

Network Working Group
Internet-Draft
Intended status: Informational
Expires: September 10, 2020

E. Rescorla
Mozilla
J. Peterson
Neustar
March 9, 2020

STIR Out-of-Band Architecture and Use Cases
draft-ietf-stir-oob-07

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, or cannot reliably deliver SIP header fields end-to-end. 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 <https://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 September 10, 2020.

Copyright Notice

Copyright (c) 2020 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 (<https://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	7
5.2. Case 2: Two Smart PSTN endpoints	7
5.3. Case 3: PSTN to VoIP Call	7
5.4. Case 4: Gateway Out-of-band	8
5.5. Case 5: Enterprise Call Center	9
6. Storing and Retrieving PASSporTs	9
6.1. Storage	10
6.2. Retrieval	11
7. Solution Architecture	12
7.1. Credentials and Phone Numbers	12
7.2. Call Flow	13
7.3. Security Analysis	13
7.4. Substitution Attacks	14
7.5. Rate Control for CPS Storage	16
8. Authentication and Verification Service Behavior for Out-of-Band	17
8.1. Authentication Service (AS)	17
8.2. Verification Service (VS)	18
8.3. Gateway Placement Services	19
9. Example HTTPS Interface to the CPS	20
10. CPS Discovery	21
11. Encryption Key Lookup	23
12. Acknowledgments	24
13. IANA Considerations	24
14. Privacy Considerations	24
15. Security Considerations	25
16. Informative References	26
Authors' Addresses	28

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, [RFC8224] defines a SIP Identity header field capable of carrying PASSporT [RFC8225] objects 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.

Not all telephone calls use SIP today, however, and even those that do use SIP do not always carry SIP signaling end-to-end. Calls from telephone numbers still routinely 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.

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; they may be used for residential, mobile, or enterprise telephone services. Additionally, various kinds of gateways increasingly front for deployments of legacy PBX and PSTN switches. 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.

The techniques described in this document therefore build on the PASSporT [RFC8225] mechanism and the work of [RFC8224] to describe 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 called a Call Placement Service (CPS) that permits the PASSporT object to be stored during call processing and retrieved for verification purposes.

Potential implementors should note that this document merely defines the operating environments in which this out-of-band STIR mechanism is intended to operate. It provides use cases, gives a broad description of the components and a potential solution architecture. Various environments may have their own security requirements: a public deployment of out-of-band STIR faces far greater challenges than a constrained intranetwork deployment. To flesh out the storage and retrieval of PASSporTs in the CPS within this context, this document includes a strawman protocol suitable for that purpose. Deploying this framework in any given environment would require additional specification outside the scope of the current document.

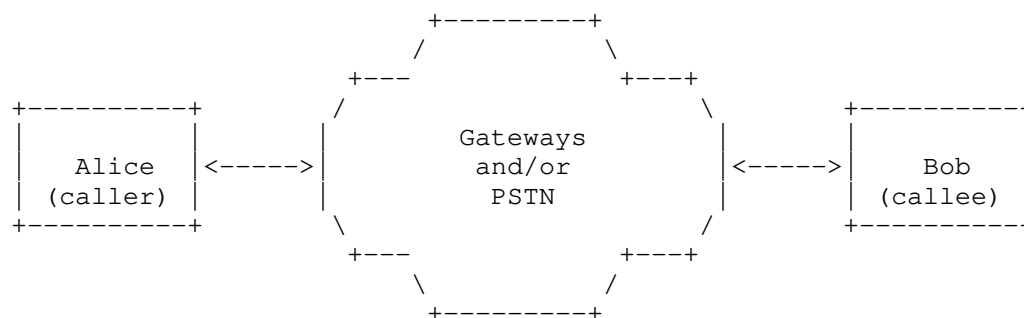
2. Terminology

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 BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

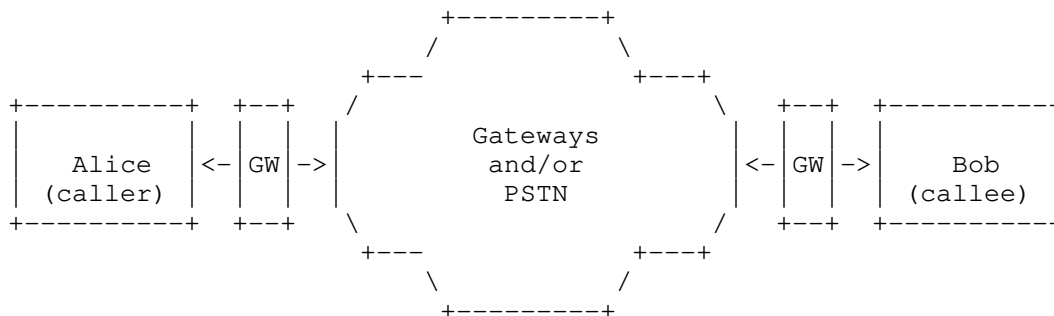
3. Operating Environments

This section describes the environments in which the proposed out-of-band STIR mechanism is intended to operate. In the simplest setting, Alice is calling Bob, and her call is routed through some set of gateways and/or the PSTN which do not support end-to-end delivery of STIR. Both Alice and Bob have smart devices which can access the Internet (perhaps enterprise devices, or even end user ones), but they do not have a clear telephone signaling 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 instead just a traditional telephone. However, 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 (e.g., PSTN) 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 Plain Ol' Telephone Service (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 11 for limitations of this design.)

This architecture does not mandate that any particular sort of entity operate a CPS, or mandate any means to discover a CPS. A CPS could be run internally within a network, or made publicly available. One or more CPSes could be run by a carrier, as repositories for PASSporTs for calls sent to its customers, or a CPS could be built-in to an enterprise PBX, or even a smartphone. To the degree possible, it is specified here generically, as an idea that may have applicability to a variety of STIR deployments.

There are roughly two plausible dataflow architectures for the CPS:

1. 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, or,
2. 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. In specialized environments, for example a call center that receives a large volume of incoming calls that originated in the PSTN, the notification channel approach might be viable.

5. Use Cases

The following are the motivating use cases for this mechanism. Bear in mind that just as in [RFC8224] 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 a SIP environment 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 [RFC8224]. 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 (per the practices of Section 10) 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, which communicates through traditional telephone signaling protocols. The PBX immediately routes the 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 route 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 field per [RFC8224]. 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 [RFC8224]. 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 a SIP environment. In the former case, perhaps the destination endpoint queries the CPS to retrieve the PASSporT provisioned by the first gateway. Or if the call ultimately returns to a SIP environment, 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.

5.5. Case 5: Enterprise Call Center

A call originates from a mobile user, and a STIR authentication service operated by their carrier creates a PASSporT for the call. As the carrier forwards the call via SIP, it attaches the PASSporT to the SIP call with an Identity header field. As a fallback in case the call will not go end-to-end over SIP, the carrier also stores the PASSporT in a CPS.

The call is then routed over SIP for a time, before it transitions to the PSTN and ultimately is handled by a legacy PBX at a high-volume call center. The call center supports the out-of-band service, and has a high-volume interface to a CPS to retrieve PASSporTs for incoming calls; agents at the call center use a general purpose computer to manage inbound calls and can receive STIR notifications through it. When the PASSporT arrives at the CPS, it is sent through a subscription/notification interface to a system that can correlate incoming calls with valid PASSporTs. The call center agent sees that a valid call from the originating number has arrived.

6. 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 [RFC8226]. These credentials provide the most obvious way that a CPS can authorize the storage and retrieval of PASSporTs. However, as use cases 3, 4 and 5 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. Requiring authorization based on a credential to store PASSporTs is therefore undesirable, though potentially acceptable if sufficient steps are taken to mitigate any privacy risk of leaking data.

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 [RFC8224] 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.

Broadly, the architecture recommended here thus is one focused on permitting any entity to store encrypted PASSporTs at the CPS, indexed under the called number. PASSporTs will be encrypted with a public key associated with the called number, so these PASSporTs may safely be retrieved by any entity, as only holders of the corresponding private key will be able to decrypt the PASSporT. This also prevents the CPS itself from learning the contents of PASSporTs, and thus metadata about calls in progress, which makes the CPS a less attractive target for pervasive monitoring (see [RFC7258]). As a first step, transport-level security can provide confidentiality from eavesdroppers for both the storing and retrieval of PASSporTs. To bolster the privacy story, prevent denial-of-service flooding of the CPS, and to complicate traffic analysis, a few additional mechanisms are also recommended below.

6.1. Storage

There are a few dimensions to authorizing the storage of PASSporTs. Encrypting PASSporTs prior to storage entails that a CPS has no way to tell if a PASSporT is valid; it simply conveys encrypted blocks that it cannot access itself, and can make no authorization decision based on the PASSporT contents. There is certainly no prospect for the CPS to verify the PASSporTs itself.

Note that this architecture requires clients that store PASSporTs to have access to an encryption key associated with the intended called party to be used to encrypt the PASSporT. Discovering this key requires the existence of a key lookup service (see Section 11); 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. Key discovery is made more complicated 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 for out-of-band use therefore might need to be encrypted with multiple keys in the hopes that one will be decipherable by the relying party.

Again, the most obvious way to authorize storage is to require the originator to authenticate themselves to the CPS with their STIR credential. However, since the call is indexed at the CPS under the called number, this can weaken the privacy story of the architecture, as it reveals to the CPS both the identity of the caller and the callee. Moreover, it does not work for the gateway use cases

described above; to support those use cases, we must effectively allow any entity to store PASSporTs at a CPS. This does not degrade the anti-impersonation security of STIR, because entities who do not possess the necessary credentials to sign the PASSporT will not be able to create PASSporTs that will be treated as valid by verifiers. In this architecture, it does 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 above. However, if literally anyone can store PASSporTs in the CPS, an attacker could easily flood the CPS with millions of bogus PASSporTs indexed under a calling number, and thereby prevent the called party from finding a valid PASSporT for an incoming call buried in a haystack of fake entries.

The solution architecture must therefore include some sort of traffic control system to prevent flooding. Preferably, this should not require authenticating the source, as this will reveal to the CPS both the source and destination of traffic. A potential solution is discussed below in Section 7.5.

6.2. Retrieval

For retrieval of PASSporTs, this architecture assumes that clients will contact the CPS through some sort of polling or notification interface to receive all current PASSporTs for calls destined to a particular telephone number, or block of numbers.

As PASSporTs stored at the CPS are encrypted with a key belonging to the intended destination, the CPS can safely allow anyone to download PASSporTs for a called number without much fear of compromising private information about calls in progress - provided that the CPS always returns at least one encrypted blob in response to a request, even if there was no call in progress. Otherwise, entities could poll the CPS constantly, or eavesdrop on traffic, to learn whether or not calls were in progress. The CPS MUST generate at least one unique and plausible encrypted response to all retrieval requests, and these dummy encrypted PASSporTs MUST NOT be repeated for later calls. An encryption scheme needs to be carefully chosen to make messages look indistinguishable from random when encrypted, so that information about called party is not discoverable from legitimate encrypted PASSporTs.

Because the entity placing a call may discover multiple keys associated with the called party number, multiple valid PASSporTs may be stored in the CPS. A particular called party who retrieves PASSporTs from the CPS may have access to only one of those keys. Thus, the presence of one or more PASSporTs that the called party cannot decrypt - which would be indistinguishable from the "dummy"

PASSporTs created by the CPS when no calls are in progress - does not entail that there is no call in progress. A retriever likely will need to decrypt all PASSporTs retrieved from the CPS, and may find only one that is valid.

In order to prevent the CPS from learning the numbers that a callee controls, callees might also request PASSporTs for numbers that they do not own, that they have no hope of decrypting. Implementations could even allow a callee to request PASSporTs for a range or prefix of numbers: a trade-off where that callee is willing to sift through bulk quantities of undecryptable PASSporTs for the sake of hiding from the CPS what numbers it controls.

Note that in out-of-band call forwarding cases, special behavior is required to manage the relationship between PASSporTs using the diversion extension [I-D.ietf-stir-passport-divert]. The originating authentication service would encrypt the initial PASSporT with the public encryption key of the intended destination, but once a call is forwarded, it may go to a destination that does not possess the corresponding private key and thus could not decrypt the original PASSporT. This requires the retargeting entity to generate encrypted PASSporTs that show a secure chain of diversion: a retargeting storer SHOULD use the "div-o" PASSporT type, with its "opt" extension, as specified in [I-D.ietf-stir-passport-divert] in order to nest the original PASSporT within the encrypted diversion PASSporT.

7. Solution Architecture

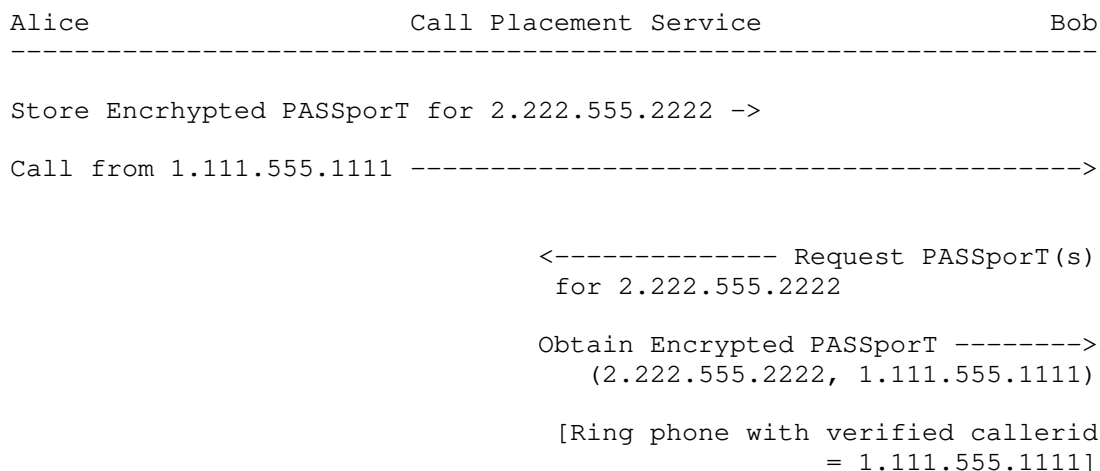
In this section, we discuss a high-level architecture for providing the service described in the previous sections. This discussion is deliberately sketchy, focusing on broad concepts and skipping over details. The intent here is merely to provide an overall architecture, not an implementable specification. A more concrete example of how this might be specified is given in Section 9.

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. [RFC8226] describes a credential 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. Call Flow

An overview of the basic calling and verification process is shown below. In this diagram, we assume that Alice has the number +1.111.555.1111 and Bob has the number +2.222.555.2222.



When Alice wishes to make a call to Bob, she contacts the CPS and stores an encrypted PASSporT on the CPS indexed under Bob's number. The CPS then awaits retrievals for that number.

When Alice places the call, Bob's phone would usually ring and display Alice's number (+1.111.555.1111), which is informed by the existing PSTN mechanisms for relaying a calling party number (e.g., the CIN field of the IAM). Instead, Bob's phone transparently contacts the CPS and requests any current PASSporTs for calls to his number. The CPS responds with any such PASSporTs (or dummy PASSporTs if no relevant ones are currently stored). If such a PASSporT exists, and the verification service in Bob's phone decrypts it using his private key, validates it, then Bob's phone can 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:

1. The attacker wishes to impersonate a target when no call from that target is in progress.
2. The attacker wishes to substitute himself for an existing call setup.

If an attacker can inject fake PASSporTs 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. Any attacker who is aware of calls in progress can attempt to mount a race to substitute themselves as described in Section 7.4. For privacy and robustness reasons, using TLS [RFC8446] 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 [RFC8224] STIR.

A secondary attack we must also prevent is denial-of-service against the CPS, which requires some form of rate control solution that will not degrade the privacy properties of the architecture.

7.4. Substitution Attacks

All the 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 ----->			
(from 111.555.1111)			
	Store PASSporT for		
	CS:Bob ----->		
Call from Attacker (forged CS caller-id info) ----->			
	Call from CS ----->		X
			<-- Retrieve PASSporT
			for CS:Bob
	PASSporT for CS:Bob ----->		
		[Ring phone with callerid =	
		111.555.1111]	

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.

