-If' header, for "Literal IKES-IF". The format of the Likes-If header field is as follows, using the previously defined ABNF fields:

field, and if so then it MAY use that matching value to continue the verification processing.

9. Usage in XMPP

This document defines a new XMPP XML child element of the 'message', 'presence', and 'iq' elements for carrying the IKES-IF data: the 'likesIf' element, for "Literal IKES-IF". The element value's type is an xs:token data type, and contains the full IKES-IF data string defined in section 7 for 'ikes-if'. [note: alternatively we could define the ikес-if fields to be syntactically broken out into distinct XML elements/attributes] It is TBD how the signature and algorithm information are encoded, for example in another XML child element, or whatever.

The following mapping is used for XMPP stanza types to IKES message types:

- o An iq stanza of type "set" with a jingle action attribute of "session-initiate" is the IKES message type of 'I'.
- o An iq stanza of type "set" with a jingle action attribute of "content-add", "content-modify", or "content-remove" is an IKES message type 'U'.
- o An iq stanza of type "set" with a jingle action attribute of "session-terminate" is the IKES message type of 'B'.
- o Other iq stanza types are currently undefined for IKES.
- o A message stanza with no type or of type "normal" is the IKES message type 'M'.
- o A message stanza of the type "chat" is the IKES message type 'I'; this follows the model of a SIP INVITE for MSRP sessions.
- o A message stanza with a jabber multi-user chat "invite" element is the IKES message type 'X'; this follows the model of a SIP REFER.
- o A presence stanza of type "subscribe" or "unsubscribe" is the message type 'S'.
- o A presence stanza of type "subscribed" is the message type 'N'.
- o A presence stanza with no 'type' attribute is currently undefined. [note: what is the correct semantics of this? It appears to server multiple message roles/types for SIP, for example]
- o [note: is there an XMPP type for transferring a media session? That would be IKES message type 'X']

The source identity SHOULD be based on the 'from' attribute of the stanza. The destination identity SHOULD be based on the 'to' attribute of the stanza.

10. Usage in SS7/ISUP

The SS7/ISUP protocol allows new ISUP parameter types to be defined, but doing so would require many SS7 devices to change behavior. The SS7 market is fairly stagnant, and in the final stages of life, making such a change untenable. Therefore, instead of defining a new ISUP parameter type, IKES uses existing ISUP parameters to carry its data: the sequence number is carried in the ISUP Call Reference parameter using a fixed/defined point code; the signature, key index, and timestamp are carried in the User-to-User Information parameter; the source identity is the Calling Party Number, and the destination identity is either the Called Party Number or carried in the Original Called Number parameter if the call has been redirected.

There is no guarantee the optional ISUP parameters used by IKES will successfully transit the PSTN. The odds are high for short hops - for example across a single link/connection, or possibly within a single SS7 carrier - but it will rarely make it across multiple SS7 carriers or across international links. The expectation is that SS7 usage will continue to decline and be replaced with SIP interconnection, making IKES more and more usable end-to-end.

SS7 support for IKES only supports voice call service through ISUP, and not instant messages such as would be sent as SMS in SS7/MAP. Also, only en bloc signaling is supported, and not overlap signaling.

Although ISUP parameters can be up to 255 octets long, in practice the User-to-User Information parameter is restricted to 131 octets. This makes it more complicated for IKES to use, and is why the operations defined in this section are cumbersome.

Instead of encoding the IKES-IF as one long string, the individual components are encoded as discrete fields in binary form.

The IKES sequence number is encoded as the reference number in the Call Reference parameter, with a fixed point code yet to be defined/reserved.

The IKES key index, timestamp, signature, and algorithm are encoded in the User-to-User Information parameter based on the following format:

0	1	2	3
0 1 2 3 4 5 6 7 8 9	0 1 2 3 4 5 6 7 8 9	0 1 2 3 4 5 6 7 8 9	0 1
+-----+			

Internet-Draft	STIR IKES OUT	July 2013
	128 octet signature	
\		\
/		/
+-----+		
	key index alg timestamp	
+-----+		

The 'signature' field is the 128 binary bytes of the IKES signature. This is only sufficient to hold the signature based on a 1024-bit key. If a longer key is necessary, IKES cannot work through SS7.

The 'key index' field is a 10-bit unsigned integer, and identifies the IKES public-key index value.

The 'alg' field identifies the signature algorithm. This document currently only specifies sha1WithRSAEncryption as described in [RFC3370], and that this algorithm has the 'alg' value of 0x00. Future algorithms need to specify one of the remaining 3 possible values for this field if they wish to transit SS7.

The 'timestamp' field identifies the lower 12 bits of the IKES timestamp, when converted to number of seconds that have elapsed since 00:00:00 UTC, Thursday, January 1, 1970 (known as Unix time). Leap seconds do not matter, since the generator and verifier do not need to have accurately synchronized clocks.

When an IKES Generator creates this value, or an SS7/ISUP gateway converts an IKES-IF into the ISUP parameters, it takes the IKES-IF info-field UTC time and converts it to Unix time and encodes the lower 12 bits into this ISUP UUI parameter's timestamp field. The signature, however, is calculated over the full UTC timestamp as encoded in an IKES-IF info-field.

An IKES Verifier or ISUP gateway receiving this ISUP parameter timestamp field, which needs to either verify the signature or convert the parameters into a different IKES-IF encoding form, needs to be able to re-create the full *original* UTC time from these lower 12 bits, or else the signature will fail. This can be accomplished because 12 bits represent 4096 seconds of time, and the verifier can assume the originator generated the parameters within a 1200 second window of local time (either 600 seconds earlier, or 600 seconds in the future); if the originator generated it outside of that window, then the resulting IKES-IF string will fail the signature check anyway. Therefore, the verifier need only compare the received lower 12 bits against its valid local Unix time range, and generate a full UTC timestamp that fits within the window and has the same lower 12 bit value - there can be only one such value.

If it cannot fit in the valid range, then it knows the signature is stale; if it does fit but the resulting signature check still fails then either it is an invalid caller-id, or the signature is so stale as to have wrapped the 12-bit seconds value by being over 4096 seconds old, or something else is wrong with the IKES-IF. Regardless of the verification result, the full UTC timestamp can be successfully recreated for all successful signature result cases, and might be incorrectly recreated for cases where the IKES result would be a failure anyway.

The following mapping is used for ISUP message types to IKES message types:

- o An IAM message is the IKES message type of 'I'.
- o An REL message is the IKES message type of 'B'.
- o Other ISUP message types are currently undefined for IKES.

The IKES source identity is the Calling Party Number parameter. The IKES destination identity is either the Called Party Number, or the Original Called Number if the call has been redirected. Both identities need to be canonicalized to a global E.164 number, if they are a local or national format.

10.1. SIP-SS7 Interworking

A likely use-case for IKES is the need to transit an SS7 carrier between SIP domains. This can be accomplished by the SIP-SS7 gateway itself, by interworking the SIP IKES-IF to/from the SS7 parameters described earlier.

Another possibility is to have a SIP-only device generate the information to/from an ISUP body following the [SIP-I] or [SIP-T] model. So long as the SS7 gateway supports SIP-I or SIP-T, including the Call Reference and User-to-User Information parameters, this can be a viable alternative. It is not uncommon for SBCs, Application Servers, and other call control systems to have the ability to generate SIP-I/SIP-T ISUP bodies and parameters from SIP header field information, and vice-versa. Since the market for such devices is still flourishing, it is reasonable to expect support for such an IKES interworking model.

11. Usage in H.323 and ISDN

Both the H.323/H.225 and native ISDN/Q.931 protocols support adding new optional fields to their defined messages, but like SS7 it is unlikely doing so would result in vendors making the necessary

changes because the market for H.323 and ISDN is dying and getting replaced with SIP. So our choices are to either use existing fields, or ignore H.323 and ISDN as use-cases for IKES. This decision is still TBD and an open issue for the STIR Working Group.

If we choose to use existing fields, the same model as that used for SS7 can be used for H.323: the format and information defined for the ISUP User-to-User Information parameter previously, would be encoded into the H.323 user-data user-information field. ISDN Q.931 also supports a User-to-User Information field, limited to 131 octets like the one in SS7/ISUP; but its Call Reference field cannot be used the way the SS7/ISUP one can, because the ISDN one is already used for other purposes today. Suitable H.323 and ISDN fields for the IKES sequence number would have to be found.

For both H.323 and ISDN, the following mapping could be used for messages:

- o A SETUP message is the IKES message type of 'I'.
- o A RELEASE message is the IKES message type of 'B'.
- o Other H.323 or ISDN message types are currently undefined for IKES.

[Note: more needs to be defined for H.323 and for native ISDN, if the WG decides it's worth the effort]

12. Open Issues

There are still many open issues in this draft. It is currently a straw-man proposal. Some of the bigger open issues are:

- o Whether support for SS7 is worth specifying or not.
- o Whether using the SS7/ISUP Call Reference parameter is possible or not.
- o Whether instant messages for SMS in SS7/MAP needs to be handled as well or not.
- o Whether support for H.323 and native ISDN is worth specifying or not, and if so then how to encode the fields.
- o Specific details for XMPP encoding syntax, and stanza handling.
- o Whether 1024-bit key size is sufficient or not.
- o Whether IKES should replace RFC 4474 or not.
- o How to handle forwarding-party authorization for call-forwarding scenarios.

13. Security Considerations

The considerations in [RFC4474] generally apply to this document's proposed mechanism as well, and will not be repeated here. There

are several additional security consideration when using this mechanism, however, as follows:

- 1) The IKES mechanism does not sign the Contact URI value, and thus a malicious party can change the value without detection. For most SIP use scenarios, this is no worse than [RFC4474], since Record-Route and Path header fields can be added into [RFC4474] signed SIP requests as well to accomplish the same malicious goal. The Contact URI is usable, however, in cases where Record-Route and Path do not apply, for example to generate subsequent out-of-dialog requests to a GRUU Contact; in such cases the IKES mechanism is weaker than [RFC4474].
- 2) The IKES mechanism does not sign the SIP message body, and therefore much of the SDP can be changed without detection. Although [RFC4474] only signs bodies when they are in requests - which is not always the case for SDP - if the SDP body *is* in the request then [RFC4474] assures the Verifier that it has not been changed by any node beyond the Authenticator. For SDP, such assurance does not guarantee media identity (see [draft-baiting]), but [RFC4474] is better than nothing. The IKES mechanism does not do this because in practice SDP is constantly changed as requests pass through intermediate domains, and is thus a wasted effort. Furthermore, it would reveal that the SDP had been changed, which service providers might not want to reveal to certain parties.
- 3) The IKES mechanism does not always prevent malicious replay attacks if the verifying domain uses multiple, separate verification systems to verify caller-ids. In such a case, it's possible for replayed messages sent within the valid time window to be received on different verifier systems, so that each verifier would not detect it as a replay because they didn't receive the original message. One solution to this is for all the verifiers to use a common database to hold and retrieve the previously accepted IKES-IF strings for the time window duration. Such a solution, however, would be complicated to manage and is likely unnecessary given the current threat model.

14. IANA Considerations

This document makes no request of IANA yet - if this document moves forward, then requests of IANA will be made here.

15. Acknowledgments

The idea of being able to canonicalize SIP To/From URIs into E.164 numbers comes from Brian Rosen. The idea of IKES in general is not new, and has been discussed for years in the IETF.

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).

16. References

16.1. Normative References

[RFC4474] Peterson, J., Jennings, C., "Enhancements for Authenticated Identity Management in the Session Initiation Protocol (SIP)", RFC 4474, August 2006.

[RFC3261] 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.

16.2. Informative References

[RFC4244] Barnes, M., "An Extension to the Session Initiation Protocol (SIP) for Request History Information", RFC 4244, November 2005.

[RFC5806] Levy, S., Mohali, M., "Diversion Indication in SIP", RFC 5806, March 2010.

[SIP-I] ITU-T, Q.1912.5, and Q.761-Q.764.

[SIP-T] Vemuri, A., Peterson, J., "Session Initiation Protocol for Telephones (SIP-T): Context and Architectures", RFC 3372, September 2002.

[RFC3204] Zimmerer, E., Peterson, J., Vemuri, A., Ong, L., Audet, F., Watson, M. and M. Zonoun, "MIME media types for ISUP and QSIG objects", RFC 3204, December 2001.

[draft-baiting] Kaplan, H., "The SIP Identity Baiting Attack", draft-kaplan-sip-baiting-attack-02, February 2008.

Internet-Draft
Author's Address

STIR IKES OUT

July 2013

Hadriel Kaplan
Oracle
Email: hadriel.kaplan@oracle.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 14, 2014

E. Rescorla
RTFM, Inc.
July 13, 2013

Secure Caller-ID Fallback Mode
draft-rescorla-stir-fallback-00

Abstract

A major challenge with RFC 4474-style identity assertions has been that SIP operates in highly mediated and interworked environments. SIP requests may pass through gateways, policy enforcement devices or other entities that receive SIP requests and effectively act as user agents, re-initiating a request. In these circumstances, intermediaries may recreate the fields protected by the RFC4474 signature, making end-to end integrity impossible. This document describes a mechanism for two compliant endpoints to exchange authentication data even in the face of intermediaries which remove all additional call signaling meta-data or which translate from SIP into protocols incapable of understanding identity meta-data (e.g., where one side is the PSTN).

Legal

THIS DOCUMENT AND THE INFORMATION CONTAINED THEREIN ARE PROVIDED ON AN "AS IS" BASIS AND THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY, THE IETF TRUST, 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 THEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Status of this Memo

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

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

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

material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 14, 2014.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

Table of Contents

1. Introduction	4
2. Operating Environment	4
3. Architectural Options	5
4. Strawman Architecture	6
4.1. Phone Number Authentication	6
4.2. Call Placement Service	6
4.3. Security Analysis	7
4.3.1. Substitution Attacks	8
5. Some Potential Enhancements	9
5.1. Encrypted CPRs	9
5.2. Signed CPRs	9
5.3. Credential Lookup	10
5.4. Federated Verification Services	10
5.5. Escalation to VoIP	10
6. Security Considerations	11
Appendix A. Acknowledgements	11
Author's Address	11

1. Introduction

A natural design for providing caller authentication is to attach a signature to the call setup messages (e.g., a SIP INVITE). This is incompatible with much of the existing communications environment. Most calls from telephone numbers still traverse the PSTN at some point. Broadly, these calls fall into one of three categories:

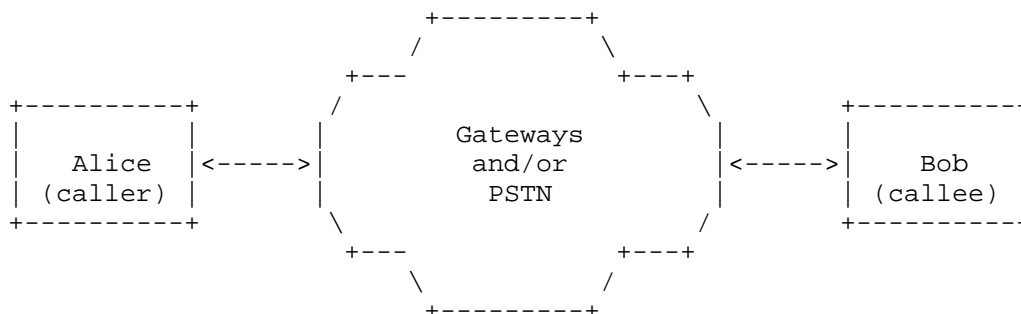
- o One or both of the endpoints is actually a PSTN endpoint.
- o Both of the endpoints are non-PSTN (SIP, Jingle, ...) but the call transits the PSTN at some point.
- o Non-PSTN calls which do not transit the PSTN at all.