In order to prevent a passive attacker from using traffic analysis or similar means to learn precisely when a call is placed, it is essential that the connection between the caller and the CPS be encrypted as recommended above. Authentication services could store dummy PASSporTs at the CPS at random intervals in order to make it more difficult for an eavesdropper to use traffic analysis to determine that a call was about to be placed.

Note that in a SIP environment, the callee might notice that there were multiple INVITEs and thus detect this attack, but in some PSTN interworking scenarios, or highly intermediated networks, only one call setup attempt will reach the target. Also note that the success of this substitution attack depends on the attacker landing their call within the narrow window that the PASSporT is retained in the CPS, so shortening that window will reduce the opportunity for the attack. Finally, smart endpoints could implement some sort of state

coordination to ensure that both sides believe the call is in progress, though methods of supporting that are outside the scope of this document.

7.5. Rate Control for CPS Storage

In order to prevent the flooding of a CPS with bogus PASSporTs, we propose the use of "blind signatures" (see [RFC5636]). A sender will initially authenticate to the CPS using its STIR credentials, and acquire a signed token from the CPS that will be presented later when storing a PASSporT. The flow looks as follows:

Sender	CPS
Authenticate to CPS ----->	
Blinded(K_temp) ----->	
<----- Sign(K_cps, Blinded(K_temp))	
[Disconnect]	
Sign(K_cps, K_temp)	
Sign(K_temp, E(K_receiver, PASSporT)) --->	

At an initial time when no call is yet in progress, a potential client connects to the CPS, authenticates, and sends a blinded version of a freshly generated public key. The CPS returns a signed version of that blinded key. The sender can then unblind the key and gets a signature on K_temp from the CPS.

Then later, when a client wants to store a PASSporT, it connects to the CPS anonymously (preferably over a network connection that cannot be correlated with the token acquisition) and sends both the signed K_temp and its own signature over the encrypted PASSporT. The CPS verifies both signatures and if they verify, stores the encrypted passport (discarding the signatures).

This design lets the CPS rate limit how many PASSporTs a given sender can store just by counting how many times K_temp appears; perhaps CPS policy might reject storage attempts and require acquisition of a new K_temp after storing more than a certain number of PASSporTs indexed under the same destination number in a short interval. This does not of course allow the CPS to tell when bogus data is being provisioned by an attacker, simply the rate at which data is being provisioned. Potentially, feedback mechanisms could be developed that would allow the called parties to tell the CPS when they are receiving unusual or bogus PASSporTs.

This architecture also assumes that the CPS will age out PASSports. A CPS SHOULD NOT keep any stored PASSport for no longer than a value that might be selected for the verification service policy for freshness of the "iat" value as described in [RFC8224] (i.e. sixty seconds). Any reduction in this window makes substitution attacks (see Section 7.4) harder to mount, but making the window too small might conceivably age PASSports out while a heavily redirected call is still alerting.

An alternative potential approach to blind signatures would be the use of oblivious pseudorandom functions (VOPRFs, per [I-D.privacy-pass]), which move prove faster.

8. Authentication and Verification Service Behavior for Out-of-Band

[RFC8224] defines an authentication service and a verification service as functions that act in the context of SIP requests and responses. This specification thus provides a more generic description of authentication service and verification service behavior that might or might not involve any SIP transactions, but depends only on placing a request for communications from an originating identity to one or more destination identities.

8.1. Authentication Service (AS)

Out-of-band authentication services perform steps similar to those defined in [RFC8224] with some exceptions:

Step 1: The authentication service MUST determine whether it is authoritative for the identity of the originator of the request, that is, the identity it will populate in the "orig" claim of the PASSport. It can do so only if it possesses the private key of one or more credentials that can be used to sign for that identity, be it a domain or a telephone number or some other identifier. For example, the authentication service could hold the private key associated with a STIR certificate [RFC8225].

Step 2: The authentication service MUST determine that the originator of communications can claim the originating identity. This is a policy decision made by the authentication service that depends on its relationship to the originator. For an out-of-band application built-in to the calling device, for example, this is the same check performed in Step 1: does the calling device hold a private key, one corresponding to a STIR certificate, that can sign for the originating identity?

Step 3: The authentication service MUST acquire the public encryption key of the destination, which will be used to encrypt the PASSport

(see Section 11). It MUST also discover (see Section 10) the CPS associated with the destination. The authentication service may already have the encryption key and destination CPS cached, or may need to query a service to acquire the key. Note that per Section 7.5 the authentication service may also need to acquire a token for PASSporT storage from the CPS upon CPS discovery. It is anticipated that the discovery mechanism (see Section 10) used to find the appropriate CPS will also find the proper key server for the public key of the destination. In some cases, a destination may have multiple public encryption keys associated with it. In that case, the authentication service MUST collect all of those keys.

Step 4: The authentication service MUST create the PASSporT object. This includes acquiring the system time to populate the "iat" claim, and populating the "orig" and "dest" claims as described in [RFC8225]. The authentication service MUST then encrypt the PASSporT. If in Step 3 the authentication service discovered multiple public keys for the destination, it MUST create one encrypted copy for each public key it discovered.

Finally, the authentication service stores the encrypted PASSporT(s) at the CPS discovered in Step 3. Only after that is completed should any call be initiated. Note that a call might be initiated over SIP, and the authentication service would place the same PASSporT in the Identity header field value of the SIP request - though SIP would carry a cleartext version rather than an encrypted version sent to the CPS. In that case, out-of-band would serve as a fallback mechanism in case the request was not conveyed over SIP end-to-end. Also, note that the authentication service MAY use a compact form of the PASSporT for a SIP request, whereas the version stored at the CPS MUST always be a full form PASSporT.

8.2. Verification Service (VS)

When a call arrives, an out-of-band verification service performs steps similar to those defined in [RFC8224] with some exceptions:

Step 1: The verification service contacts the CPS and requests all current PASSporTs for its destination number; or alternatively it may receive PASSporTs through a push interface from the CPS in some deployments. The verification service MUST then decrypt all PASSporTs using its private key. Some PASSporTs may not be decryptable for any number of reasons: they may be intended for a different verification service, or they may be "dummy" values inserted by the CPS for privacy purposes. The next few steps will narrow down the set of PASSporTs that the verification service will examine from that initial decryptable set.

Step 2: The verification service MUST determine if any "ppt" extensions in the PASSporTs are unsupported. It takes only the set of supported PASSporTs and applies the next step to them.

Step 3: The verification service MUST determine if there is an overlap between the calling party number presented in call signaling and the "orig" field of any decrypted PASSporTs. It takes the set of matching PASSporTs and applies the next step to them.

Step 4: The verification service MUST determine if the credentials that signed each PASSporT are valid, and if the verification service trusts the CA that issued the credentials. It takes the set of trusted PASSporTs to the next step.

Step 5: The verification service MUST check the freshness of the "iat" claim of each PASSporT. The exact interval of time that determines freshness is left to local policy. It takes the set of fresh PASSporTs to the next step.

Step 6: The verification service MUST check the validity of the signature over each PASSporT, as described in [RFC8225].

Finally, the verification service will end up with one or more valid PASSporTs corresponding to the call it has received. In keeping with baseline STIR, this document does not dictate any particular treatment of calls that have valid PASSporTs associated with them; the handling of the call after the verification process depends on how the verification service is implemented and on local policy. However, it is anticipated that local policies could involve making different forwarding decisions in intermediary implementations, or changing how the user is alerted or how identity is rendered in UA implementations.

8.3. Gateway Placement Services

The STIR out-of-band mechanism also supports the presence of gateway placement services, which do not create PASSporTs themselves, but instead take PASSporTs out of signaling protocols and store them at a CPS before gatewaying to a protocol that cannot carry PASSporTs itself. For example, a SIP gateway that sends calls to the PSTN could receive a call with an Identity header field, extract a PASSporT from the Identity header field, and store that PASSporT at a CPS.

To place a PASSporT at a CPS, a gateway MUST perform Step 3 of Section 8.1 above: that is, it must discover the CPS and public key associated with the destination of the call, and may need to acquire a PASSporT storage token (see Section 6.1). Per Step 3 of

Section 8.1 this may entail discovering several keys. The gateway then collects the in-band PASSporT(s) from the in-band signaling, encrypts the PASSporT(s), and stores them at the CPS.

A similar service could be performed by a gateway that retrieves PASSporTs from a CPS and inserts them into signaling protocols that support carrying PASSporTs in-band. This behavior may be defined by future specifications.

9. Example HTTPS Interface to the CPS

As a rough example, we show a Call Placement Service implementation here which uses a REST API to store and retrieve objects at the CPS. The calling party stores the PASSporT at the CPS prior to initiating the call; the PASSporT is stored at a location at the CPS that corresponds to the called number. Note that it is possible for multiple parties to be calling a number at the same time, and that for called numbers such as large call centers, many PASSporTs could legitimately be stored simultaneously, and it might prove difficult to correlate these with incoming calls.

Assume that an authentication service has created the following PASSporT for a call to the telephone number 2.222.555.2222 (note that these are dummy values):

```
eyJhbGciOiJFUzI1NiIsInR5cCI6InBhc3Nwb3J0IiwieDV1IjoiaHR0cHM6Ly9jZ
ZXJOmV4YW1wbGUub3JnL3Bhc3Nwb3J0LmN1ciJ9.eyJkZXN0Ijp7InRuIjpbIj
lYmJiNTUyMjIyIl19LCJpYXQiOiIxNTgzMjUxODEwIiwib3JpZyI6eyJ0biI6Ij
ExMTE1NTUxMTE1In19.pn1j4K1LlH0r4vID0u3CT1e9Hq4xLNgZUTv45Vbxml
d3IvYzUq4K0SA378vfp4x6twY0KTDlDvpsereS438ZHaO
```

Through some discovery mechanism (see Section 10), the authentication service discovers the network location of a web service that acts as the CPS for 2.222.555.2222. Through the same mechanism, we will say that it has also discovered one public encryption key for that destination. It uses that encryption key to encrypt the PASSport, resulting in the encrypted PASSport:

```
rlWuoTpvBvWSHmV1AvVfVaE5pPV6VaOup3Ajo3W0VvjvrQI1VwbvnUE0pUZ6Y19w
MKW0YzI4LJ1joThho3WaY3Oup3Ajo3W0YzAypvW9r1WxMKA0Vvc7VaIlnFV6JlWm
nKN6LJkcL2INMKuuokOgMF5wo20vKK0fVzyuqPv6VwR0AQZLzQtmAQHvYPWipzyaV
w7VaEhVwlvbZGVkAGH1AGRLZGVvskOed3cwGlubeEjnxRTWp1ZuJfHafuq0-mW6S1
IbtSJfWU0e8Dwcwyx-pcSLcSLfbswAPcGmB3DsCBypxTnF6uRpx7j
```

Having concluded the numbered steps in Section 8.1, including acquiring any token (per Section 6.1) needed to store the PASSporT at the CPS, the authentication service then stores the encrypted PASSporT:

```
POST /cps/2.222.555.2222/ppts HTTP/1.1
Host: cps.example.com
Content-Type: application/passport
```

```
rlWuoTpvBvWSHmVlAvVfVaE5pPV6VaOup3Ajo3W0VvjvrQIlVwbvnUE0pUZ6Yl9w
MKW0YzI4LJljoTHho3WaY3Oup3Ajo3W0YzAypvW9rlWxMKA0Vwc7VaIlNFV6JlWm
nKN6LJkcL2INMKuuOKOfMF5wo20vKK0fVzyuqPV6VwR0AQZlZQtmaQHvYPWipzyaV
wc7VaEhVwbvZGVkAGH1AGRlZGVvsK0ed3cwGlubEjnxRTwUPaJfjHafuq0-mW6S1
IBtSJFwU0e8Dwcwyx-pcSLcSLfbwAPcGmB3DsCBypxTnF6uRpx7j
```

The web service assigns a new location for this encrypted PASSporT in the collection, returning a 201 OK with the location of /cps/2.222.222.2222/ppts/ppt1. Now the authentication service can place the call, which may be signaled by various protocols. Once the call arrives at the terminating side, a verification service contacts its CPS to ask for the set of incoming calls for its telephone number (2.222.222.2222).

```
GET /cps/2.222.555.2222/ppts
Host: cps.example.com
```

This returns to the verification service a list of the PASSporTs currently in the collection, which currently consists of only /cps/2.222.222.2222/ppts/ppt1. The verification service then sends a new GET for /cps/2.222.555.2222/ppts/ppt1/ which yields:

```
HTTP/1.1 200 OK
Content-Type: application/passport
Link: <https://cps.example.com/cps/2.222.555.2222/ppts>
```

```
rlWuoTpvBvWSHmVlAvVfVaE5pPV6VaOup3Ajo3W0VvjvrQIlVwbvnUE0pUZ6Yl9w
MKW0YzI4LJljoTHho3WaY3Oup3Ajo3W0YzAypvW9rlWxMKA0Vwc7VaIlNFV6JlWm
nKN6LJkcL2INMKuuOKOfMF5wo20vKK0fVzyuqPV6VwR0AQZlZQtmaQHvYPWipzyaV
wc7VaEhVwbvZGVkAGH1AGRlZGVvsK0ed3cwGlubEjnxRTwUPaJfjHafuq0-mW6S1
IBtSJFwU0e8Dwcwyx-pcSLcSLfbwAPcGmB3DsCBypxTnF6uRpx7j
```

That concludes Step 1 of Section 8.2; the verification service then goes on to the next step, processing that PASSporT through its various checks. A complete protocol description for CPS interactions is left to future work.

10. CPS 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. Because the storage of PASSporTs in this architecture is indexed by the called party

number, it makes sense to discover a CPS based on the called party number as well. 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 largely 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. Likely models include ones where a carrier operates one or more CPS instances on behalf of its customers, enterprises run a CPS instance on behalf of their PBX users, or where third-party service providers offer a CPS as a cloud service.

Some service discovery possibilities under consideration include the following:

For some deployments in closed (e.g. intranetwork) environments, the CPS location can simply be provisioned in implementations, obviating the need for a discovery protocol.

If a credential lookup service is already available (see Section 11), the CPS location can also be recorded in the callee's credentials; an extension to [RFC8226] 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.ietf-modern-teri] protocol, could also serve for this application.

Another possibility is to use a single distributed service for this function. VIPR [I-D.jennings-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.

This document does not prescribe any single way to do service discovery for a CPS; it is envisioned that initial deployments will provision the location of the CPS at the Authentication Service and Verification Service.

11. Encryption Key Lookup

In order to encrypt a PASSporT (see Section 6.1), the caller needs access to the callee's public encryption key. Note that because STIR uses ECDSA for signing PASSporTs, the public key used to verify PASSporTs is not suitable for this function, and thus the encryption key must be discovered separately. This requires some sort of directory/lookup system.

Some initial STIR deployments have fielded certificate repositories so that verification services can acquire the signing credentials for PASSporTs, which are linked through a URI in the "x5u" element of the PASSporT. These certificate repositories could clearly be repurposed for allowing authentication services to download the public encryption key for the called party - provided they can be discovered by calling parties. This document does not specify any particular discovery scheme, but instead offers some general guidance about potential approaches.

It is a desirable property that the public encryption key for a given party be linked to their STIR credential. An ECDH [RFC7748] public-private key pair might be generated for a subcert [I-D.ietf-tls-subcerts] of the STIR credential. That subcert could be looked up along with the STIR credential of the called party. Further details of this subcert, and the exact lookup mechanism involved, are deferred for future protocol work.

Obviously, if there is a single central database that the caller and callee each access in real time to download the other's keys, then this represents a real privacy risk, as the central key database learns about each call. A number of mechanisms are potentially available to mitigate this:

- Have endpoints pre-fetch keys 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 keys they are fetching.

Clearly, there is a privacy/timeliness tradeoff in that getting up-to-date knowledge about credential validity requires contacting the credential directory in real-time (e.g., via OCSP [RFC2560]). 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.

12. Acknowledgments

The ideas in this document come out of discussions with Richard Barnes and Cullen Jennings. We'd also like to thank Russ Housley, Chris Wendt, Eric Burger, Mary Barnes, Ben Campbell, Ted Huang, Jonathan Rosenberg and Robert Sparks for helpful suggestions.

13. IANA Considerations

This memo includes no request to IANA.