The first two categories represent the vast majority of these calls. The network elements that operate the PSTN are legacy devices that are unlikely to change at this point. However, these devices are also unlikely to pass signatures--or indeed any inband signaling data--intact. In many cases they will strip the signatures; in others, they will damage them to the point where they cannot be verified. In either case, any in-band authentication scheme does not seem practical in the current environment.

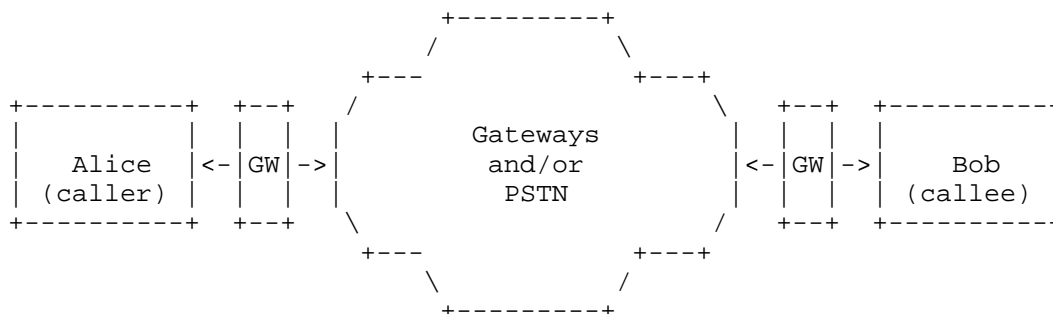
While the core network of the PSTN remains fixed, the endpoints of the telephone network are becoming increasingly programmable and sophisticated. Landline "plain old telephone service" deployments, especially in the developed world, are shrinking, and increasingly being replaced by three classes of intelligent devices: smart phones, IP PBXs, and terminal adapters. All three are general purpose computers, and typically all three have Internet access as well as access to the PSTN. This provides a potential avenue for building an authentication system that changes only the endpoints while leaving the PSTN intact.

2. Operating Environment

This section describes the environment in which the proposed mechanism is intended to operate. In the simplest setting, Alice is calling Bob through some set of gateways and/or the PSTN. Both Alice and Bob have smart devices which we can modify, but they do not have a clear connection between them: Alice cannot inject any data into the system which Bob can read, with the exception of her asserted E.164 number. Thus, this number is the only value which can be used for coordination.



In a more complicated setting, Alice and/or Bob may not have a programmable device, but have a programmable gateway that services them, as shown below:



In such a case, Alice might have an analog connection to her gateway/switch which is responsible for her identity. Similarly, the gateway would verify Alice's identity, generate the right caller-id information and provide caller-id information to Bob using ordinary POTS mechanisms.

3. Architectural Options

Because endpoints cannot communicate directly, any solution must involve some rendezvous mechanism to allow endpoints to communicate. We call this rendezvous service a "call placement service" (CPS). In principle they could communicate any information, but minimally we expect it to include a "call placement record" (CPR) that describes the caller, callee, and the time of the call. The callee can use the existence of a CPR for a given incoming call as rough validation of the asserted origin of that call. (See Section 6 for limitations of this design.)

There are roughly two plausible dataflow architectures for the CPS:

- o The callee registers with the CPS. When the caller wishes to place a call to the callee, it sends the CPR to the CPS which forwards it to the callee.
- o The caller stores the CPR with the CPS at the time of call placement. When the callee receives the call, it contacts the CPS and retrieves the CDR.

While the first architecture is roughly isomorphic to current VoIP protocols, it shares their drawbacks. Specifically, the callee must maintain a full-time connection to the CPS to serve as a notification channel. This comes with the usual networking costs to the callee and is especially problematic for mobile endpoints. Thus, we focus on the second architecture in which the PSTN incoming call serves as the notification channel and the callee can then contact the CPS to retrieve the CPR.

4. Strawman Architecture

In this section, we discuss a strawman architecture along the lines described in the previous section. This discussion is deliberately sketchy, focusing on broad concepts and skipping over details. The intent here is merely to provide a rough concept, not a complete solution.

4.1. Phone Number Authentication

We start from the premise that each phone number in the system is associated with a set of credentials which can be used to prove ownership of that number. For purposes of exposition we will assume that ownership is associated with the endpoint (e.g., a smartphone) but it might well be associated with a gateway acting for the endpoint instead. It might be the case that multiple entities are able to act for a given number, provided that they have the appropriate authority. The question of how an entity is determined to have control of a given number is out of scope for this document.

4.2. Call Placement Service

An overview of the basic calling and verification process is shown below. In this diagram, we assume that Alice has the number +1.111.111.1111 and Bob has the number +2.222.222.2222.

```

Alice                               Call Placement Service          Bob
-----
<-  Authenticate as 1.111.111.1111  ---->

Store (1.222.222.2222,1.111.111.1111) ->

Call from 1.111.111.1111 ----->

                                <-  Authenticate as 1.222.222.2222  ---->

                                <----- Retrieve call record
                                    from 1.111.111.1111?

                                (1.222.222.2222,1.111.111.1111) -->

                                [Ring phone with callerid
                                    = 1.111.111.1111]

```

When Alice wishes to make a call to Bob, she contacts the CPS and authenticates to prove her ownership of her E.164 number. Once she has authenticated, she then stores a Call Placement Record (CPR) on the CPS. The CPR should also have some sort of timestamp to prevent replay. The CPR is stored under Alice's number.

Once Alice has stored the CPR, she then places the call to Bob as usual. At this point, Bob's phone would usually ring and display Alice's number (+1.111.111.1111), which is provided by the usual caller-id mechanisms (i.e., the CIN field of the IAM). Instead, Bob's phone transparently contacts the CPS and requests any current CPRs from Alice. The CPS responds with any such CPRs (assuming they exists). If such a CPR exists, he can then present the callerid information as valid. Otherwise, the call is unverifiable. Note that this does not necessarily mean that the call is bogus; because we expect incremental deployment many legitimate calls will be unverifiable

4.3. Security Analysis

The primary attack we seek to prevent is an attacker convincing the callee that a given call is from some other caller C. There are two scenarios to be concerned with:

- o The attacker wishes to simulate a call when none exists.
- o The attacker wishes to substitute himself for an existing call as described in Section 4.3.1

If an attacker can inject fake CPRs into the CPS or in the

communication from the CPS to the callee, he can mount either attack. In order to prevent this, either the communication to the CPS should be secured in transport (e.g., with TLS) or the CPRs should be digitally signed by the caller and verified by the callee (Section 5.2. For privacy and robustness reasons, both are preferable. In particular, if only transport security is used, then a compromised CPS can forge call origination information.