14. Privacy Considerations

Delivering PASSporTs out-of-band offers a different set of privacy properties than traditional in-band STIR. In-band operations convey PASSporTs as headers in SIP messages in cleartext, which any forwarding intermediaries can potentially inspect. By contrast, out-of-band STIR stores these PASSporTs at a service after encrypting them as described in Section 6, effectively creating a path between the authentication and verification service in which the CPS is the sole intermediary, but the CPS cannot read the PASSporTs. Potentially, out-of-band PASSporT delivery could thus improve on the privacy story of STIR.

The principle actors in the operation of out-of-band are the AS, VS, and CPS. The AS and VS functions differ from baseline [RFC8224] behavior, in that they interact with an CPS over a non-SIP interface, of which the REST interface in Section 9 serves as an example. Some out-of-band deployments may also require a discovery service for the CPS itself (Section 10) and/or encryption keys (Section 11). Even with encrypted PASSporTs, the network interactions by which the AS and VS interact with the CPS, and to a lesser extent any discovery

services, thus create potential opportunities for data leakage about calling and called parties.

The process of storing and retrieving PASSporTs at a CPS can itself reveal information about calls being placed. The mechanism takes care not to require that the AS authenticate itself to the CPS, relying instead on a blind signature mechanism for flood control prevention. Section 7.4 discusses the practice of storing "dummy" PASSporTs at random intervals to thwart traffic analysis, and as Section 8.2 notes, a CPS is required to return a dummy PASSporT even if there is no PASSporT indexed for that calling number, which similarly enables the retrieval side to randomly request PASSporTs when there are no calls in progress. These measures can help to mitigate information disclosure in the system. In implementations that require service discovery (see Section 10), perhaps through key discovery (Section 11), similar measures could be used to make sure that service discovery does not itself disclose information about calls.

Ultimately, this document only provides a framework for future implementation of out-of-band systems, and the privacy properties of a given implementation will depend on architectural assumptions made in those environments. More closed systems for intranet operations may adopt a weaker security posture but otherwise mitigate the risks of information disclosure, where more open environment will require careful implementation of the practices described here.

For general privacy risks associated with the operations of STIR, also see the Privacy Considerations of [RFC8224].

15. Security Considerations

This entire document is about security, but the detailed security properties will vary depending on how the framework is applied and deployed. General guidance for dealing with the most obvious security challenges posed by this framework is given in Section 7.3 and Section 7.4, along proposed solutions for problems like denial-of-service attacks or traffic analysis against the CPS.

Although there are considerable security challenges associated with widespread deployment of a public CPS, those must be weighed against the potential usefulness of a service that delivers a STIR assurance without requiring the passage of end-to-end SIP. Ultimately, the security properties of this mechanism are at least comparable to in-band STIR: the substitution attack documented in Section 7.4 could be implemented by any in-band SIP intermediary or eavesdropper who happened to see the PASSporT in transit, say, and launch its own call

with a copy of that PASSport to race against the original to the destination.

16. Informative References

- [I-D.ietf-modern-teri]
Peterson, J., "An Architecture and Information Model for Telephone-Related Information (TeRI)", draft-ietf-modern-teri-00 (work in progress), July 2018.
- [I-D.ietf-stir-passport-divert]
Peterson, J., "PASSport Extension for Diverted Calls", draft-ietf-stir-passport-divert-07 (work in progress), November 2019.
- [I-D.ietf-tls-subcerts]
Barnes, R., Iyengar, S., Sullivan, N., and E. Rescorla, "Delegated Credentials for TLS", draft-ietf-tls-subcerts-06 (work in progress), February 2020.
- [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.privacy-pass]
Davidson, A. and N. Sullivan, "The Privacy Pass Protocol", draft-privacy-pass-00 (work in progress), November 2019.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC2560] Myers, M., Ankney, R., Malpani, A., Galperin, S., and C. Adams, "X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP", RFC 2560, DOI 10.17487/RFC2560, June 1999, <<https://www.rfc-editor.org/info/rfc2560>>.
- [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, <<https://www.rfc-editor.org/info/rfc3261>>.

- [RFC5636] Park, S., Park, H., Won, Y., Lee, J., and S. Kent, "Traceable Anonymous Certificate", RFC 5636, DOI 10.17487/RFC5636, August 2009, <<https://www.rfc-editor.org/info/rfc5636>>.
- [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, <<https://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, <<https://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, <<https://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, <<https://www.rfc-editor.org/info/rfc7340>>.
- [RFC7748] Langley, A., Hamburg, M., and S. Turner, "Elliptic Curves for Security", RFC 7748, DOI 10.17487/RFC7748, January 2016, <<https://www.rfc-editor.org/info/rfc7748>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8224] Peterson, J., Jennings, C., Rescorla, E., and C. Wendt, "Authenticated Identity Management in the Session Initiation Protocol (SIP)", RFC 8224, DOI 10.17487/RFC8224, February 2018, <<https://www.rfc-editor.org/info/rfc8224>>.
- [RFC8225] Wendt, C. and J. Peterson, "PASSporT: Personal Assertion Token", RFC 8225, DOI 10.17487/RFC8225, February 2018, <<https://www.rfc-editor.org/info/rfc8225>>.
- [RFC8226] Peterson, J. and S. Turner, "Secure Telephone Identity Credentials: Certificates", RFC 8226, DOI 10.17487/RFC8226, February 2018, <<https://www.rfc-editor.org/info/rfc8226>>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <<https://www.rfc-editor.org/info/rfc8446>>.

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@team.neustar

Network Working Group
Internet-Draft
Updates: RFC8224 (if approved)
Intended status: Standards Track
Expires: January 14, 2021

J. Peterson
Neustar
July 13, 2020

PASSporT Extension for Diverted Calls
draft-ietf-stir-passport-divert-09

Abstract

PASSporT is specified in RFC 8225 to convey cryptographically-signed information about the people involved in personal communications. This document extends PASSporT to include an indication that a call has been diverted from its original destination to a new one. This information can greatly improve the decisions made by verification services in call forwarding scenarios. Also specified here is an encapsulation mechanism for nesting a PASSporT within another PASSporT that assists relying parties in some diversion scenarios.

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 <https://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, 2021.

Copyright Notice

Copyright (c) 2020 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 (<https://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. The 'div' PASSporT Type and Claim	4
4. Using 'div' in SIP	6
4.1. Authentication Service Behavior	6
4.2. Verification Service Behavior	8
5. The 'div-o' PASSporT Type	10
5.1. Processing 'div-o' PASSporTs	12
6. Definition of 'opt'	13
7. 'div' and Redirection	13
8. Extending 'div' to work with Service Logic Tracking	14
9. Acknowledgments	15
10. IANA Considerations	15
10.1. JSON Web Token Claims Registrations	15
10.1.1. 'div' registration	15
10.1.2. 'opt' registration	16
10.2. PASSporT Type Registrations	16
11. Privacy Considerations	16
12. Security Considerations	17
13. References	17
13.1. Normative References	17
13.2. Informative References	18
Appendix A. Appendix A: Keys for Examples	19
Author's Address	19

1. Introduction

A Personal Assertion Token (PASSporT [RFC8225]) is a token format based on the JSON Web Token (JWT [RFC7519]) for conveying cryptographically-signed information about the people involved in personal communications; it is used by the Secure Telephone Identity Revisited (STIR [RFC8224]) protocol to convey a signed assertion of the identity of the participants in real-time communications established via a protocol like SIP. This specification extends PASSporT to include an indication that a call has been diverted from its original destination to a new one.

Although the STIR problem statement [RFC7340] is focused on preventing the impersonation of the caller's identity, which is a common enabler for threats such as robocalling and voicemail hacking on the telephone network today, it also provides a signature over the

called number at the time that the authentication service sees it. As [RFC8224] Section 12.1 describes, this protection over the contents of the To header field is intended to prevent a class of cut-and-paste attacks. If Alice calls Bob, for example, Bob might attempt to cut-and-paste the Identity header field in Alice's INVITE into a new INVITE that Bob sends to Carol, and thus be able to fool Carol into thinking the call came from Alice and not Bob. With the signature over the To header field value, the INVITE Carol sees will clearly have been destined originally for Bob, and thus Carol can view the INVITE as suspect.

However, as [RFC8224] Section 12.1.1 points out, it is difficult for Carol to confirm or reject these suspicions based on the information she receives from the baseline PASSporT object. The common "call forwarding" service serves as a good example of the reality that the original called party number is not always the number to which a call is delivered. There are a number of potential ways for intermediaries to indicate that such a forwarding operation has taken place. The address in the To header field value of SIP requests is not supposed to change, according to baseline SIP behavior [RFC3261]; instead, it is the Request-URI that is supposed to be updated when a call is retargeted. Practically speaking, however, many operational environments do alter the To header field. The History-Info header field [RFC7044] was created to store the Request-URIs that are discarded by a call in transit. The SIP Diversion header field [RFC5806], though historic, is still used for this purpose by some operators today. Neither of these header fields provide any cryptographic assurance of secure redirection, and they both record entries for minor syntactical changes in URIs that do not reflect a change to the actual target of a call.

This specification therefore extends PASSporT with an explicit indication that the original called number in PASSporT no longer reflects the destination to which a call is intended to be delivered. For this purpose, it specifies a Divert PASSporT type ("div") for use in common SIP retargeting cases; it is expected that in this case, SIP INVITE requests will carry multiple Identity header fields, each containing its own PASSporT. Throughout this document, PASSporTs that contain a "div" element will be referred to as "div" PASSporTs. Verification services and the relying parties who make authorization decisions about communications may use this diversion indication to confirm that a legitimate retargeting of the call has taken place, rather than a cut-and-paste attack. For out-of-band [I-D.ietf-stir-oob] use cases, and other non-SIP applications of PASSporT, a separate "div-o" PASSporT type is also specified, which defines an "opt" PASSporT element for carrying nested PASSporTs within a PASSporT. These shall in turn be referred to in this document as "div-o" PASSporTs.

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 BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. The 'div' PASSporT Type and Claim

This specification defines a PASSporT [RFC8225] type called "div" that may be employed by authentication services located at retargeting entities. All "div" PASSporTs MUST contain a new JSON Web Token "div" claim, also specified in this document, which indicates a previous destination for a call during its routing process. When a retargeting entity receives a call signed with a PASSporT, it may act as an authentication service and create a new PASSporT containing the "div" claim to attach to the call.

Note that a new PASSporT is only necessary when the canonical form of the "dest" identifier (per the canonicalization procedures in [RFC8224] Section 8.3) changes due to this retargeting. If the canonical form of the "dest" identifier is not changed during retargeting, then a new PASSporT with a "div" claim MUST NOT be produced.

The headers of the new PASSporTs generated by retargeting entities MUST include the "div" PASSporT type, and an "x5u" field pointing to a credential that the retargeting entity controls. "div" PASSporTs MUST use full form instead of compact form. The new PASSporT header will look as follows:

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

A "div" PASSporT claims set is populated with elements drawn from the PASSporT(s) received for a call by the retargeting entity: at a high level, the original identifier for the called party in the "dest" object will become the "div" claim in the new PASSporT. If the "dest" object of the original PASSporT contains multiple identifiers, because it contains one or more name/value pairs with an array as its value, the retargeting entity MUST select only one identifier from the value(s) of the "dest" object to occupy the value of the "div" field in the new PASSporT. Moreover, it MUST select an identifier that is within the scope of the credential that the retargeting

entity will specify in the "x5u" of the PASSporT header (as described below).

The new target for the call selected by the retargeting entity becomes the value of the "dest" object of the new PASSporT. The "orig" object MUST be copied into the new PASSporT from the original PASSporT received by the retargeting entity. The retargeting entity SHOULD retain the "iat" object from the original PASSporT, though if in the underlying signaling protocol (e.g. SIP) the retargeting entity changes the date and time information in the retargeted request, the new PASSporT should instead reflect that date and time. No other claims or extensions are to be copied from the original PASSporT to the "div" PASSporT.

So, for an original PASSporT claims set of the form:

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

If the retargeting entity is changing the target from 12155551213 to 12155551214, the claims set of a "div" PASSporT generated by the retargeting entity would look as follows:

```
{ "dest":{"tn":["12155551214"]},
  "div":{"tn":"12155551213"},
  "iat":1443208345,
  "orig":{"tn":"12155551212"} }
```

The combined full form PASSporT (with a signature covered by the ES256 keys given in Appendix A) would look as follows:

```
eyJhbGciOiJFUzI1NiIsInBwdCI6ImRpdIIsInR5cCI6InBhc3Nwb3J0IiwieDV1Ij \
oiaHR0cHM6Ly93d3cuZXhhbXBsZS5jb20vY2VydC5jZXIifQ.eyJkZXN0Ijp7InRuI \
jpbIjEyMTU1NTUxMjE0Ii19LCJkaXYiOmsidG4iOiIxMjE1NTU1NTEyMTMifSwiaWF \
0IjoxNDQzMjA4MzQ1LCJvcmlnIjp7InRuIjoiaMTIxNTU1NTEyMTIifX0.xBHWipDEE \
J8a6TsdX6xUXAnblsFiGUiAxwLiv0HLC9IICj6eG9jQd6WzeSSjHRBwxmChHhVIiMT \
SqIlk3yCNkg
```

The same "div" PASSporT would result if the "dest" object of the original PASSporT contained an array value, such as {"tn":["12155551213","19995551234"]}, and the retargeting entity chose to retarget from the first telephone number in the array. Every "div" PASSporT is diverting from only one identifier.

Note that the "div" element may contain other name/value pairs than just a destination, including a History-Info indicator (see Section 8). After the PASSporT header and claims have been

constructed, their signature is generated per the guidance in [RFC8225] - except for the credential required to sign it. While in the ordinary construction of a PASSporT, the credential used to sign will have authority over the identity in the "orig" claim (for example, a certificate with authority over the telephone number in "orig" per [RFC8226]), for all PASSporTs using the "div" type the signature MUST be created with a credential with authority over the identity present in the "div" claim. So for the example above, where the original "dest" is "12155551213", the signer of the new PASSporT object MUST have authority over that telephone number, and need not have any authority over the telephone number present in the "orig" claim.

Note that Identity header fields are not ordered in a SIP request, and in a case where there is a multiplicity of Identity header fields in a request, some sorting may be required to match "div" PASSporTs to their originals.

PASSporTs of type "div" MUST NOT contain an "opt" (see Section 6) element in their payload.

4. Using 'div' in SIP

This section specifies SIP-specific usage for the "div" PASSporT type and its handling in the SIP Identity header field "ppt" parameter value. Other protocols using PASSporT may define behavior specific to their use of the "div" claim.

4.1. Authentication Service Behavior

An authentication service only adds an Identity header field value containing the "div" PASSporT type to a SIP request that already contains at least one Identity header field value; it MUST NOT add a "div" PASSporT to an INVITE that contains no Identity header field. The retargeting entity SHOULD act as a verification service and validate the existing Identity header field value(s) in the request before proceeding; in some high-volume environments, it may instead put that burden of validating the chain entirely on the terminating verification service. As the authentication service will be adding a new PASSporT that refers to an original, it MUST NOT remove the original request's Identity header field value before forwarding.

As was stated in Section 3, the authentication service MUST sign any "div" PASSporT with a credential that has a scope of authority covering the identity it populates in the "div" element value. Note that this is a significant departure from baseline STIR authentication service behavior, in which the PASSporT is signed by a credential with authority over the "orig" field. The "div" value

reflects the URI that caused the call to be routed to the retargeting entity, so in ordinary operations, it would already be the STIR entity holding the appropriate private keying material for calls originating from that identity.

A SIP authentication service typically will derive the "dest" element value of a "div" PASSporT from a new Request-URI that is set for the SIP request before it is forwarded. Older values of the Request-URI may appear in header fields like Diversion or History-Info; this document specifies an optional interaction with History-Info below in Section 8. Note as well that because PASSporT operates on canonicalized telephone numbers and normalized URIs, many smaller changes to the syntax of identifiers that might be captured by other mechanisms that record retargeting (like History-Info) will likely not require a "div" PASSporT.