The entire system depends on the security of the authentication infrastructure. If the authentication credentials for a given number are compromised, then an attacker can impersonate calls from that number.

4.3.1. Substitution Attacks

All that receipt of the CPR proves is that Alice is trying to call Bob (or at least was as of very recently). It does not prove that this particular incoming call is from Alice. Consider the scenario in which we have a service which provides an automatic callback to a user-provided number. In that case, the attacker can arrange for a false caller-id value, as shown below:

Attacker	Callback Service	CPS	Bob

Place call to Bob ----->			
	Store CPR for CS:Bob ----->		
Call from CS (forged caller-id info) ----->			
	Call from CS -----> X		
			<----- Retrieve CPR for CS:Bob
	CPR for CS:Bob ----->		
			[Ring phone with callerid = CS]

In order to mount this attack, the attacker contacts the Callback Service (CS) and provides it with Bob's number. This causes the CS to initiate a call to Bob. As before, the CS contacts the CPS to insert an appropriate CPR and then initiates a call to Bob. Because it is a valid CS injecting the CPR, none of the security checks mentioned above help. However, the attacker simultaneously initiates a call to Bob using forged caller-id information corresponding to the

CS. If he wins the race with the CS, then Bob's phone will attempt to verify the attacker's call (and succeed since they are indistinguishable) and the CS's call will go to busy/voice mail/call waiting. Note: in a SIP environment, the callee might notice that there were multiple INVITES and thus detect this attack.

5. Some Potential Enhancements

Section 4 provides a broad sketch of an approach. In this section, we consider some potential enhancements. Readers can feel free to skip this section, as it is not necessary to get the flavor of the document.

5.1. Encrypted CPRs

In the system described in Section 4, the CPS learns the CPRs for every call, which is undesirable from a privacy perspective. The situation can be improved by having the caller store encrypted CPRs. A number of schemes are possible, but for concreteness we sketch one possibility.

The general idea is that each user's credentials are not just suitable for authentication to the CPS but also are an asymmetric key pair suitable for use in an encryption mode. When Alice wants to store a CPR for Bob she retrieves Bob's credentials (see Section 5.3) and then encrypts the CPR under Bob's public key. [The encryption needs to be done in such a way that if you don't have Bob's key, the message is indistinguishable from random. This is straightforward, but not compatible with typical secure message formats, which tend to indicate the recipient's identity.] The CPR is then stored with the CPS under Alice's identity. When Bob receives a call, he just asks the CPR (anonymously) for any calls from Alice to anyone. He then trial-decrypts each and if any of them is for him, he proceeds as before. In this way, the CPS learns Alice's call velocity but not who she is calling.

5.2. Signed CPRs

In the system described in Section 4, the CPS can forge CPRs. This threat can be removed by having the CPR signed by the originator along with a timestamp. If such a signature is required, the originator cannot make bogus calls appear to be valid but can still make valid calls appear to be bogus by removing the relevant CPRs.

5.3. Credential Lookup

In order to encrypt the CPR, the caller needs access to the callee's credentials (specifically the public key). This requires some sort of directory/lookup system. This document does not specify any particular scheme, but a list of requirements would be something like:

Obviously, if there is a single central database and the caller and callee each contact it in real time to determine the other's credentials, then this represents a real privacy risk, as the central database learns about each call. A number of mechanisms are potentially available to mitigate this:

- o Have endpoints pre-fetch credentials for potential counterparties (e.g., their address book or the entire database).
- o Have caching servers in the user's network that proxy their fetches and thus conceal the relationship between the user and the credentials they are fetching.

Clearly, there is a privacy/timeliness tradeoff in that getting really up-to-date knowledge about credential validity requires contacting the credential directory in real-time (e.g., via OCSP). This is somewhat mitigated for the caller's credentials in that he can get short-term credentials right before placing a call which only reveals his calling rate, but not who he is calling. Alternately, the CPS can verify the caller's credentials via OCSP, though of course this requires the callee to trust the CPS's verification. This approach does not work as well for the callee's credentials, but the risk there is more modest since an attacker would need to both have the callee's credentials and regularly poll the database for every potential caller.

We consider the exact best point in the tradeoff space to be an open issue.

5.4. Federated Verification Services

The discussion above is written in terms of a single CPS, but this potentially has scaling problems, as well as allowing the CPS to learn about every call. These issues can be alleviated by having a federated CPS. If a credential lookup service is already available, the CPS location can also be stored in the callee's credentials.

5.5. Escalation to VoIP

If the call is to be carried over the PSTN, then the security properties described above are about the best we can do. However, if

Alice and Bob are both VoIP capable, then there is an opportunity to provide a higher quality of service and security. The basic idea is that the CPR contains rendezvous information for Alice (e.g., Alice's SIP URI). Once Bob has verified Alice's CPR, he can initiate a VoIP connection directly to Alice, thus bypassing the PSTN. Mechanisms of this type are out of scope of this document.

6. Security Considerations

This entire document is about security, but the detailed security properties depend on having a single concrete scheme to analyze.

Appendix A. Acknowledgements

Jon Peterson provided some of the text in this document. The ideas in this document come out of discussions with Richard Barnes, Cullen Jennings, and Jon Peterson.

Author's Address

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

Phone: +1 650 678 2350
Email: ekr@rtfm.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: December 16, 2017

E. Rescorla
Mozilla
J. Peterson
Neustar
June 14, 2017

STIR Out of Band Architecture and Use Cases
draft-rescorla-stir-fallback-02.txt

Abstract

The PASSporT format defines a token that can be carried by signaling protocols, including SIP, to cryptographically attest the identity of callers. Not all telephone calls use Internet signaling protocols, however, and some calls use them for only part of their signaling path. This document describes use cases that require the delivery of PASSporT objects outside of the signaling path, and defines architectures and semantics to provide this functionality.

Status of This Memo

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

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

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

This Internet-Draft will expire on December 16, 2017.

Copyright Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must

include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Terminology	4
3. Operating Environments	4
4. Dataflows	5
5. Use Cases	6
5.1. Case 1: VoIP to PSTN Call	6
5.2. Case 2: Two Smart PSTN endpoints	6
5.3. Case 3: PSTN to VoIP Call	7
5.4. Case 4: Gateway Out-of-band	7
6. Authorization for Storing and Retrieving PASSporTs	8
6.1. Storage	8
6.2. Retrieval	9
6.2.1. Authentication	10
6.2.2. Encryption	10
7. Solution Architecture	12
7.1. Credentials and Phone Numbers	12
7.2. Solution Architecture	12
7.3. Security Analysis	13
7.4. Substitution Attacks	13
8. Call Placement Service Discovery	14
9. To Do	15
9.1. Credential Lookup	16
10. Acknowledgments	16
11. IANA Considerations	16
12. Security Considerations	17
13. Informative References	17
Authors' Addresses	18

1. Introduction

The STIR problem statement [RFC7340] describes widespread problems enabled by impersonation in the telephone network, including illegal robocalling, voicemail hacking, and swatting. As telephone services are increasingly migrating onto the Internet, and using Voice over IP (VoIP) protocols such as SIP [RFC3261], it is necessary for these protocols to support stronger identity mechanisms to prevent impersonation. For example, [I-D.ietf-stir-rfc4474bis] defines an Identity header of SIP requests capable of carrying a PASSporT [I-D.ietf-stir-passport] object in SIP as a means to cryptographically attest that the originator of a telephone call is authorized to use the calling party number (or, for native SIP cases, SIP URI) associated with the originator of the call. of the request.