When adding an Identity header field with a PASSport claims set containing a "div" claim, SIP authentication services MUST also add a "ppt" parameter to that Identity header with a value of "div". For the example PASSport given in Section 3, the new Identity header added after retargeting might look as follows:

[illegible]

Note that in some deployments, an authentication service will need to generate "div" PASSporTs for a request that contains multiple non-"div" Identity header field values. For example, a request arriving at a retargeting entity might contain in different Identity header fields a baseline [RFC8224] PASSporT and a PASSporT of type "rph" [RFC8443] signed by a separate authority. Provided that these PASSporTs share the same "orig" and "dest" values, the retargeting entity's authentication service SHOULD generate only one "div" PASSporT. If the "orig" or "dest" of these PASSporTs differ, however, one "div" PASSporT SHOULD be generated for each non-"div" PASSporT. Note that this effectively creates multiple chains of "div" PASSporTs in a single request, which complicates the procedures that need to be performed at verification services.

Furthermore, a request may also be retargeted a second time, at which point the subsequent retargeting entity SHOULD generate one "div"

PASSporT for each previous "div" PASSporT in the request which contains a "dest" object with the value of the current target - but not for "div" PASSporTs with earlier targets. Ordinarily, the current target will be readily identifiable, as it will be in the last "div" PASSporT in each chain, and in SIP cases it will correspond to the Request-URI received by the retargeting entity. Moreover, the current target will be an identifier that the retargeting entity possesses a credential to sign for, which may not be true for earlier targets. Ultimately, on each retargeting, the number of PASSporTs added to a request will be equal to the number of non-"div" PASSporTs that do not share the same "orig" and "dest" object values.

4.2. Verification Service Behavior

[RFC8224] Section 6.2 Step 5 requires that specifications defining "ppt" values describe any additional or alternative verifier behavior. The job of a SIP verification service handling one or more "div" PASSporTs is very different from that of a traditional verification service. At a high level, the immediate responsibility of the verification service is to extract all PASSporTs from the two or more Identity header fields in a request, identify which are "div" PASSporTs and which are not, and then order and link the "div" PASSporTs to the original PASSporT(s) in order to build one or more chains of retargeting.

In order to validate a SIP request using the "div" PASSporT type, a verification service needs to inspect all of the valid Identity header field values associated with a request, as an Identity header field value containing "div" necessarily refers to an earlier PASSporT already in the message. For each "div" PASSporT, the verification service MUST find an earlier PASSporT that contains a "dest" claim with a value equivalent to the "div" claim in each "div" PASSporT. It is possible that this earlier PASSporT will also contain a "div", and that it will in turn chain to a still earlier PASSporT stored in a different Identity header field value. If a complete chain cannot be constructed, the verification service cannot complete "div" validation; it MAY still validate any non-"div" PASSporTs in the request per normal [RFC8224] procedures. If a chain has been successfully constructed, the verification service extracts from the outermost (that is, the most recent) PASSporT in the chain a "dest" field; this will be a "div" PASSporT that no other "div" PASSporT in the SIP request refers to. Its "dest" element value will be referred to in the procedures that follow as the value of the "outermost "dest" field."

Ultimately, by looking at this chain of transformations and validating the associated signatures, the verification service will

be able to ascertain that the appropriate parties were responsible for the retargeting of the call to its current destination. This can help the verification service to determine that the original PASSporT in the call was not simply used in a cut-and-paste attack and inform any associated authorization decisions in terms of how the call will be treated - though, per [RFC8224] Section 6.2.1, that decision is a matter of local policy and is thus outside the scope of this specification.

A verification service parses a chain of PASSporTs as follows:

First, the verification service MUST compare the value in the outermost "dest" field to the target of the call. As it is anticipated that SIP authentication services that create "div" PASSporTs will populate the "dest" header from the retargeted Request-URI (see Section 4.1), in ordinary SIP operations, the Request-URI is where verification services will find the latest call target. Note however that after a "div" PASSporT has been added to a SIP request, the Request-URI may have been updated during normal call processing to an identifier that no longer contains the logical destination of a call; in this case, the verification service MAY compare the "dest" field to a provisioned telephone number for the recipient.

Second, the verification service MUST validate the signature over the outermost "div" PASSporT, and establish that the credential that signed the "div" PASSporT has the authority to attest for the identifier in the "div" element of the PASSporT (per [RFC8224] Section 6.2 Step 3).

Third, the verification service MUST validate that the "orig" field of the innermost PASSporT of the chain (the only PASSporT in the chain which will not be of PASSporT type "div") is equivalent to the "orig" field of the outermost "div" PASSporT; in other words, that the original calling identifier has not been altered by retargeting authentication services. If the "orig" value has changed, the verification service MUST treat the entire PASSporT chain as invalid. The verification service MUST also verify that all other "div" PASSporTs in the chain share the same "orig" value. Then the verification service validates the relationship of the "orig" field to the SIP-level call signaling per the guidance in [RFC8224] Section 6.2 Step 2.

Fourth, the verification service MUST check the date freshness in the outermost "div" PASSporT per [RFC8224] Section 6.2 Step 4. It is furthermore RECOMMENDED that the verification service check that the "iat" field of the innermost PASSporT is also within the date freshness interval; otherwise the verification service could

allow attackers to replay an old, stale PASSporT embedded in a fresh "div". However, note that in some use cases, including certain ways that call transfers are implemented, it is possible that an established call will be retargeted long after it has originally been placed, and verification services may want to allow a longer window for the freshness of the innermost PASSporT if the call is transferred from a trusted party (as an upper bound, a freshness window on the order of three hours might suffice).

Fifth, the verification service MUST inspect and validate the signatures on each and every PASSporT object in the chain between the outermost "div" PASSporT and the innermost PASSporT. Note that (per Section 4.1) a chain may terminate at more than one innermost PASSporT, in cases where a single "div" is used to retarget from multiple innermost PASSporTs. Also note that [RFC8224] Section 6.2 Step 1 applies to the chain validation process: if the innermost PASSporT contains an unsupported "ppt", its chain MUST be ignored.

Note that the To header field is not used in the first step above. Optionally, the verification service MAY verify that the To header field value of the received SIP signaling is equal to the "dest" value in the innermost PASSporT; however, as has been observed in some deployments, the original To header field value may be altered by intermediaries to reflect changes of target. Deployments that change the original To header field value to conceal the original destination of the call from the ultimate recipient should note that the original destination of a call may be preserved in the innermost PASSporT. Future work on "div" might explore methods to implement that sort of policy while retaining a secure chain of redirection.

5. The 'div-o' PASSporT Type

This specification defines a "div-o" PASSporT type that uses the "div" claim element in conjunction with the "opt" (Section 6) claim element. As is the case with "div" PASSporT type, a "div-o" PASSporT is created by an authentication service acting for a retargeting entity, but instead of generating a separate "div" PASSporT to be conveyed alongside an original PASSporT, the authentication service in this case embeds the original PASSporT inside the "opt" element of the "div-o" PASSporT. The "div-o" extension is designed for use in non-SIP or gatewayed SIP environments where the conveyance of PASSporTs in separate Identity header fields is impossible, such as out-of-band [I-D.ietf-stir-oob] STIR scenarios.

The syntax of "div-o" PASSporTs is very similar to "div". A "div-o" PASSporT header object might look as follows:

```
{
  "typ": "passport",
  "ppt": "div-o",
  "alg": "ES256",
  "x5u": "https://www.example.com/cert.cer" }
```

Whereas a "div" PASSporT claims set contains only the "orig", "dest", "iat", and "div" elements, the "div-o" additionally MUST contain an "opt" element (see Section 6), which encapsulates the full form of the previous PASSporT from which the call was retargeted, triggering the generation of this "div-o". The format of the "opt" element is identical to the encoded PASSporT format given in Appendix A of [RFC8225].

So, for an original PASSporT claims set of the form:

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

If the retargeting entity is changing the target from 12155551213 to 12155551214, the new PASSporT claims set for "div-o" would look as follows:

```
{ "orig":{"tn":"12155551212"},  
  "dest":{"tn":["12155551214"]},  
  "iat":1443208345,  
  "div":{"tn":"121555551213"},  
  "opt":{"eyJhbGciOiJFUzI1NiIsInR5cCI6InBhc3Nwb3J0IiwieDVlIjoiaHR0c \\  
HM6Ly93d3cuZXBhbXB5ZS5jb20vY2VydC5jZXIifQ.eyJkZXN0Ijp7InRuIjpbIj \\  
EymTU1NTUxMjEzIEllL19LCjPjYXQiOjE0NDMyMDgzNDUsIm9yaWciOnsidG4iOiIxMj \\  
ElNTU1MTUxMjEzJfQ.1bEzkzcNbKvgz4QoMx0_DJT2T8qFMDC1SPqHPXl1WvbauzRJ \\  
RvYlZqQ0qqGTls8tJ_wXjVe07Z3wvDrdApHhhYw"} }
```

While in ordinary operations, it is not expected that SIP would carry a "div-o" PASSporT, it might be possible in some gatewaying scenarios. The resulting full form Identity header field with a "div-o" PASSporT would look as follows:

```
Identity:eyJhbGciOiJFUzI1NiIsInBwdCI6ImRpdilvIiwidHlwIjoicGFzc3Bvc\nnQlLCJ4NXUiOiJodHRwczovL3d3dy5leGFtcGxlLmNvbS9jZXJ0LmNlciJ9.eyJkZX\nN0Ijp7InRuIjoimTIxNTU1NTEyMTQifSwiZGl2Ijp7InRuIjoimTIxNTU1NTUxMjEz\nIn0sImIhdCI6MTQ0MzIwODM0NSwib3B0IjoizXlKaGJHY2lPaUpGVXpJMU5pSXNJbl\nI1YONJNklUQmhjM053YjNKMElpd2llRFYxSWpvaWFIUjBjSE02THk5M2QzY3VaWGho\nYlhCc1pTNWpiMjB2WTJWeWRDNWpaWElpZlEuZXlKa1pYTjBJanA3SW5SdUlcGJJak\nV5TVRVMU5UVXhNakV6SWwxOUxDSnBZWFFpT2pFME5ETXlNRGd6TkrVc0ltOXlhV2Np\nT25zaWRHNGlPaU14TWpFMU5UVTFNVEl4TWlKOWZRLjFiRXpremNOYkt2Z3o0UW9NeD\nBfREoyVDhxRk1EQzFzUHF1UFhsMVd2YmF1elJKUnZzBfpxUTBxZ0dUbFM4dEpfdlhq\nVmUwNl0zd3ZecmRbcEhoAF13Iiwib3JpZyI6eyJ0biI6IjEyMTU1NTUxMjEyIn19.C\nHeA9wRnth17paMe6rP0ATARpmFCXjmi_vF_HRz2O_oulB_R-G9xZNiLVvmvHv4gk6LI\nLaDV2y2VtHTLIEgmHig;\ninfo=<https://www.example.com/cert.cer>;ppt="div-o"
```

5.1. Processing 'div-o' PASSporTs

The authentication and verification service procedures required for "div-o" closely follow the guidance given in Section 4.1 and Section 4.2, with the major caveats being first, that they do store or retrieve PASSporTs via the Identity header field values of SIP requests, and second, that they process nested PASSporTs in the "opt" claim element. But transposing the rest of the behaviors described above to creating and validating "div-o" PASSporTs is straightforward.

For the "div-o" PASSporT type, retargeting authentication services that handle calls with one or more existing PASSporTs will create a corresponding "div-o" PASSporT for each received PASSporT. Each "div-o" PASSporT MUST contain an "opt" claim set element with the value of the original PASSporT from which the "div-o" was created; and as specified in Section 4.1, the authentication service MUST populate the "div" claim set element of the "div-o" PASSporT with the "dest" field of the original PASSporT. Each received PASSporT may in turn contain its own "opt" claim set element, if the retargeting authentication service is not the first in its chain. Note that if the retargeting authentication service is handling a call with multiple PASSporTs, which in ordinary SIP operation would result in the construction of multiple "div" chains, it will in effect be generating one "div-o" PASSporT per chain.

The job of a verification service is in many ways easier for "div-o" than for "div", as the verification service has no need to correlate the PASSporTs it receives and assemble them into chains, as any chains in "div-o" will be nested through the "opt" element. Nonetheless, the verification services MUST perform the same chain validation described in Section 4.2 to validate that each nested PASSporT shares the same "orig" field as its enclosing PASSporT, and that the "dest" field of each nested PASSporT corresponds to the

"div" field of its enclosing PASSporT. The same checks MUST also be performed for freshness, signature validation, and so on. It is similarly OPTIONAL for the verification service to determine that the "dest" claims element of the outermost PASSporT corresponds to the called party indication of receive telephone signaling, where such indication would vary depending on the using protocol.

How authentication services or verification services receive or transport PASSporTs for "div-o" is outside the scope of this document, and dependent on the using protocol.

6. Definition of 'opt'

The presence of an "Original PASSporT" ("opt") claims set element signifies that a PASSporT encapsulates another entire PASSporT within it, typically a PASSporT that was transformed in some way to create the current PASSporT. Relying parties may need to consult the encapsulated PASSporT in order to validate the identity of a caller. "opt" as defined in this specification may be used by future PASSporT extensions as well as in conjunction with "div-o".

"opt" MUST contain a quoted full-form PASSporT as specified by [RFC8225] Appendix A; it MUST NOT contain a compact form PASSporT. For an example of a "div-o" PASSporT containing "opt," see Section 5.

7. 'div' and Redirection

The "div" mechanism exists primarily to prevent false negatives at verification services when an arriving SIP request, due to intermediary retargeting, does not appear to be intended for its eventual recipient, because the original PASSporT "dest" value designates a different destination.

Any intermediary that assigns a new target to a request can, instead of retargeting and forwarding the request, instead redirect with a 3xx response code. In ordinary operations, a redirection poses no difficulties for the operations of baseline STIR: when the user agent client (UAC) receives the 3xx response, it will initiate a new request to the new target (typically the target carried in the Contact header field value of the 3xx), and the "dest" of the PASSporT created for the new request will match that new target. As no impersonation attack can arise from this case, it creates no new requirements for STIR.

However, some UACs record the original target of a call with mechanisms like History-Info [RFC7044] or Diversion [RFC5806], and may want to leverage STIR to demonstrate to the ultimate recipient that the call has been redirected securely: that is, that the

original destination was the one that sent the redirection message that led to the recipient receiving the request. The semantics of the PASSporT necessary for that assertion are the same as those for the "div" retargeting cases above. The only wrinkle is that the PASSporT needs to be generated by the redirecting entity and sent back to the originating user agent client within the 3xx response.

This introduces more complexity than might immediately be apparent. In the first place, a 3xx response can convey multiple targets through the Contact header field value; to accommodate this, the "div" PASSporT MAY include one "dest" object array value per Contact, but if the retargeting entity wants to keep the Contact list private from targets, it may need to generate one PASSporT per Contact. Bear in mind as well that the original SIP request could have carried multiple Identity header field values that had been added by different authentication services in the request path, so a redirecting entity might need to generate one "div" PASSporT for each PASSporT in the original request. Often, this will mean just one "div" PASSporT, but for some deployment scenarios, it could require an impractical number of combinations. But in very complex call routing scenarios, attestation of source identity would only add limited value anyway.

STIR-aware SIP intermediaries that redirect requests MAY therefore convey one or more PASSporTs in the backwards direction within Identity header fields. These redirecting entities will act as authentication services for "div" as described in Section 4.1. This document consequently updates [RFC8224] to permit carrying Identity header fields in SIP 300-class responses. It is left to the originating user agent to determine which Identity header fields should be copied from the 3xx into any new requests resulting from the redirection, if any: use of these Identity header fields by entities receiving a 3xx response is OPTIONAL.

Finally, note that if an intermediary in the response path consumes the 3xx and explores new targets itself while performing sequential forking, it will effectively retarget the call on behalf of the redirecting server, and this will create the same need for "div" PASSporTs as any other retargeted call. These intermediaries MAY also copy PASSporTs from the 3xx response and insert them into sequential forking requests, if appropriate.

8. Extending 'div' to work with Service Logic Tracking

It is anticipated that "div" may be used in concert with History-Info [RFC7044] in some deployments. It may not be clear from the "orig" and "dest" values which History-Info header a given PASSporT correlates to, especially because some of the target changes tracked

by History-Info will not be reflected in a "div" PASSport (see Section 1). Therefore an "hi" element as defined here may appear in "div" corresponding to the History-Info header field index parameter value. So for a History-Info header field with an index value of "1.2.1", the claims set of the corresponding PASSport with "div" might look like:

```
{ "orig":{"tn":"12155551212"},
  "dest":{"tn":["12155551214"]},
  "iat":1443208345,
  "div":{"tn":"12155551213",
        "hi":"1.2.1"} }
```

Past experience has shown that there may be additional information about the motivation for retargeting that relying parties might consider when making authorization decisions about a call, see for example the "reason" associated with the SIP Diversion header field [RFC5806]. Future extensions to this specification might incorporate reasons into "div".

9. Acknowledgments

We would like to thank Ning Zhang, Dave Hancock, Chris Wendt, Sean Turner, Russ Housley, Ben Campbell, Eric Burger, and Robert Sparks for contributions to this document.

10. IANA Considerations

This document contains actions for the IANA.

10.1. JSON Web Token Claims Registrations

This specification requests that the IANA add two new claims to the JSON Web Token Claims registry as defined in [RFC7519].

10.1.1. 'div' registration

Claim Name: "div"

Claim Description: Diverted Target of a Call

Change Controller: IESG

Specification Document(s): [RFCThis]

10.1.2. 'opt' registration

Claim Name: "opt"

Claim Description: Original PASSport (in Full Form)

Change Controller: IESG

Specification Document(s): [RFCThis]

10.2. PASSport Type Registrations

This specification defines two new PASSport types for the PASSport Extensions Registry defined in [RFC8225], which resides at <https://www.iana.org/assignments/passport/passport.xhtml#passport-extensions>. They are:

"div" as defined in [RFCThis] Section 3.

"div-o" as defined in [RFCThis] Section 5.

11. Privacy Considerations

There is an inherent trade-off in any mechanism that tracks in SIP signaling how calls are routed through a network, as routing decisions may expose policies set by users for how calls are forwarded, potentially revealing relationships between different identifiers representing the same user. Note however that in ordinary operations, this information is revealed to the user agent service of the called party, not the calling party. It is usually the called party who establishes these forwarding relationships, and if indeed some other party is responsible for calls being forwarded to the called party, many times the called party should likely be entitled to information about why they are receiving these calls. Similarly, a redirecting entity who sends a 3xx in the backwards direction knowingly shares information about service logic with the caller's network. However, as there may be unforeseen circumstances where the revelation of service logic to the called party poses a privacy risk, implementers and users of this or similar diversion-tracking techniques should understand the trade-off.

Furthermore, it is a general privacy risk of identity mechanisms overall that they do not interface well with anonymization services; the interaction of STIR with anonymization services is detailed in [RFC8224] Section 11. Any forwarding service that acts as an anonymizing proxy may not be able to provide a secure chain of retargeting due to the obfuscation of the originating identity.

Also see [RFC8224] Section 11 for further considerations on the privacy of using PASSporTs in SIP.

12. Security Considerations

This specification describes a security feature, and is primarily concerned with increasing security when calls are forwarded. Including information about how calls were retargeted during the routing process can allow downstream entities to infer particulars of the policies used to route calls through the network. However, including this information about forwarding is at the discretion of the retargeting entity, so if there is a requirement to keep an intermediate called number confidential, no PASSporT should be created for that retargeting - the only consequence will be that downstream entities will be unable to correlate an incoming call with the original PASSporT without access to some prior knowledge of the policies that could have caused the retargeting.

Any extension that makes PASSporTs larger creates a potential amplification mechanism for SIP-based DDoS attacks. Since diversion PASSporTs are created as a part of normal forwarding activity, this risk arises at the discretion of the retargeting domain: simply using 3xx response redirections rather than retargeting (by supplying a "div" per Section 7) mitigates the potential impact. Under unusual traffic loads, even domains that might ordinarily retarget requests can switch to redirection.

SIP has an inherent capability to redirect requests, including forking them to multiple parties -- potentially a very large numbers of parties. The use of the "div" PASSporT type does not grant any additional powers to attackers who hope to place bulk calls; if present, the "div" PASSporT instead identifies the party responsible for the forwarding. As such, senders of bulk unsolicited traffic are unlikely to find the use of "div" attractive.

13. References

13.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://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, <<https://www.rfc-editor.org/info/rfc3261>>.
- [RFC7044] Barnes, M., Audet, F., Schubert, S., van Elburg, J., and C. Holmberg, "An Extension to the Session Initiation Protocol (SIP) for Request History Information", RFC 7044, DOI 10.17487/RFC7044, February 2014, <<https://www.rfc-editor.org/info/rfc7044>>.
- [RFC7519] Jones, M., Bradley, J., and N. Sakimura, "JSON Web Token (JWT)", RFC 7519, DOI 10.17487/RFC7519, May 2015, <<https://www.rfc-editor.org/info/rfc7519>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8224] Peterson, J., Jennings, C., Rescorla, E., and C. Wendt, "Authenticated Identity Management in the Session Initiation Protocol (SIP)", RFC 8224, DOI 10.17487/RFC8224, February 2018, <<https://www.rfc-editor.org/info/rfc8224>>.
- [RFC8225] Wendt, C. and J. Peterson, "PASSporT: Personal Assertion Token", RFC 8225, DOI 10.17487/RFC8225, February 2018, <<https://www.rfc-editor.org/info/rfc8225>>.
- [RFC8226] Peterson, J. and S. Turner, "Secure Telephone Identity Credentials: Certificates", RFC 8226, DOI 10.17487/RFC8226, February 2018, <<https://www.rfc-editor.org/info/rfc8226>>.

13.2. Informative References

- [I-D.ietf-stir-oob] Rescorla, E. and J. Peterson, "STIR Out-of-Band Architecture and Use Cases", draft-ietf-stir-oob-07 (work in progress), March 2020.
- [RFC5806] Levy, S. and M. Mohali, Ed., "Diversion Indication in SIP", RFC 5806, DOI 10.17487/RFC5806, March 2010, <<https://www.rfc-editor.org/info/rfc5806>>.

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

[RFC8443] Singh, R., Dolly, M., Das, S., and A. Nguyen, "Personal Assertion Token (PASSporT) Extension for Resource Priority Authorization", RFC 8443, DOI 10.17487/RFC8443, August 2018, <<https://www.rfc-editor.org/info/rfc8443>>.

Appendix A. Appendix A: Keys for Examples

The following EC256 keys are used in the signing examples given in this document. WARNING: Do not use this key pair in production systems.

-----BEGIN PUBLIC KEY-----

MFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAE mzGM1VsO+3IqbMF54rQMaYKQftO4
hUYm9wv5wutLgEd9FsiTy3+4+Wa2O7pffOXPC0QzO+yD8hGEXGP/2mZo6w==

-----END PUBLIC KEY-----

-----BEGIN EC PRIVATE KEY-----

MHcCAQEEIFKCsFZ4Wsw3ZpBxgc4Z0sOjaXDdMk07NylfKg6OntAkoAoGCCqGSM49
AwEHoUQDQgAE mzGM1VsO+3IqbMF54rQMaYKQftO4hUYm9wv5wutLgEd9FsiTy3+4
+Wa2O7pffOXPC0QzO+yD8hGEXGP/2mZo6w==

-----END EC PRIVATE KEY-----

Author's Address

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

Email: jon.peterson@team.neustar

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: 26 October 2022

C. Wendt
Somos Inc.
J. Peterson
Neustar Inc.
24 April 2022

PASSport Extension for Rich Call Data
draft-ietf-stir-passport-rcd-17

Abstract

This document extends PASSport, a token for conveying cryptographically-signed call information about personal communications, to include rich meta-data about a call and caller that can be signed and integrity protected, transmitted, and subsequently rendered to the called party. This framework is intended to include and extend caller and call specific information beyond human-readable display name comparable to the "Caller ID" function common on the telephone network. The JSON element defined for this purpose, Rich Call Data (RCD), is an extensible object defined to either be used as part of STIR or with SIP Call-Info to include related information about calls that helps people decide whether to answer an incoming set of communications from another party. This signing of the RCD information is also enhanced with a integrity mechanism that is designed to protect the authoring and transport of this information between authoritative and non-authoritative parties generating and signing the Rich Call Data for support of different usage and content policies.

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 <https://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 26 October 2022.

Copyright Notice

Copyright (c) 2022 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 (<https://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 Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction	3
2. Terminology	4
3. Overview of the use of the Rich Call Data PASSporT extension	4
4. Overview of Rich Call Data Integrity	5
5. PASSporT Claim "rcd" Definition and Usage	7
5.1. PASSporT "rcd" Claim	7
5.1.1. "nam" key	7
5.1.2. "apn" key	7
5.1.3. "icn" key	8
5.1.4. "jcd" key	9
5.1.5. "jcl" key	9
6. "rcdi" RCD Integrity Claim Definition and Usage	10
6.1. Creation of the "rcd" element digests	11
6.1.1. "nam" and "apn" elements	12
6.1.2. "icn" elements	12
6.1.3. "jcd" elements	12
6.1.4. "jcl" elements	14
6.2. JWT Claim Constraints for "rcd" claims only	15
7. JWT Claim Constraints usage for "rcd" and "rcdi" claims	15
8. PASSporT "crn" claim - Call Reason Definition and Usage	16
8.1. JWT Constraint for "crn" claim	16
9. Rich Call Data Claims Usage Rules	17
9.1. "rcd" PASSporT Verification	17
9.2. "rcdi" Integrity Verification	18
9.3. Example "rcd" PASSporTs	18
10. Compact form of "rcd" PASSporT	20
10.1. Compact form of the "rcd" PASSporT claim	20
10.2. Compact form of the "rcdi" PASSporT claim	21
10.3. Compact form of the "crn" PASSporT claim	21
11. Further Information Associated with Callers	21
12. Third-Party Uses	22

12.1. Signing as a Third Party	23
13. Levels of Assurance	24
14. Using "rcd" in SIP	24
14.1. Authentication Service Behavior	24
14.2. Verification Service Behavior	25
15. Using "rcd" and "rcdi" as additional claims to other PASSporT extensions	26
15.1. Procedures for applying "rcd" as claims only	27
15.2. Example for applying "rcd" as claims only	27
16. Acknowledgements	28
17. IANA Considerations	28
17.1. JSON Web Token Claim	28
17.2. PASSporT Types	29
17.3. PASSporT RCD Types	29
18. Security Considerations	29
18.1. The use of JWT Claim Constraints in delegate certificates to exclude unauthorized claims	30
19. References	30
19.1. Normative References	30
19.2. Informative References	32
Authors' Addresses	32

1. Introduction

PASSporT [RFC8225] is a token format based on JWT [RFC7519] for conveying cryptographically-signed information about the parties involved in personal communications; it is used to convey a signed assertion of the identity of the participants in real-time communications established via a protocol like SIP [RFC8224]. The STIR problem statement [RFC7340] declared securing the display name of callers outside of STIR's initial scope, so baseline STIR provides no features for caller name. This specification documents an optional mechanism for PASSporT and the associated STIR procedures which extend PASSporT objects to protect additional elements conveying richer information: information that is intended to be rendered to assist a called party in determining whether to accept or trust incoming communications. This includes the name of the person or entity on one side of a communications session, the traditional "Caller ID" of the telephone network, along with related display information that would be rendered to the called party during alerting, or potentially used by an automaton to determine whether and how to alert a called party.

Traditional telephone network signaling protocols have long supported delivering a 'calling name' from the originating side, though in practice, the terminating side is often left to derive a name from the calling party number by consulting a local address book or an external database. SIP similarly can carry this information in a

'display-name' in the From header field value from the originating to terminating side, or alternatively in the Call-Info header field. However, both are unsecured fields that really cannot be trusted in most interconnected SIP deployments, and therefore is a good starting point for a framework that utilizes STIR techniques and procedures for protecting call related information including but not limited to calling name.

As such, the baseline use-case for this document extends PASSporT to provide cryptographic protection for the "display-name" field of SIP requests as well as further "rich call data" (RCD) about the caller, which includes the contents of the Call-Info header field or other data structures that can be added to the PASSporT. This document furthermore specifies a third-party profile that would allow external authorities to convey rich information associated with a calling number via a new type of PASSporT. Finally, this document describes how to preserve the integrity of the RCD in scenarios where there may be non-authoritative users initiating and signing RCD and therefore a constraint on the RCD data that a PASSporT can attest via certificate-level controls.

2. Terminology

The keywords "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 BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Overview of the use of the Rich Call Data PASSporT extension

The main intended use of the signing of Rich Call Data (RCD) using STIR within SIP [RFC8224] or more generally as a PASSporT extension [RFC8225] is for the entity that originates a call, either directly the caller themselves, if they are authoritative, or a service provider or third-party service that may be authoritative over the rich call data on behalf of the caller.

The RCD associated with the identity of the calling party described in this document is of two main categories. The first data is a more traditional set of info about a caller associated with "display-name" in SIP [RFC3261], typically a textual description of the caller, or alternate presentation numbers often used in From Header field [RFC3261] or P-Asserted-ID [RFC3325]. The second category is a set of RCD that is defined as part of the jCard definitions or extensions to that data. [I-D.ietf-sipcore-callinfo-rcd] describes the optional use of jCard in Call-Info header field as RCD with the "jcard" Call-Info purpose token. Either or both of these two types of data can be incorporated into an "rcd" claim defined in this document.

Additionally, in relation to the description of the specific communications event itself (versus the identity description in previous paragraph), [I-D.ietf-sipcore-callinfo-rcd] also describes a "call-reason" parameter intended for description of the intent or reason for a particular call. A new PASSporT claim "crn", or call reason, can contain the string or object that describes the intent of the call. This claim is intentionally kept separate from the "rcd" claim because it is envisioned that call reason is not the same as information associated with the caller and may change on a more frequent, per call, type of basis.

4. Overview of Rich Call Data Integrity

When incorporating call data that represents a user, even in traditional calling name services today, often there is policy and restrictions around what data is allowed to be used. Whether preventing offensive language or icons or enforcing uniqueness, potential trademark or copyright violations or other policy enforcement, there might be the desire to pre-certify or "vet" the specific use of rich call data. This document defines a mechanism that allows for a direct or indirect party that controls the policy to approve or certify the content, create a cryptographic digest that can be used to validate that data and applies a constraint in the certificate to allow the recipient and verifier to validate that the specific content of the RCD is as intended at its creation and approval or certification.

There are two mechanisms that are defined to accomplish that for two distinct categories of purposes. The first of the mechanisms include the definition of an integrity claim. The RCD integrity mechanism is a process of generating a sufficiently strong cryptographic digest for each resource referenced by a URI within a claim value (e.g., an image file referenced by "jcd" or a jCard referenced by "jcl"). This mechanism is inspired by and based on the W3C Subresource Integrity specification (<http://www.w3.org/TR/SRI/>). The second of the mechanisms uses the capability called JWT Claim Constraints, defined

in [RFC8226] and extended in [RFC9118]. The JWT Claim Constraints specifically guide the verifier within the certificate used to sign the PASSporT for the inclusion (or exclusion) of specific claims and their values, so that the content intended by the signer can be verified to be accurate.

Both of these mechanisms, integrity digests and JWT Claims Constraints, can be used together or separately depending on the intended purpose. The first category of purpose is whether the rich call data conveyed by the RCD passport is pass-by-value or passed-by-reference; i.e., is the information contained in the passport claims and therefore integrity protected by the passport signature, or is the information contained in an external resource referenced by a URI in the RCD PASSporT. The second category of purpose is whether the signer is authoritative or has responsibility for the accuracy of the RCD based on the policies of the eco-system the RCD PASSporTs are being used.

The following table provides an overview of the framework for how integrity should be used with RCD. (Auth represents authoritative in this table)

Modes	No external URIs	Includes URI refs
Auth	1: No integrity req	2: RDC Integrity
Non-Auth	3: JWT Claim Const.	4: RCD Integ./JWT Claim Const.

The first and simplest mode is exclusively for when all RCD content is directly included as part of the claims (i.e. no external reference URIs are included in the content) and when the signer is authoritative over the content. In this mode, integrity protection is not required and the set of claims is simply protected by the signature of the standard PASSporT [RFC8225] and SIP identity header [RFC8224] procedures. The second mode is an extension of the first where the signer is authoritative and an "rcd" claim contents include a URI identifying external resources. In this mode, an RCD Integrity or "rcdi" claim MUST be included. This integrity claim is defined later in this document and provides a digest of the "rcd" claim content so that, particularly for the case where there are URI references in the RCD, the content of that RCD can be comprehensively validated that it was received as intended by the signer of the PASSporT.