Not all telephone calls use SIP today, however; and even those that do use SIP do not always carry SIP signaling end-to-end. Most calls from telephone numbers still traverse the Public Switched Telephone Network (PSTN) at some point. Broadly, calls fall into one of three categories:

1. One or both of the endpoints is actually a PSTN endpoint.
2. Both of the endpoints are non-PSTN (SIP, Jingle, ...) but the call transits the PSTN at some point.
3. Non-PSTN calls which do not transit the PSTN at all (such as native SIP end-to-end calls).

The first two categories represent the majority of telephone calls associated with problems like illegal robocalling: many robocalls today originate on the Internet but terminate at PSTN endpoints. However, the core network elements that operate the PSTN are legacy devices that are unlikely to be upgradable at this point to support an in-band authentication system. As such, those devices largely cannot be modified to pass signatures originating on the Internet--or indeed any inband signaling data--intact. Even if fields for tunneling arbitrary data can be found in traditional PSTN signaling, in some cases legacy elements would strip the signatures from those fields; in others, they might damage them to the point where they cannot be verified. For those first two categories above, any in-band authentication scheme does not seem practical in the current environment.

But while the core network of the PSTN remains fixed, the endpoints of the telephone network are becoming increasingly programmable and sophisticated. Landline "plain old telephone service" deployments, especially in the developed world, are shrinking, and increasingly being replaced by three classes of intelligent devices: smart phones, IP PBXs, and terminal adapters. All three are general purpose computers, and typically all three have Internet access as well as access to the PSTN. Additionally, various kinds of gateways increasingly front for legacy equipment. All of this provides a potential avenue for building an authentication system that implements stronger identity while leaving PSTN systems intact.

This capability also provides an ideal transitional technology while in-band STIR adoption is ramping up. It permits early adopters to use the technology even when intervening network elements are not yet STIR-aware, and through various kinds of gateways it may allow providers with a significant PSTN investment to still secure their calls with STIR.

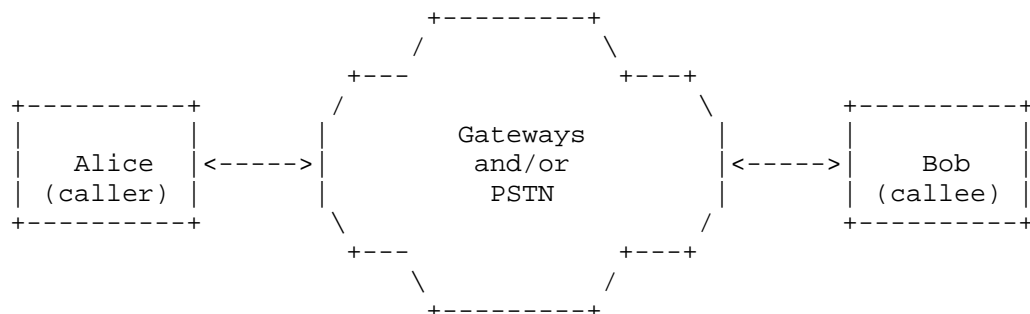
This specification therefore builds on the PASSport [I-D.ietf-stir-passport] mechanism and the work of [I-D.ietf-stir-rfc4474bis] to define a way that a PASSport object created in the originating network of a call can reach the terminating network even when it cannot be carried end-to-end in-band in the call signaling. This relies on a new service defined in this document that permits the PASSport object to be stored during call processing and retrieved for verification purposes.

2. Terminology

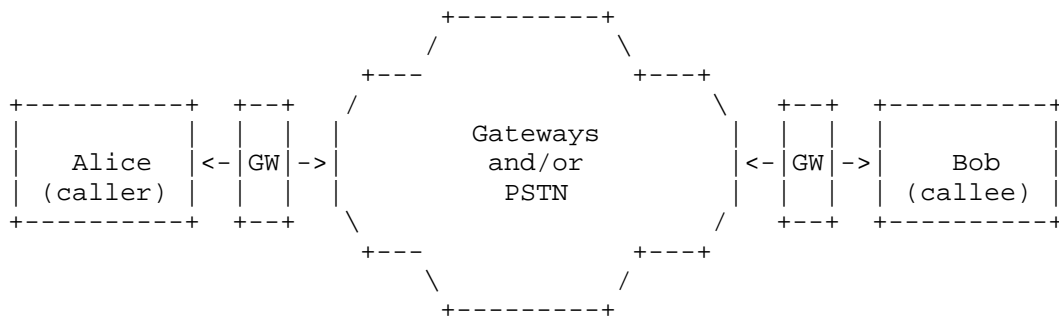
The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

3. Operating Environments

This section describes the environments in which the proposed mechanism is intended to operate. In the simplest setting, Alice is calling Bob through some set of gateways and/or the PSTN. Both Alice and Bob have smart devices which can be modified, but they do not have a clear connection between them: Alice cannot inject any data into signaling which Bob can read, with the exception of the asserted destination and origination E.164 numbers. The calling party number might originate from her own device or from the network. These numbers are effectively the only data that can be used for coordination between the endpoints.



In a more complicated setting, Alice and/or Bob may not have a smart or programmable device, but one or both of them are behind a STIR-aware gateway that can participate in out-of-band coordination, as shown below:



In such a case, Alice might have an analog connection to her gateway/switch which is responsible for her identity. Similarly, the gateway would verify Alice's identity, generate the right calling party number information and provide that number to Bob using ordinary POTS mechanisms.

4. Dataflows

Because in these operating environments endpoints cannot pass cryptographic information to one another directly through signaling, any solution must involve some rendezvous mechanism to allow endpoints to communicate. We call this rendezvous service a "call placement service" (CPS), a service where a record of call placement, in this case a PASSporT, can be stored for future retrieval. In principle this service could communicate any information, but minimally we expect it to include a full-form PASSporT that attests the caller, callee, and the time of the call. The callee can use the existence of a PASSporT for a given incoming call as rough validation of the asserted origin of that call. (See Section 9.1 for limitations of this design.)

There are roughly two plausible dataflow architectures for the CPS:

The callee registers with the CPS. When the caller wishes to place a call to the callee, it sends the PASSporT to the CPS, which immediately forwards it to the callee.

The caller stores the PASSporT with the CPS at the time of call placement. When the callee receives the call, it contacts the CPS and retrieves the PASSporT.

While the first architecture is roughly isomorphic to current VoIP protocols, it shares their drawbacks. Specifically, the callee must maintain a full-time connection to the CPS to serve as a notification channel. This comes with the usual networking costs to the callee and is especially problematic for mobile endpoints. Indeed, if the

endpoints had the capabilities to implement such an architecture, they could surely just use SIP or some other protocol to set up a secure session; even if the media were going through the traditional PSTN, a "shadow" SIP session could convey the PASSporT. Thus, we focus on the second architecture in which the PSTN incoming call serves as the notification channel and the callee can then contact the CPS to retrieve the PASSporT.