The third and fourth mode cover cases where there is a different authoritative entity responsible for the content of the RCD, separate from the signer of the PASSporT itself, allowing the ability to have forward control at the time of the creation of the certificate of the allowed or vetted content included in or referenced by the RCD claim contents. The primary framework for allowing the separation of authority and the signing of PASSporTs by non-authorized entities is detailed in [RFC9060] although other cases may apply. As with the first and second modes, the third and fourth modes differ with the absence or inclusion of externally referenced content using URIs.

5. PASSporT Claim "rcd" Definition and Usage

5.1. PASSporT "rcd" Claim

This specification defines a new JSON Web Token claim for "rcd", Rich Call Data, the value of which is a JSON object that can contain one or more key value pairs. This document defines a default set of key values.

5.1.1. "nam" key

The "nam" key value is a display name, associated with the originator of personal communications, which may for example derive from the display-name component of the From header field value of a SIP request or alternatively from the P-Asserted-Identity header field value, or a similar field in other PASSporT using protocols. This key **MUST** be included once as part of the "rcd" claim value JSON object. If there is no string associated with a display name, the claim value **MUST** then be an empty string.

5.1.2. "apn" key

The "apn" key value is an optional alternate presentation number associated with the originator of personal communications, which may for example derive from the user component of the From header field value of a SIP request (in cases where a network number is carried in the P-Asserted-Identity [RFC3325]), or alternatively from the Additional-Identity header field value [3GPP TS 24.229 v16.7.0], or a similar field in other PASSporT using protocols. Its intended semantics are to convey a number that the originating user is authorized to show to called parties in lieu of their default number, such as cases where a remote call agent uses the main number of a call center instead of their personal telephone number. The "apn" key value is a canonicalized telephone number per [RFC8224] Section 8.3. If present, this key **MUST** be included once as part of the "rcd" claim value JSON object.

The use of the optional "apn" key is intended for cases where the signer of an rcd PASSporT authorizes the use of an alternate presentation number by the user. How the signer determines that a user is authorized to present the number in question is a policy decision outside the scope of this document, however, the vetting of the alternate presentation number should follow the same level of vetting as telephone identities or any other information contained in an RCD PASSporT. This usage is intended as an alternative to conveying the presentation number in the "tel" key value of a jCard, in situations where no other rich jCard data needs to be conveyed with the call. Only one "apn" key may be present. "apn" MUST be used when it is the intent of the caller or signer to display the alternate presentation number even if "jcd" or "jcl" keys are present in a PASSporT with a "tel" key value.

5.1.3. "icn" key

The "icn" key value is an optional URI reference to an image that can be used to pictorially represent the originator of personal communications. This icon key value should be used as a base or default method of associating an image with a calling party.

When being used for SIP [RFC3261] this claim key value used to protect the call-info header field with a purpose parameter value of "icon" as described in Section 20.9 [RFC3261]. Example as follows:

```
Call-Info: <http://www.example.com/alice/photo.jpg>;  
  purpose=icon
```

Note that [I-D.ietf-sipcore-callinfo-rcd] extends the specific usage of "icon" in SIP in the context of the larger rich call data framework with specific guidance on referencing images and image types, sizes and formats.

It should be also noted that with jCard, as described in the following "jcd" and "jcl" key value sections and in [I-D.ietf-sipcore-callinfo-rcd], there are alternative ways of including photos and logos as URI references. The "icn" key should be then considered a base or default image and jCard usage should be considered for profiles and extensions that provide more direct guidance on the usage of specific defined usage of what each image type represents for the proper rendering to end users.

5.1.4. "jcd" key

The "jcd" key value is defined to contain a jCard [RFC7095] JSON object. This jCard object is intended to represent and derives from the Call-Info header field value defined in [I-D.ietf-sipcore-callinfo-rcd] with a type of "jcard". As also defined in [I-D.ietf-sipcore-callinfo-rcd], format of the jCard and properties used should follow the normative usage and formatting rules and procedures. It is an extensible object where the calling party can provide both the standard types of information defined in jCard or can use the built-in extensibility of the jCard specification to add additional information. The "jcd" key is optional. If included, this key MUST only be included once in the "rcd" JSON object and MUST NOT be included if there is a "jcl" key included. The use of "jcd" and "jcl" keys are mutually exclusive.

The jCard object value for "jcd" MUST only have referenced content for URI values that do not further reference URIs. Future specifications may extend this capability, but as stated in [I-D.ietf-sipcore-callinfo-rcd] it constrains the security properties of RCD information and the integrity of the content referenced by URIs.

Note: even though we refer to [I-D.ietf-sipcore-callinfo-rcd] as the definition of the jcard properties for usage in an "rcd" PASSport, other future specifications and protocols are encouraged to be adapted for use of "jcd" (or similarly "jcl" below) key beyond SIP and Call-Info.

5.1.5. "jcl" key

The "jcl" key value is defined to contain a URI that refers the recipient to a jCard [RFC7095] JSON object hosted on a HTTPS enabled web server. The web server MUST use the MIME media type for JSON text as application/json with a default encoding of UTF-8 [RFC4627]. This link may derive from the Call-Info header field value defined in [I-D.ietf-sipcore-callinfo-rcd] with a type of "jcard". As also defined in [I-D.ietf-sipcore-callinfo-rcd], format of the jCard and properties used should follow the normative usage and formatting rules and procedures. The "jcl" key is optional. If included, this key MUST only be included once in the "rcd" JSON object and MUST NOT be included if there is a "jcd" key included. The use of "jcd" and "jcl" keys are mutually exclusive.

The jCard object referenced by the URI value for "jcl" MUST only have referenced content for URI values that do not further reference URIs. Future specifications may extend this capability, but as stated in [I-D.ietf-sipcore-callinfo-rcd] it constrains the security properties of RCD information and the integrity of the content referenced by URIs.

6. "rcdi" RCD Integrity Claim Definition and Usage

The "rcdi" claim is included for the second and fourth modes described in the integrity overview Section 4 of this document. If this claim is present it MUST be included only once with the corresponding single "rcd" claim. The value of the "rcdi" claim is a JSON object that is defined as follows.

The claim value of "rcdi" claim key is a JSON object with a set of JSON key/value pairs. These objects correspond to each of the elements of the "rcd" claim object that require integrity protection with an associated digest over the content referenced by the key string. The individual digest of different elements of the "rcd" claim data and external URI referenced content is kept specifically separate to allow the ability to verify the integrity of only the elements that are ultimately retrieved or downloaded or rendered to the end-user.

The key value references a specific object within the "rcd" claim value using a JSON pointer defined in [RFC6901] with a minor additional rule to support external URI references that include JSON objects themselves, for the specific case of the use of "jcl". JSON pointer syntax is the key value that specifies exactly the part of JSON that is used to generate the digest which produce the resulting string that makes up the value for the corresponding key. Detailed procedures are provided below, but an example "rcdi" is provided here:

```
"rcdi" : {  
  "/jcl": "sha256-7kdCBZqH0nqMSPsmABvsKlHPhZESTgjojhdSJGRr3rk",  
  "/jcl/1/2/3": "sha256-jL4f47fF82LuwcrOrSyckA4SWr1ElfARHkW6kYo1JdI"  
}
```

The values of each key/value pair consists of a digest across either the direct values or indirectly referenced resources, combined with a string that defines the crypto algorithm used to generate the digest. RCD implementations MUST support the following hash algorithms, "SHA256", "SHA384", and "SHA512". The SHA-256, SHA-384, and SHA-512 are part of the SHA-2 set of cryptographic hash functions defined by the National Institute of Standards and Technologies (NIST). Implementations MAY support additional algorithms, but MUST NOT

support known weak algorithms such as MD5 or SHA-1. In the future, the list of algorithms may be re-evaluated based on security best practices. The algorithms are represented in the text by "sha256", "sha384", or "sha512". The character following the algorithm string MUST be a minus character, "-". The subsequent characters are the base64 encoded [RFC4648] digest of a canonicalized and concatenated string or binary data based on the JSON pointer referenced elements of "rcd" claim or the URI referenced content contained in the claim. The details of the determination of the input string used to determine the digest are defined in the next section.

6.1. Creation of the "rcd" element digests

"rcd" claim objects can contain "nam", "apn", "icn", "jcd", or "jcl" keys as part of the "rcd" JSON object claim value. This specification defines the use of JSON pointer [RFC6901] as a mechanism to reference specific "rcd" claim elements.

In order to facilitate proper verification of the digests and whether the "rcd" elements or content referenced by URIs were modified, the input to the digest must be completely deterministic at three points in the process. First, at the certification point where the content is evaluated to conform to the application policy and the JWT Claim Constraints is applied to the certificate containing the digest. Second, when the call is signed at the Authentication Service, there may be a local policy to verify that the provided "rcd" claim corresponds to each digest. Third, when the "rcd" data is verified at the Verification Service, the verification is performed for each digest by constructing the input digest string for the element being verified and referenced by the JSON pointer string.

The procedure for the creation of each "rcd" element digest string corresponding to a JSON pointer string key is as follows.

1. The JSON pointer either refers to a value that is a part or the whole of a JSON object or to a string that is a URI referencing an external resource.
2. For a JSON value, serialize the JSON to remove all white space and line breaks. The procedures of this deterministic JSON serialization are defined in [RFC8225], Section 9. The resulting string is the input for the hash function.
3. For any URI referenced content, the bytes of the body of the HTTP response is the input for the hash function.

6.1.1. "nam" and "apn" elements

In the case of "nam" and "apn", the only allowed value is a string. For both of these key values an "rcdi" JSON pointer or integrity digest is optional because the direct value is protected by the signature and can be constrained directly with JWTClaimConstraints. If used, the JSON key value referenced by the JSON pointer is the string includes the quotes, so quotes MUST be included to compute the digest.

6.1.2. "icn" elements

In the case of "icn", the only allowed value is a URI value that references an image file. If the URI references externally linked content there would need to be a JSON pointer and digest entry for the content in that linked resource. In order to reference the "icn" value for a digest, the JSON pointer string would be "/icn" and the digest string would be created using the image file data following the rules of JSON pointer. Even though this is probably not the typical case, an "rcdi" JSON pointer or integrity digest is optional if the image value is directly included via a data URI. However, even though the direct value can be protected by the signature and can be constrained directly with JWTClaimConstraints, since the length of the image data is likely much larger than the integrity digest, this specification would recommend the use of the "rcdi" JSON pointer and integrity digest as the constraint value in JWTClaimConstraints over the image data.

6.1.3. "jcd" elements

In the case of "jcd", the value associated is a jCard JSON object, which happens to be a JSON array with sub-arrays. JSON pointer notation uses numeric indexes into elements of arrays, including when those elements are arrays themselves.

As example, for the following "rcd" claim:

```

"rcd": {
  "jcd": ["vcard",
    [ ["version", {}, "text", "4.0"],
      [fn, {}, "text", "Q Branch"],
      [org, {}, "text", "MI6;Q Branch Spy Gadgets"],
      ["photo", {}, "uri",
        "https://example.com/photos/quartermaster-256x256.png"],
      ["logo", {}, "uri",
        "https://example.com/logos/mi6-256x256.jpg"],
      ["logo", {}, "uri",
        "https://example.com/logos/mi6-64x64.jpg"]
    ]
  ],
  "nam": "Q Branch Spy Gadgets"
}

```

In order to use JSON pointer to refer to the URIs, the following example "rcdi" claim includes a digest for the entire "jcd" array string as well as three additional digests for the URIs, where, as defined in [RFC6901] zero-based array indexes are used to reference the URI strings.

```

"rcdi": {
  "/jcd": "sha256-tbxXX9mRY2dtss3vNdNkNkt9hrV9N1LqGST2hDlw97I",
  "/jcd/1/3/3": "sha256-RojgWwU6xUtI4q82+kHPyHm1JKbm7+663bMvzymhk14",
  "/jcd/1/4/3": "sha256-jL4f47fF82LuwcrOrSyckA4SWr1ElfARHkW6kYo1JdI",
  "/jcd/1/5/3": "sha256-GKNxxqlLRarbyBNh7hc/4lbZAdK6B0kMRf1AMRWPkSo"
}

```

The use of a JSON pointer and integrity digest for the "jcd" claim key and value is optional. The "jcd" value is the directly included jCard array and can be protected by the signature and can be constrained directly with JWTClaimConstraints. However, for data length reasons (as with "icn" above) or more importantly for potential privacy and/or security considerations with a publically accessible certificate this specification would recommend the use of the "rcdi" JSON pointer and integrity digest as the constraint value in JWTClaimConstraints over the jCard data.

It is important to remember the array indexes for JSON Pointer are dependent on the order of the elements in the jCard. The use of digest for the "/jcd" corresponding to the entire jCard array string can be included as a redundant mechanism to avoid any possibility of substitution, insertion attacks, or other potential techniques that may be possible to avoid integrity detection.

Each URI referenced in the jCard array string MUST have a corresponding JSON pointer string key and digest value.

6.1.4. "jcl" elements

In the case of the use of a "jcl" URI reference to an external jCard, the procedures are similar to "jcd" with the exception and the minor modification to JSON pointer, where "/jcl" is used to refer to the external jCard array string and any following numeric array indexes added to the "jcl" (e.g., "/jcl/1/2/3") are treated as if the externally referenced jCard was directly part of the overall "rcd" claim JSON object. The following example illustrates a "jcl" version of the above "jcd" example.

```
"rcd": {
  "jcl": "https://example.com/qbranch.json",
  "nam": "Q Branch Spy Gadgets"
},
"rcdi": {
  "/jcl": "sha256-Gb010kj7Z9+plqbOkN32H+YX0Yav3fbioSk7DxQdGZU",
  "/jcl/1/3/3": "sha256-RojgWwU6xUtI4q82+kHPyHm1JKbm7+663bMvzymhkl4",
  "/jcl/1/4/3": "sha256-jL4f47fF82LuwcrOrSyckA4SWr1ElfARHkW6kYo1JdI",
  "/jcl/1/5/3": "sha256-GKNxxqlLRarbyBNh7hc/4lbZAdK6B0kMRf1AMRWPkSo"
}
```

The following is the example contents of resource pointed to by <https://example.com/qbranch.json> used to calculate the above digest for "/jcl"

```
[ "vcard",
  [ [ "version", {}, "text", "4.0" ],
    [ "fn", {}, "text", "Q Branch" ],
    [ "org", {}, "text", "MI6;Q Branch Spy Gadgets" ],
    [ "photo", {}, "uri",
      "https://example.com/photos/quartermaster-256x256.png" ],
    [ "logo", {}, "uri",
      "https://example.com/logos/mi6-256x256.jpg" ],
    [ "logo", {}, "uri",
      "https://example.com/logos/mi6-64x64.jpg" ]
  ]
]
```

6.2. JWT Claim Constraints for "rcd" claims only

For the third mode described in the integrity overview Section 4 of this document, where only JWT Claim Constraints for "rcd" claims without an "rcdi" claim is required, the procedure when creating the certificate with the intent to always include an "rcd" claim, to include a JWT Claim Constraints on inclusion of an "rcd" claim with the intended values required to be constrained by the certificate used to sign the PASSporT.

The "permittedValues" for the "rcd" claim may optionally contain multiple entries, to support the case where the certificate holder is authorized to use different sets of rich call data.

Only including "permittedValues" for "rcd" (with no "mustInclude") provides the ability to either have no "rcd" claim or only the set of constrained "permittedValues" values for an included "rcd" claim.

7. JWT Claim Constraints usage for "rcd" and "rcdi" claims

The integrity overview Section 4 of this document describes a fourth mode where both "rcdi" and JWT Claim Constraints is used. The use of this mode implies the signing of an "rcdi" claim is required to be protected by the authoritative certificate creator using JWT Claims Constraints in the certificate. The objective of the use of both of these mechanisms is to constrain the signer to construct the "rcd" and "rcdi" claims with the "rcd" jCard object including reference external content via URI. Once both the contents of the "rcd" claim and any linked content is certified by the party that is authoritative for the certificate being created and the construction of the "rcdi" claim is complete, the "rcdi" claim is linked to the STIR certificate associated with the signature in the PASSporT via JWT Claim Constraints extension as defined in [RFC8226] Section 8. It should be recognized that the "rcdi" set of digests is intended to be unique for only a specific combination of "rcd" content and URI referenced external content, and therefore provides a robust integrity mechanism for an authentication service being performed by a non-authoritative party. This would often be associated with the use of delegate certificates [RFC9060] for the signing of calls by the calling party directly as an example, even though the "authorized party" is not necessarily the subject of a STIR certificate.