5. Use Cases

The following are the motivating use cases for this mechanism. Bear in mind that just as in [I-D.ietf-stir-rfc4474bis] there may be multiple Identity headers in a single SIP INVITE, so there may be multiple PASSporTs in this out-of-band mechanism associated with a single call. For example, a SIP user agent might create a PASSporT for a call with an end user credential, and as the call exits the originating administrative domain the network authentication service might create its own PASSporT for the same call. As such, these use cases may overlap in the processing of a single call.

5.1. Case 1: VoIP to PSTN Call

A call originates in the SIP world in a STIR-aware administrative domain. The local authentication service for that administrative domain creates a PASSporT which is carried in band in the call per [I-D.ietf-stir-rfc4474bis]. The call is routed out of the originating administrative domain and reaches a gateway to the PSTN. Eventually, the call will terminate on a mobile smartphone that supports this out-of-band mechanism.

In this use case, the originating authentication service can store the PASSporT with the appropriate CPS for the target telephone number as a fallback in case SIP signaling will not reach end-to-end. When the destination mobile smartphone receives the call over the PSTN, it consults the CPS and discovers a PASSporT from the originating telephone number waiting for it. It uses this PASSporT to verify the calling party number.

5.2. Case 2: Two Smart PSTN endpoints

A call originates with an enterprise PBX that has both Internet access and a built-in gateway to the PSTN. It will immediately drop its call to the PSTN, but before it does, it provisions a PASSporT on the CPS associated with the target telephone number.

After normal PSTN routing, the call lands on a smart mobile handset that supports the STIR out-of-band mechanism. It queries the appropriate CPS over the Internet to determine if a call has been

placed to it by a STIR-aware device. It finds the PASSporT provisioned by the enterprise PBX and uses it to verify the calling party number.

5.3. Case 3: PSTN to VoIP Call

A call originates with an enterprise PBX that has both Internet access and a built-in gateway to the PSTN. It will immediately drop the call to the PSTN, but before it does, it provisions a PASSporT with the CPS associated with the target telephone number. However, it turns out that the call will eventually route through the PSTN to an Internet gateway, which will translate this into a SIP call and deliver it to an administrative domain with a STIR verification service.

In this case, there are two subcases for how the PASSporT might be retrieved. In subcase 1, the Internet gateway that receives the call from the PSTN could query the appropriate CPS to determine if the original caller created and provisioned a PASSporT for this call. If so, it can retrieve the PASSporT and, when it creates a SIP INVITE for this call, add a corresponding Identity header per [I-D.ietf-stir-rfc4474bis]. When the SIP INVITE reaches the destination administrative domain, it will be able to verify the PASSporT normally. Note that to avoid discrepancies with the Date header field value, only full-form PASSporT should be used for this purpose. In subcase 2, the gateway does not retrieve the PASSporT itself, but instead the verification service at the destination administrative domain does so. Subcase 1 would perhaps be valuable for deployments where the destination administrative domain supports in-band STIR but not out-of-band STIR.

5.4. Case 4: Gateway Out-of-band

A call originates in the SIP world in a STIR-aware administrative domain. The local authentication service for that administrative domain creates a PASSporT which is carried in band in the call per [I-D.ietf-stir-rfc4474bis]. The call is routed out of the originating administrative domain and eventually reaches a gateway to the PSTN.

In this case, the originating authentication service does not support the out-of-band mechanism, so instead the gateway to the PSTN extracts the PASSporT from the SIP request and provisions it to the CPS. (When the call reaches the gateway to the PSTN, the gateway might first check the CPS to see if a PASSporT object had already been provisioned for this call, and only provision a PASSporT if none is present).

Ultimately, the call may terminate on the PSTN, or be routed back to the IP world. In the former case, perhaps the destination endpoints queries the CPS to retrieve the PASSport provisioned by the first gateway. Or if the call ultimately returns to the IP world, it might be the gateway from the PSTN back to the Internet that retrieves the PASSport from the CPS and attaches it to the new SIP INVITE it creates, or it might be the terminating administrative domain's verification service that checks the CPS when an INVITE arrives with no Identity header field. Either way the PASSport can survive the gap in SIP coverage caused by the PSTN leg of the call.

6. Authorization for Storing and Retrieving PASSports

The use cases show a variety of entities accessing the CPS to store and retrieve PASSports. The question of how the CPS authorizes the storage and retrieval of PASSport is thus a key design decision in the architecture.

The STIR architecture assumes that service providers and in some cases end user devices will have credentials suitable for attesting authority over telephone numbers per [I-D.ietf-stir-certificates]. These credentials provide the most obvious way that a CPS can authorize the storage and retrieval of PASSports. However, as use cases 3 and 4 in Section 5 show, it may sometimes make sense for the entity storing or retrieving PASSports to be an intermediary rather than a device associated with either the originating or terminating side of a call, and those intermediaries often would not have access to STIR credentials covering the telephone numbers in question.

It is an explicit design goal of this mechanism to minimize the potential privacy exposure of using a CPS. Ideally, the out-of-band mechanism should not result in a worse privacy situation than in-band [I-D.ietf-stir-rfc4474bis] STIR: for in-band, we might say that a SIP entity is authorized to receive a PASSport if it is an intermediate or final target of the routing of a SIP request. As the originator of a call cannot necessarily predict the routing path a call will follow, an out-of-band mechanism could conceivably even improve on the privacy story. As a first step, transport-level security can provide confidentiality from eavesdroppers for both the storage and retrieval of PASSports.

6.1. Storage

For authorizing the storage of PASSports, the architecture can permit some flexibility. A CPS could adopt a policy where it will store any valid PASSport - that is, the CPS could act as a limited verification service and validate the PASSport, only storing it if the timestamp and signature are valid. In that case, it would not matter whether

the CPS received a PASSporT from the authentication service that created it or from an intermediary gateway downstream in the routing path as in case 4: so long as the PASSporT is valid, it would be stored.

6.2. Retrieval

For retrieval of PASSporTs, the story is a bit more complicated. Beyond using transport-level security when storing and retrieving PASSporTs, the architecture must include some way to constrain access to the PASSporTs stored at a CPS. How those constraints should operate depends on the semantics of the request used to retrieve PASSporTs. A retrieval request could have one of the following three semantics:

- a) Are there any current PASSporTs for calls originating from 1.111.111.1111?
- b) Are there any current PASSporTs for calls destined to 2.222.222.2222?
- c) Are there any current PASSporTs for calls originating from 1.111.111.1111 and destined to 2.222.222.2222?

Each of these three semantics results in very different properties for the architecture. If a CPS permitted just anyone to ask for all PASSporTs that happen to exist for current calls to or from a given telephone number, that would be an unacceptable privacy situation. Although on the surface semantic (c) may seem sufficiently strict, a particular adversary might only be interested in learning when one specific party calls another, and there are certainly cases in which that could pose a significant security risk. While a CPS could eventually refuse to answer repeated requests from a single device that is obviously polling to collect the state of calls in progress, more sophisticated adversaries could outwit any attempt to do source filtering on requests at the CPS.

The semantics of (a) or (b) vs. (c) could be very significant when the originating and destination numbers are for call centers or similar organizations that send or receive a vast amount of calls for a single number. In a case where many thousands of people are trying to call a number where tickets have just gone on sale, for example, it might be difficult using semantics (b) to sift through all of the call setup attempts in progress to find a PASSporT matching any particular call. A more narrow semantic like (c) would make it far easier.

Sometimes the more narrow semantics of (c) can pose an obstacle to acquiring the right PASSporT, for example in call forwarding cases where retargeting of the request has occurred. Even using semantic (b) would be problematic if the PASSporT stored by the originating authentication service had a different original "dest". Mechanisms have been proposed for STIR to patch this by creating PASSporTs that record the diversion (see [I-D.peterson-passport-divert]), and potentially a CPS could store these additional PASSporT objects and supply them through the retrieval interface.

If we assume that the party retrieving PASSporTs from the CPS has a STIR credential attesting authority over the terminating number, then two more attractive mechanisms become possible: using authentication and encryption. Note however that in some use cases, like case 3 subcase 1 above, the retrieving party is an intermediary who would not have access to the necessary credentials. However, this might argue that subcase 1 should be disallowed for security reasons, and only subcase 2 should be permitted.

6.2.1. Authentication

For any of the three proposed retrieval semantics, a CPS could authenticate a request to retrieve PASSporTs and only release PASSporTs that have a destination that matches the credential provided by the requestor. Per semantic (b), if a smart endpoint has a credential for 2.222.222.2222, it could send a request to the CPS signed with that credential to retrieve any PASSporTs for calls in progress to 2.222.222.2222. In this case, (a) and (c) have very similar semantics: when the requestor asks for (a), effectively they would receive only those PASSporTs coming from 1.111.111.1111 that are destined to 2.222.222.2222 - though perhaps in cases where the call had been forwarded, a CPS aware of the situation could understand that the new destination should be authorized to see the original PASSporT.

On balance, an approach along the lines of requiring authenticating requests with semantic (a) appears attractive as a direction for out-of-band.

6.2.2. Encryption

Some of the privacy risks on the retrieval side could potentially be mitigated with encryption. If all PASSporTs stored at a CPS were encrypted with a key belonging to the intended destination, then potentially the CPS could allow almost anyone to download PASSporTs using semantics (a) or (b) without much fear of compromising private information about calls in progress - provided that the CPS always provided at least one encrypted blob in response to a request, even

if there was no call in progress. It would also prevent the CPS itself from learning the contents of PASSporTs, and thus metadata about calls in progress, which would make the CPS a less attractive target for pervasive monitoring (see [RFC7258]). However, encrypting PASSporTs faces some substantial difficulties.

First, this requires the entity that stores the PASSporT to have access to a public key associated with the intended called party to be used to encrypt the PASSporT. Discovering this key would require some new service that does not exist today; depending on how the CPS is architected, however, some kind of key store or repository could be implemented adjacent to it, and perhaps even incorporated into its operation. This key discovery problem is compounded by the fact that there can potentially be multiple entities that have authority over a telephone number: a carrier, a reseller, an enterprise, and an end user might all have credentials permitting them to attest that they are allowed to originate calls from a number, say. PASSporTs might need to be encrypted with multiple keys in the hopes that one will be decipherable by the relying party.

Second, in call forwarding cases, the difficulties in managing the relationship between PASSporTs with the diversion extension [I-D.peterson-passport-divert] become more serious. The originating authentication service would encrypt the PASSporT with the public key of the intended destination, but when a call is forwarded, it may go to a destination that does not possess the corresponding private key. It would require special behavior on the part of the retargeting entity, and probably the CPS as well, to accommodate encrypted PASSporTs that show a secure chain of diversion.

Another side effect of encrypting PASSporTs before storing them is that the CPS can no longer validate the PASSporTs since it cannot in fact read them. However, a CPS needs to know enough about PASSporTs so that it can respond to requests to retrieve them, whichever semantics are used - which means the CPS will always process some amount of metadata (even if some sort of hash function is used to index PASSporTs). Unless the storer of PASSporTs is authenticated, it may be possible for attackers to inject bogus PASSporTs into the system. Note however that merely injecting a bogus PASSporT into a CPS will not allow attackers to impersonate parties. That is because verification services trust a PASSporT based its own internal signature, not based on where the verification service found it. This is orthogonal to the current question of how the CPS authorizes an endpoint to acquire a PASSporT; though of course spamming a CPS with large numbers of bogus PASSporTs could cause a denial of service or similar problems with retrieval of PASSporTs.

7. Solution Architecture

In this section, we discuss a strawman architecture for providing the service described in the previous sections. This discussion is deliberately sketchy, focusing on broad concepts and skipping over details. The intent here is merely to provide a rough concept, not a complete solution.

7.1. Credentials and Phone Numbers

We start from the premise of the STIR problem statement [RFC7340] that phone numbers can be associated with credentials which can be used to attest ownership of numbers. For purposes of exposition, we will assume that ownership is associated with the endpoint (e.g., a smartphone) but it might well be associated with a provider or gateway acting for the endpoint instead. It might be the case that multiple entities are able to act for a given number, provided that they have the appropriate authority. [I-D.ietf-stir-certificates] describes a credentials system suitable for this purpose; the question of how an entity is determined to have control of a given number is out of scope for the current document.

7.2. Solution Architecture

An overview of the basic calling and verification process is shown below. In this diagram, we assume that Alice has the number +1.111.111.1111 and Bob has the number +2.222.222.2222.

Alice	Call Placement Service	Bob
Store PASSporT ----->		
Call from 1.111.111.1111 ----->		
	<- Authenticate as 1.222.222.2222 ---->	
	<----- Retrieve call record from 1.111.111.1111?	
	(1.222.222.2222,1.111.111.1111) -->	
	[Ring phone with callerid = 1.111.111.1111]	

When Alice wishes to make a call to Bob, she contacts the CPS and stores a PASSporT on the CPS. The CPS validates the PASSporT before

indexing it so that it can be acquired with a request from Bob's number

Once Alice has stored the PASSporT, she then places the call to Bob as usual. At this point, Bob's phone would usually ring and display Alice's number (+1.111.111.1111), which is informed by the existing PSTN mechanisms for relying a calling party number (i.e., the CIN field of the IAM). Instead, Bob's phone transparently contacts the CPS, authenticates itself, and requests any current PASSporTs for calls from Alice. The CPS responds with any such PASSporTs (assuming they exist). If such a PASSporT exists, and the verification service in Bob's phone validates it, then Bob's phone can then present the calling party number information as valid. Otherwise, the call is unverifiable. Note that this does not necessarily mean that the call is bogus; because we expect incremental deployment many legitimate calls will be unverifiable.

7.3. Security Analysis

The primary attack we seek to prevent is an attacker convincing the callee that a given call is from some other caller C. There are two scenarios to be concerned with:

The attacker wishes to impersonate a target when no call from that target is in progress.

The attacker wishes to substitute himself for an existing call setup as described in Section 7.4.

If an attacker can inject fake PASSporT into the CPS or in the communication from the CPS to the callee, he can mount either attack. As PASSporTs should be digitally signed by an appropriate authority for the number and verified by the callee (see Section 7.1), this should not arise in ordinary operations. For privacy and robustness reasons, using TLS on the originating side when storing the PASSporT at the CPS is recommended.

The entire system depends on the security of the credential infrastructure. If the authentication credentials for a given number are compromised, then an attacker can impersonate calls from that number. However, that is no different from in-band [I-D.ietf-stir-rfc4474bis] STIR.