For the case that there should always be both "rcd" and "rcdi" values included in the "rcd" PASSporT, the certificate JWT Claims Constraint extension MUST include both of the following:

- * a "mustInclude" for the "rcd" claim, which simply constrains the fact that an "rcd" must be included

- * a "mustInclude" for the "rcdi" claim and a "permittedValues" equal to the created "rcdi" claim value string.

Note that optionally the "rcd" claims may be included in the "permittedValues" however it is recognized that this may be redundant with the "rcdi" permittedValues because the "rcdi" digest will imply the content of the "rcd" claims themselves.

The "permittedValues" for the "rcdi" claims (or "rcd" claims more generally) may contain multiple entries, to support the case where the certificate holder is authorized to use different sets of rich call data.

8. PASSporT "crn" claim - Call Reason Definition and Usage

This specification defines a new JSON Web Token claim for "crn", Call Reason, the value of which is a single string or object that can contain information as defined in [I-D.ietf-sipcore-callinfo-rcd] corresponding to the "call-reason" parameter for the Call-Info header. This claim is optional.

Example "crn" claim with "rcd":

```
"crn" : "For your ears only",
"rcd": { "nam": "James Bond",
        "jcl": "https://example.org/james_bond.json" }
```

As also noted in [I-D.ietf-sipcore-callinfo-rcd] this claim is included as corresponding to "call-reason" Call-Info parameter, but there is an alternative suggested way to include call-reason which is to use the "cif" claim with a "call-reason" key value, as defined below in this document.

8.1. JWT Constraint for "crn" claim

The integrity of the "crn" claim can optionally be protected by the authoritative certificate creator using JWT Constraints in the certificate. If the intent of the issuer of the certificate is to always including a call reason, a "mustInclude" for the "crn" claim indicates that a "crn" claim must be present. If the issuer of the certificate wants to constrain the contents of "crn", then it may set "permittedValues" for "crn" in the certificate.

9. Rich Call Data Claims Usage Rules

Either or both the "rcd" or "crn" claims may appear in any PASSporT claims object as optional elements. The creator of a PASSporT MAY also add a "ppt" value of "rcd" to the header of a PASSporT as well, in which case the PASSporT claims MUST contain either an "rcd" or "crn" claim, and any entities verifying the PASSporT object are required to understand the "ppt" extension in order to process the PASSporT in question. An example PASSporT header with the "ppt" included is shown as follows:

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

The PASSporT claims object contains the "rcd" key with its corresponding value. The value of "rcd" is an array of JSON objects, of which one, the "nam" object, is mandatory. The key syntax of "nam" follows the display-name ABNF given in [RFC3261].

After the header and claims PASSporT objects have been constructed, their signature is generated normally per the guidance in [RFC8225].

9.1. "rcd" PASSporT Verification

An "rcd" PASSporT that uses claims defined in this specification, in order to have a successful verification outcome, MUST conform to the following:

- * have a valid signature
- * abide by all rules set forth in the proper construction of the claims
- * abide by JWT Claims Constraint rules defined in [RFC8226] Section 8 or extended in [RFC9118] if present in the certificate used to sign the PASSporT

Consistent with the verification rules of PASSporTs more generally [RFC8225], if any of the above criteria is not met, relying parties MUST NOT use any of the claims in the PASSporT.

9.2. "rcdi" Integrity Verification

If the "rcdi" claim exists, any party that dereferences a URI (i.e. downloading content for display to users) from the "rcd" claim MUST perform integrity validation of the content against the corresponding digest. Consequently, if URIs with contents covered by integrity digests are passed to another entity, the corresponding integrity digest MUST also be included, for example by passing the PASSporT. Entities that pass on the content without the URI do not have to pass on the corresponding integrity digest. An entity that does not otherwise need to dereference a URI from the "rcd" claim would be discouraged from unnecessarily dereferencing the URI solely to perform integrity verification.

If there is any issue with completing the integrity verification procedures for externally referenced content, including HTTP or HTTPS errors, the referenced content MUST be considered not verified. This SHOULD NOT however impact the result of base PASSporT verification for claims content that is directly included in the claims of the PASSporT.

9.3. Example "rcd" PASSporTs

An example of a "nam" only PASSporT claims object is shown next (with line breaks for readability only).

```
{  "orig":{"tn":"12025551000"},
   "dest":{"tn":["12025551001"]},
   "iat":1443208345,
   "rcd":{"nam":"James Bond"} }
```

An example of a "nam" and "apn" only PASSporT claims object is shown next (with line breaks for readability only).

```
{  "orig":{"tn":"12025551000"},
   "dest":{"tn":["12155551001"]},
   "iat":1443208345,
   "rcd":{"
     "apn":"12025559990",
     "nam":"Her Majesty's Secret Service" } }
```

An example of an "rcd" claims object that includes the "jcd" and also contains URI references to content which requires the inclusion of an "rcdi" claim and corresponding digests.

```

{
  "crn": "Rendezvous for Little Nellie",
  "orig": { "tn": "12025551000"},
  "dest": { "tn": ["12155551001"]},
  "iat": 1443208345,
  "rcd": {
    "jcd": ["vcard",
      [ ["version", {}, "text", "4.0"],
        ["fn", {}, "text", "Q Branch"],
        ["org", {}, "text", "MI6;Q Branch Spy Gadgets"],
        ["photo", {}, "uri", "https://example.com/photos/q-256x256.png"],
        ["logo", {}, "uri", "https://example.com/logos/mi6-256x256.jpg"],
        ["logo", {}, "uri", "https://example.com/logos/mi6-64x64.jpg"]
      ] ],
    "nam": "Q Branch Spy Gadgets"
  },
  "rcdi": {
    "/jcd/1/3/3": "sha256-RojgWwU6xUtI4q82+kHPyHm1JKbm7+663bMvzymhkl4",
    "/jcd/1/4/3": "sha256-jL4f47fF82LuwcrOrSyckA4SWrlElfARHkW6kYo1JdI",
    "/jcd/1/5/3": "sha256-GKNxxqlLRarbyBNh7hc/4lbZAdK6B0kMRf1AMRWPkSo"
  }
}

```

In an example PASSporT, where a jCard is linked via HTTPS URL using "jcl", a jCard file served at a particular URL.

An example jCard JSON file hosted at the example web address of <https://example.com/qbranch.json> is shown as follows:

```

["vcard",
  [ ["version", {}, "text", "4.0"],
    ["fn", {}, "text", "Q Branch"],
    ["org", {}, "text", "MI6;Q Branch Spy Gadgets"],
    ["photo", {}, "uri", "https://example.com/photos/q-256x256.png"],
    ["logo", {}, "uri", "https://example.com/logos/mi6-256x256.jpg"],
    ["logo", {}, "uri", "https://example.com/logos/mi6-64x64.jpg"]
  ]
]

```

For the above referenced jCard, the corresponding PASSporT claims object would be as follows:

```
{
  "crn": "Rendezvous for Little Nellie",
  "orig": {"tn": "12025551000"},
  "dest": {"tn": ["12155551001"]},
  "iat": 1443208345,
  "rcd": {
    "nam": "Q Branch Spy Gadgets",
    "jcl": "https://example.com/qbranch.json"
  },
  "rcdi": {
    "/jcl": "sha256-qCn4pEH6BJu7zXndLFuAP6DwlTv5fRmJlAFkqftwnCs",
    "/jcl/1/3/3": "sha256-RojgWwU6xUtI4q82+kHPyHm1JKbm7+663bMvzymhk14",
    "/jcl/1/4/3": "sha256-jL4f47fF82LuwcrOrSyckA4SWrlElfARHkW6kYo1JdI",
    "/jcl/1/5/3": "sha256-GKNxxq1LRarbyBNh7hc/4lbZAdK6B0kMRf1AMRWPkSo"
  }
}
```

An example "rcd" PASSporT that uses "nam" and "icn" keys with "rcdi" for calling name and referenced icon image content:

```
{
  "crn": "Rendezvous for Little Nellie",
  "orig": {"tn": "12025551000"},
  "dest": {"tn": ["12155551001"]},
  "iat": 1443208345,
  "rcd": {
    "nam": "Q Branch Spy Gadgets",
    "icn": "https://example.com/photos/q-256x256.png"
  },
  "rcdi": {
    "/nam": "sha256-sM275lTgzCte+LHOKHtU4SxG8shl0o6OS4ot8IJQImY",
    "/icn": "sha256-RojgWwU6xUtI4q82+kHPyHm1JKbm7+663bMvzymhk14"
  }
}
```

10. Compact form of "rcd" PASSporT

10.1. Compact form of the "rcd" PASSporT claim

Compact form of an "rcd" PASSporT claim has some restrictions that will be enumerated below, but mainly follows standard PASSporT compact form procedures. For re-construction of the "nam" claim the string for the display-name in the From header field. "jcl" and "jcd" MAY NOT be used with compact form due to integrity rules and URI reference rules in this specification leading to too restrictive of a set of constraints. Future specifications may revisit this to propose a consistent and comprehensive way of addressing integrity and security of information.

10.2. Compact form of the "rcdi" PASSporT claim

Compact form of an "rcdi" PASSporT claim is not supported, so if "rcdi" is required compact form MUST NOT be used.

10.3. Compact form of the "crn" PASSporT claim

Compact form of a "crn" PASSporT claim shall be re-constructed using the "call-reason" parameter of a Call-Info header as defined by [I-D.ietf-sipcore-callinfo-rcd].

11. Further Information Associated with Callers

Beyond naming information and the information that can be contained in a jCard [RFC7095] object, there may be additional human-readable information about the calling party that should be rendered to the end user in order to help the called party decide whether or not to pick up the phone. This is not limited to information about the caller, but includes information about the call itself, which may derive from analytics that determine based on call patterns or similar data if the call is likely to be one the called party wants to receive. Such data could include:

- * information related to the location of the caller, or
- * any organizations or institutions that the caller is associated with, or even categories of institutions (is this a government agency, or a bank, or what have you), or
- * hyperlinks to images, such as logos or pictures of faces, or to similar external profile information, or
- * information processed by an application before rendering it to a user, like social networking data that shows that an unknown caller is a friend-of-a-friend, or reputation scores derived from crowdsourcing, or confidence scores based on broader analytics about the caller and callee.

All of these data elements would benefit from the secure attestations provided by the STIR and PASSporT frameworks. A new IANA registry has been defined to hold potential values of the "rcd" array; see Section 17.3. Specific extensions to the "rcd" PASSporT claim are left for future specification.

There is a few ways RCD can be extended in the future, jCard is an extensible object and the key/values in the RCD claim object can also be extended. General guidance for future extensibility that were followed by the authors is that jCard generally should refer to data

that references the caller as an individual or entity, where other claims, such as "crn" refer to data regarding the specific call. There may be other considerations discovered in the future, but this logical grouping of data to the extent possible should be followed for future extensibility.

12. Third-Party Uses

While rich data about the call can be provided by an originating authentication service, an intermediary in the call path could also acquire rich call data by querying a third-party service. Such a service effectively acts as a STIR Authentication Service, generating its own PASSporT, and that PASSporT could be attached to a SIP call by either the originating or terminating side. This third-party PASSporT attests information about the calling number, rather than the call or caller itself, and as such its RCD MUST NOT be used when a call lacks a first-party PASSporT that assures verification services that the calling party number is not spoofed. It is intended to be used in cases when the originating side does not supply a display-name for the caller, so instead some entity in the call path invokes a third-party service to provide rich caller data for a call.

In telephone operations today, a third-party information service is commonly queried with the calling party's number in order to learn the name of the calling party, and potentially other helpful information could also be passed over that interface. The value of using a PASSporT to convey this information from third parties lies largely in the preservation of the third party's signature over the data, and the potential for the PASSporT to be conveyed from intermediaries to endpoint devices. Effectively, these use cases form a sub-case of out-of-band [RFC8816] use cases. The manner in which third-party services are discovered is outside the scope of this document.

An intermediary use case might look as follows: a SIP INVITE carries a display name in its From header field value and an initial PASSporT object without the "rcd" claim. When a terminating verification service implemented at a SIP proxy server receives this request, and determines that the signature is valid, it might query a third-party service that maps telephone numbers to calling party names. Upon receiving the PASSporT in a response from that third-party service, the terminating side could add a new Identity header field to the request for the "rcd" PASSporT object provided by the third-party service. It would then forward the INVITE to the terminating user agent. If the display name in the "rcd" PASSporT object matches the display name in the INVITE, then the name would presumably be rendered to the end user by the terminating user agent.

A very similar flow could be followed by an intermediary closer to the origination of the call. Presumably such a service could be implemented at an originating network in order to decouple the systems that sign for calling party numbers from the systems that provide rich data about calls.

In an alternative use case, the terminating user agent might query a third-party service. In this case, no new Identity header field would be generated, though the terminating user agent might receive a PASSporT object in return from the third-party service, and use the "rcd" field in the object as a calling name to render to users while alerting.

While in the traditional telephone network, the business relationship between calling customers and their telephone service providers is the ultimate root of information about a calling party's name, some other forms of data like crowdsourced reputation scores might derive from third parties. When those elements are present, they MUST be in a third-party "rcd" PASSporT using "iss" claim described in the next section.

12.1. Signing as a Third Party

A third-party PASSporT contains an "iss" element to distinguish its PASSporTs from first-party PASSporTs. Third-party "rcd" PASSporTs are signed with credentials that do not have authority over the identity that appears in the "orig" element of the PASSporT claims. The presence of "iss" signifies that a different category of credential is being used to sign a PASSporT than the [RFC8226] certificates used to sign STIR calls; it is instead a certificate that identifies the source of the "rcd" data. How those credentials are issued and managed is outside the scope of this specification; the value of "iss" however MUST reflect the Subject of the certificate used to sign a third-party PASSporT. The explicit mechanism for reflecting the subject field of the certificate is out of scope of this document and left to the certificate governance policies that define how to map the "iss" value in the PASSporT to the subject field in the certificate. Relying parties in STIR have always been left to make their own authorization decisions about whether to trust the signers of PASSporTs, and in the third-party case, where an entity has explicitly queried a service to acquire the PASSporT object, it may be some external trust or business relationship that induces the relying party to trust a PASSporT.

An example of a Third Party issued PASSporT claims object is as follows.

```
{  "orig":{"tn":"12025551000"},
  "dest":{"tn":["12025551001"]},
  "iat":1443208345,
  "iss":"Zorin Industries",
  "rcd":{"nam":"James St. John Smythe"} }
```

13. Levels of Assurance

As "rcd" can be provided by either first or third parties, relying parties could benefit from an additional claim that indicates the relationship of the attesting party to the caller. Even in first party cases, this admits of some complexity: the Communications Service Provider (CSP) to which a number was assigned might in turn delegate the number to a reseller, who would then sell the number to an enterprise, in which case the CSP might have little insight into the caller's name. In third party cases, a caller's name could derive from any number of data sources, on a spectrum between public data scraped from web searches to a direct business relationship to the caller. As multiple PASSporTs can be associated with the same call, potentially a verification service could receive attestations of the caller name from multiple sources, which have different levels of granularity or accuracy. Therefore, third-party PASSporTs that carry "rcd" data MUST also carry an indication of the relationship of the generator of the PASSporT to the caller in the form of the "iss" claim. As stated in the previous section, the use of "iss" MUST reflect the subject field of the certificate used to sign a third-party PASSporT to represent that relationship.

14. Using "rcd" in SIP

This section specifies SIP-specific usage for the "rcd" claim in PASSporT, and in the SIP Identity header field value. Other using protocols of PASSporT may define their own usages for the "rcd" claim.