7.4. Substitution Attacks

All that receipt of the PASSporT from the CPS proves to the called party is that Alice is trying to call Bob (or at least was as of very recently) - it does not prove that any particular incoming call is

from Alice. Consider the scenario in which we have a service which provides an automatic callback to a user-provided number. In that case, the attacker can try to arrange for a false caller-id value, as shown below:

Attacker	Callback Service	CPS	Bob

Place call to Bob ----->			
	Store PASSport for CS:Bob ----->		
Call from CS (forged caller-id info) ----->			
	Call from CS -----> X		
			<----- Retrieve PASSport for CS:Bob
	PASSport for CS:Bob ----->		
			[Ring phone with callerid = CS]

In order to mount this attack, the attacker contacts the Callback Service (CS) and provides it with Bob's number. This causes the CS to initiate a call to Bob. As before, the CS contacts the CPS to insert an appropriate PASSport and then initiates a call to Bob. Because it is a valid CS injecting the PASSport, none of the security checks mentioned above help. However, the attacker simultaneously initiates a call to Bob using forged caller-id information corresponding to the CS. If he wins the race with the CS, then Bob's phone will attempt to verify the attacker's call (and succeed since they are indistinguishable) and the CS's call will go to busy/voice mail/call waiting. Note: in a SIP environment, the callee might notice that there were multiple INVITEs and thus detect this attack.

8. Call Placement Service Discovery

In order for the two ends of the out-of-band dataflow to coordinate, they must agree on a way to discover a CPS and retrieve PASSport objects from it based solely on the rendezvous information available: the calling party number and the called number. There are a number of potential service discovery mechanisms that could be used for this purpose. The means of service discovery may vary by use case.

Although the discussion above is written in terms of a single CPS, having a significant fraction of all telephone calls result in

storing and retrieving PASSporTs at a single monolithic CPS has obvious scaling problems, and would as well allow the CPS to gather metadata about a very wide set of callers and callees. These issues can be alleviated by operational models with a federated CPS; any service discovery mechanism for out-of-band STIR should enable federation of the CPS function.

Some service discovery possibilities under consideration include the following:

If a credential lookup service is already available, the CPS location can also be recorded in the callee's credentials; an extension to [I-D.ietf-stir-certificates] could for example provide a link to the location of the CPS where PASSporTs should be stored for a destination.

There exist a number of common directory systems that might be used to translate telephone numbers into the URIs of a CPS. ENUM [RFC6116] is commonly implemented, though no "golden root" central ENUM administration exists that could be easily reused today to help the endpoints discover a common CPS. Other protocols associated with queries for telephone numbers, such as the TerI [I-D.peterson-modern-teri] protocol, could also serve for this application.

Another possibility is to use a single distributed service for this function. VIPR [I-D.rosenberg-dispatch-vipr-overview] proposed a RELOAD [RFC6940] usage for telephone numbers to help direct calls to enterprises on the Internet. It would be possible to describe a similar RELOAD usage to identify the CPS where calls for a particular telephone number should be stored. One advantage that the STIR architecture has over VIPR is that it assumes a credential system that proves authority over telephone numbers; those credentials could be used to determine whether or not a CPS could legitimately claim to be the proper store for a given telephone number.

Future versions of this specification will identify suitable service discovery mechanisms for out-of-band STIR.

9. To Do

Section 4 provides a broad sketch of an approach. In this section, we consider some areas for additional work. Readers can feel free to skip this section, as it is not necessary to get the flavor of the document.

9.1. Credential Lookup

In order to encrypt a PASSporT (see Section 6.2.2), the caller needs access to the callee's credentials (specifically their public key). This requires some sort of directory/lookup system. This document does not specify any particular scheme, but a list of requirements would be something like:

Obviously, if there is a single central database and the caller and callee each contact it in real time to determine the other's credentials, then this represents a real privacy risk, as the central database learns about each call. A number of mechanisms are potentially available to mitigate this:

- Have endpoints pre-fetch credentials for potential counterparties (e.g., their address book or the entire database).

- Have caching servers in the user's network that proxy their fetches and thus conceal the relationship between the user and the credentials they are fetching.

Clearly, there is a privacy/timeliness tradeoff in that getting really up-to-date knowledge about credential validity requires contacting the credential directory in real-time (e.g., via OCSP). This is somewhat mitigated for the caller's credentials in that he can get short-term credentials right before placing a call which only reveals his calling rate, but not who he is calling. Alternately, the CPS can verify the caller's credentials via OCSP, though of course this requires the callee to trust the CPS's verification. This approach does not work as well for the callee's credentials, but the risk there is more modest since an attacker would need to both have the callee's credentials and regularly poll the database for every potential caller.

We consider the exact best point in the tradeoff space to be an open issue.

10. Acknowledgments

The ideas in this document come out of discussions with Richard Barnes and Cullen Jennings.

11. IANA Considerations

This memo includes no request to IANA.

12. Security Considerations

This entire document is about security, but the detailed security properties depend on having a single concrete scheme to analyze.

13. Informative References

[I-D.ietf-stir-certificates]

Peterson, J. and S. Turner, "Secure Telephone Identity Credentials: Certificates", draft-ietf-stir-certificates-14 (work in progress), May 2017.

[I-D.ietf-stir-passport]

Wendt, C. and J. Peterson, "Personal Assertion Token (PASSporT)", draft-ietf-stir-passport-11 (work in progress), February 2017.

[I-D.ietf-stir-rfc4474bis]

Peterson, J., Jennings, C., Rescorla, E., and C. Wendt, "Authenticated Identity Management in the Session Initiation Protocol (SIP)", draft-ietf-stir-rfc4474bis-16 (work in progress), February 2017.

[I-D.peterson-modern-teri]

Peterson, J., "An Architecture and Information Model for Telephone-Related Information (TeRI)", draft-peterson-modern-teri-02 (work in progress), October 2016.

[I-D.peterson-passport-divert]

Peterson, J., "PASSporT Extension for Diverted Calls", draft-peterson-passport-divert-01 (work in progress), June 2017.

[I-D.rosenberg-dispatch-vipr-overview]

Rosenberg, J., Jennings, C., and M. Petit-Huguenin, "Verification Involving PSTN Reachability: Requirements and Architecture Overview", draft-rosenberg-dispatch-vipr-overview-04 (work in progress), October 2010.

[RFC2119]

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

- [RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, DOI 10.17487/RFC3261, June 2002, <<http://www.rfc-editor.org/info/rfc3261>>.
- [RFC6116] Bradner, S., Conroy, L., and K. Fujiwara, "The E.164 to Uniform Resource Identifiers (URI) Dynamic Delegation Discovery System (DDDS) Application (ENUM)", RFC 6116, DOI 10.17487/RFC6116, March 2011, <<http://www.rfc-editor.org/info/rfc6116>>.
- [RFC6940] Jennings, C., Lowekamp, B., Ed., Rescorla, E., Baset, S., and H. Schulzrinne, "REsource LOcation And Discovery (RELOAD) Base Protocol", RFC 6940, DOI 10.17487/RFC6940, January 2014, <<http://www.rfc-editor.org/info/rfc6940>>.
- [RFC7258] Farrell, S. and H. Tschofenig, "Pervasive Monitoring Is an Attack", BCP 188, RFC 7258, DOI 10.17487/RFC7258, May 2014, <<http://www.rfc-editor.org/info/rfc7258>>.
- [RFC7340] Peterson, J., Schulzrinne, H., and H. Tschofenig, "Secure Telephone Identity Problem Statement and Requirements", RFC 7340, DOI 10.17487/RFC7340, September 2014, <<http://www.rfc-editor.org/info/rfc7340>>.

Authors' Addresses

Eric Rescorla
Mozilla

Email: ekr@rtfm.com

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

Email: jon.peterson@neustar.biz