14.1. Authentication Service Behavior

An authentication service creating a PASSporT containing an "rcd" claim MAY include a "ppt" for "rcd" or not. Third-party authentication services following the behavior in Section 12.1 MUST include a "ppt" of "rcd". If "ppt" does contain an "rcd", then any SIP authentication services MUST add a "ppt" parameter to the Identity header containing that PASSporT with a value of "rcd". The resulting Identity header might look as follows:

```
Identity: sv5CTo05KqpSmtHt3dcEiO/1CWTSZtnG3iV+1nmurLXV/HmtYNS7Ltrg9
  dlXkWzoeU7d7OV8HweTTDobV3itTmgPwCFjaEmMyEI3d7SyN21yNDo2ER/Ovgt
  w0Lu5csIppPqOgluXndzHbG7mR6Rl9BnUhHufVRbp51Mn3w0gfUs=;
  info=<https://biloxi.example.org/biloxi.cer>;alg=ES256;
  ppt="rcd"
```

This specification assumes that by default, a SIP authentication service derives the value of "rcd", specifically only for the "nam" key value, from the display-name component of the From header field value of the request, alternatively for some calls this may come from the P-Asserted-ID header. It is however a matter of authentication service policy to decide how it populates the value of "nam" key, which MAY also derive from other fields in the request, from customer profile data, or from access to external services. If the authentication service generates an "rcd" claim containing "nam" with a value that is not equivalent to the From header field display-name value, it MUST use the full form of the PASSporT object in SIP.

14.2. Verification Service Behavior

[RFC8224] Section 6.2 Step 5 requires that specifications defining "ppt" values describe any additional verifier behavior. The behavior specified for the "ppt" values of "rcd" is as follows. If the PASSporT is in compact form, then the verification service SHOULD extract the display-name from the From header field value, if any, and use that as the value for the "nam" key when it recomputes the header and claims of the PASSporT object. Additionally, if there exists a Call-Info header field as defined in [I-D.ietf-sipcore-callinfo-rcd], the "jcard" value can be derived to determine the "jcd" key when it recomputes the header and claims of the PASSporT object. If the signature validates over the recomputed object, then the verification should be considered successful.

However, if the PASSporT is in full form with a "ppt" value of "rcd", then the verification service MUST extract the value associated with the "rcd" "nam" key in the object. If the signature validates, then the verification service can use the value of the "rcd" "nam" key as the display name of calling party, which would in turn be rendered to alerted users or otherwise leveraged in accordance with local policy. This allows SIP networks that convey the display name through a field other than the From header field to interoperate with this specification. Similarly, the "jcd" or linked "jcl" jcard information and "crn" can be optionally, based on local policy for devices that support it, used to populate a Call-Info header field following the format of [I-D.ietf-sipcore-callinfo-rcd].

The third-party "rcd" PASSporT cases presents some new challenges, as an attacker could attempt to cut-and-paste such a third-party PASSporT into a SIP request in an effort to get the terminating user agent to render the display name or confidence values it contains to a call that should have no such assurance. A third-party "rcd" PASSporT provides no assurance that the calling party number has not been spoofed: if it is carried in a SIP request, for example, then some other PASSporT in another Identity header field value would have to carry a PASSporT attesting that. A verification service MUST determine that the calling party number shown in the "orig" of the "rcd" PASSporT corresponds to the calling party number of the call it has received, and that the "iat" field of the "rcd" PASSporT is within the date interval that the verification service would ordinarily accept for a PASSporT.

Verification services may alter their authorization policies for the credentials accepted to sign PASSporTs when third parties generate PASSporT objects, per Section 12.1. This may include accepting a valid signature over a PASSporT even if it is signed with a credential that does not attest authority over the identity in the "orig" claim of the PASSporT, provided that the verification service has some other reason to trust the signer. No further guidance on verification service authorization policy is given here.

The behavior of a SIP UAS upon receiving an INVITE containing a PASSporT object with an "rcd" claim largely remains a matter of implementation policy. In most cases, implementations would render this calling party name information to the user while alerting. Any user interface additions to express confidence in the veracity of this information are outside the scope of this specification.

15. Using "rcd" and "rcdi" as additional claims to other PASSporT extensions

Rich Call Data, including calling name information, as a common example, is often data that is additive to the personal communications information defined in the core PASSporT data required to support the security properties defined in [RFC8225]. For cases where the entity originating the personal communications is supporting the authentication service for the calling identity and is the authority of the Rich Call Data, rather than creating multiple Identity header fields cooresponding to multiple PASSporT extensions, the authentication service can alternatively directly add the "rcd" claim to a PASSporT that authenticates the calling identity.

Note: There is one very important caveat to this capability, because generally if there is URI referenced content in an "rcd" PASSporT there is often the requirement to use "rcdi" and JWT Claims

Constraints. So, it is important for the user of this specification to recognize that the certificates used should include the necessary JWT Claims Constraints for proper integrity and security of the values in the "rcd" claim incorporated into PASSporTs that are not "rcd".

15.1. Procedures for applying "rcd" as claims only

For a given PASSporT using some other extension than "rcd", the Authentication Service MAY additionally include the "rcd" claim as defined in this document. This would result in a set of claims that correspond to the original intended extension with the addition of the "rcd" claim.

The Verification service that receives the PASSporT, if it supports this specification and chooses to, should interpret the "rcd" claim as simply just an additional claim intended to deliver and/or validate delivered Rich Call Data.

15.2. Example for applying "rcd" as claims only

In the case of [RFC8588] which is the PASSporT extension supporting the SHAKEN specification [ATIS-1000074.v002], a common case for an Authentication service to co-exist in a CSP network along with the authority over the calling name used for the call. Rather than require two identity headers, the CSP Authentication Service can apply both the SHAKEN PASSporT claims and extension and simply add the "rcd" required claims defined in this document.

For example, the PASSporT claims for the "shaken" PASSporT with "rcd" claims would be as follows:

Protected Header

```
{
  "alg":"ES256",
  "typ":"passport",
  ppt:shaken,
  "x5u":"https://cert.example.org/passport.cer"
}
```

Payload

```
{
  attest:A,
  "dest":{"tn":["12025551001"]},
  "iat":1443208345,
  "orig":{"tn":"12025551000"},
  origid:123e4567-e89b-12d3-a456-426655440000,
  "rcd":{"nam":"James Bond"}
}
```

A Verification Service that supports "rcd" and "shaken" PASSport extensions is able to receive the above PASSport and interpret both the "shaken" claims as well as the "rcd" defined claim.

If the Verification Service only understands the "shaken" PASSport extension claims and doesn't support "rcd" PASSport extension, then the "rcd" claim is used during PASSport signature validation but is otherwise ignored and disregarded.

16. Acknowledgements

We would like to thank David Hancock, Robert Sparks, Russ Housley, Eric Burger, Alec Fenichel, Ben Campbell, Jack Rickard, Jordan Simpson for helpful suggestions, review, and comments.

17. IANA Considerations

17.1. JSON Web Token Claim

This specification requests that the IANA add three new claims to the JSON Web Token Claims registry as defined in [RFC7519].

Claim Name: "rcd"

Claim Description: Rich Call Data Information

Change Controller: IESG

Specification Document(s): [RFCThis]

Claim Name: "rcdi"

Claim Description: Rich Call Data Integrity Information

Change Controller: IESG

Specification Document(s): [RFCThis]

Claim Name: "crn"

Claim Description: Call Reason

Change Controller: IESG

Specification Document(s): [RFCThis]

17.2. PASSporT Types

This specification requests that the IANA add a new entry to the PASSporT Types registry for the type "rcd" which is specified in [RFCThis].

17.3. PASSporT RCD Types

This document requests that the IANA create a new registry for PASSporT RCD types. Registration of new PASSporT RCD types shall be under the Specification Required policy.

This registry is to be initially populated with four values, "nam", "apn", "jcd", and "jcl", which are specified in [RFCThis].

18. Security Considerations

Whether its identities, alternate identities, images, logos, physical addresses, all of the information contained in a RCD PASSporT must follow some form of vetting in which the authoritative entity or user of the information being signed MUST follow an applicable policy of the eco-system using RCD. This can be of many forms, depending on the setup and constraints of the eco-system so is therefore out-of-scope of this document. However, the general chain of trust that signers of RCD PASSporT are either directly authoritative or have been delegated authority through certificates using JWT Claim Constraints and integrity mechanisms defined in this and related documents is critical to maintain the integrity of the eco-system utilizing this and other STIR related specifications.

Revealing information such as the name, location, and affiliation of a person necessarily entails certain privacy risks. Baseline PASSporT has no particular confidentiality requirement, as the information it signs over in a using protocol like SIP is all information that SIP carries in the clear anyway. Transport-level security can hide those SIP fields from eavesdroppers, and the same confidentiality mechanisms would protect any PASSporT(s) carried in SIP.

The use of JWTClaimConstraints, a mechanism defined in [RFC8226] and extended in [RFC9118] to constrain any of the RCD information in the public certificate by including that information in the certificate, depending on the availability in the deployment of the PKI system, may present a privacy issue. The use of "rcdi" claim and digests for representing JWT claim contents may be a recommended way of preventing the exposure of that information through the certificates which are often publically accessible and available.

Since computation of "rcdi" digests for URIs requires the loading of referenced content, it would be best practice to validate that content at the creation of the "rcdi" or corresponding JWT claim constraint value by checking for content that may cause issues for verification services or that doesn't follow the behavior defined in this document, e.g., unreasonably sized data, the inclusion of recursive URI references, etc. Along the same lines, the verification service should also use precautionary best practices to avoid attacks when accessing URI linked content.

18.1. The use of JWT Claim Constraints in delegate certificates to exclude unauthorized claims

While this can apply to any PASSporT that is signed with a STIR Delegate Certificates [RFC9060], it is important to note that when constraining PASSporTs to include specific claims or contents of claims, it is also important to consider potential attacks by non-authorized signers that may include other potential PASSporT claims that weren't originally vetted by the authorized entity providing the delegate certificate. The use of JWT claims constraints as defined in [RFC9118] for preventing the ability to include claims beyond the claims defined in this document may need to be considered.

Certificate issuers SHOULD NOT include an entry in mustExclude for the "rcdi" claim for a certificate that will be used with the PASSporT Extension for Rich Call Data defined in this document. Excluding this claim would prevent the integrity protection mechanism from working properly.

19. References

19.1. Normative References

- [I-D.ietf-sipcore-callinfo-rcd]
Wendt, C. and J. Peterson, "SIP Call-Info Parameters for Rich Call Data", Work in Progress, Internet-Draft, draft-ietf-sipcore-callinfo-rcd-04, 7 March 2022,
<<https://www.ietf.org/archive/id/draft-ietf-sipcore-callinfo-rcd-04.txt>>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997,
<<https://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, <<https://www.rfc-editor.org/info/rfc3261>>.
- [RFC3325] Jennings, C., Peterson, J., and M. Watson, "Private Extensions to the Session Initiation Protocol (SIP) for Asserted Identity within Trusted Networks", RFC 3325, DOI 10.17487/RFC3325, November 2002, <<https://www.rfc-editor.org/info/rfc3325>>.
- [RFC4627] Crockford, D., "The application/json Media Type for JavaScript Object Notation (JSON)", RFC 4627, DOI 10.17487/RFC4627, July 2006, <<https://www.rfc-editor.org/info/rfc4627>>.
- [RFC4648] Josefsson, S., "The Base16, Base32, and Base64 Data Encodings", RFC 4648, DOI 10.17487/RFC4648, October 2006, <<https://www.rfc-editor.org/info/rfc4648>>.
- [RFC6901] Bryan, P., Ed., Zyp, K., and M. Nottingham, Ed., "JavaScript Object Notation (JSON) Pointer", RFC 6901, DOI 10.17487/RFC6901, April 2013, <<https://www.rfc-editor.org/info/rfc6901>>.
- [RFC7095] Kewisch, P., "jCard: The JSON Format for vCard", RFC 7095, DOI 10.17487/RFC7095, January 2014, <<https://www.rfc-editor.org/info/rfc7095>>.
- [RFC7340] Peterson, J., Schulzrinne, H., and H. Tschofenig, "Secure Telephone Identity Problem Statement and Requirements", RFC 7340, DOI 10.17487/RFC7340, September 2014, <<https://www.rfc-editor.org/info/rfc7340>>.
- [RFC7519] Jones, M., Bradley, J., and N. Sakimura, "JSON Web Token (JWT)", RFC 7519, DOI 10.17487/RFC7519, May 2015, <<https://www.rfc-editor.org/info/rfc7519>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8224] Peterson, J., Jennings, C., Rescorla, E., and C. Wendt, "Authenticated Identity Management in the Session Initiation Protocol (SIP)", RFC 8224, DOI 10.17487/RFC8224, February 2018, <<https://www.rfc-editor.org/info/rfc8224>>.

- [RFC8225] Wendt, C. and J. Peterson, "PASSport: Personal Assertion Token", RFC 8225, DOI 10.17487/RFC8225, February 2018, <<https://www.rfc-editor.org/info/rfc8225>>.
- [RFC8226] Peterson, J. and S. Turner, "Secure Telephone Identity Credentials: Certificates", RFC 8226, DOI 10.17487/RFC8226, February 2018, <<https://www.rfc-editor.org/info/rfc8226>>.
- [RFC8588] Wendt, C. and M. Barnes, "Personal Assertion Token (PaSSporT) Extension for Signature-based Handling of Asserted information using toKENs (SHAKEN)", RFC 8588, DOI 10.17487/RFC8588, May 2019, <<https://www.rfc-editor.org/info/rfc8588>>.
- [RFC9060] Peterson, J., "Secure Telephone Identity Revisited (STIR) Certificate Delegation", RFC 9060, DOI 10.17487/RFC9060, September 2021, <<https://www.rfc-editor.org/info/rfc9060>>.
- [RFC9118] Housley, R., "Enhanced JSON Web Token (JWT) Claim Constraints for Secure Telephone Identity Revisited (STIR) Certificates", RFC 9118, DOI 10.17487/RFC9118, August 2021, <<https://www.rfc-editor.org/info/rfc9118>>.

19.2. Informative References

- [ATIS-1000074.v002]
ATIS/SIP Forum NNI Task Group, "Signature-based Handling of Asserted information using toKENs (SHAKEN) <https://access.atis.org/apps/group_public/download.php/62391/ATIS-1000074.v002.pdf>", November 2021.
- [RFC8816] Rescorla, E. and J. Peterson, "Secure Telephone Identity Revisited (STIR) Out-of-Band Architecture and Use Cases", RFC 8816, DOI 10.17487/RFC8816, February 2021, <<https://www.rfc-editor.org/info/rfc8816>>.

Authors' Addresses

Chris Wendt
Somos Inc.
Email: chris-ietf@chriswendt.net

Jon Peterson
Neustar Inc.
Email: jon.peterson@neustar.biz

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 12, 2019

J. Peterson
Neustar
March 11, 2019

STIR Certificate Delegation
draft-peterson-stir-cert-delegation-00.txt

Abstract

The Secure Telephone Identity Revisited (STIR) certificate profile provides a way to attest authority over telephone numbers and related identifiers for the purpose of preventing telephone number spoofing. This specification details how that authority can be delegated from a parent certificate to a subordinate certificate, in cases where service providers grant credentials to enterprises or other customers capable of signing calls with STIR.

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 <https://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 September 12, 2019.

Copyright Notice

Copyright (c) 2019 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 (<https://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	3
3. Delegation of STIR Certificates	3
3.1. Authentication Services Signing with Delegate Certificates	4
3.2. Verification Service Behavior for Delegate Certificate Signatures	4
4. Acquiring Certificate Chains in STIR	5
5. ACME and Delegation	5
6. IANA Considerations	6
7. Privacy Considerations	6
8. Security Considerations	6
9. Acknowledgments	6
10. References	6
10.1. Normative References	6
10.2. Informative References	7
Author's Address	9

1. Introduction

The STIR problem statement [RFC7340] reviews the difficulties facing the telephone network that are enabled by impersonation, including various forms of robocalling, voicemail hacking, and swatting. One of the most important components of a system to prevent impersonation is the implementation of credentials which identify the parties who control telephone numbers. The STIR certificates [RFC8226] specification describes a credential system based on [X.509] version 3 certificates in accordance with [RFC5280] for that purpose. Those credentials can then be used by STIR authentication services [RFC8224] to sign PASSport objects [RFC8225] carried in SIP [RFC3261] requests.

[RFC8226] specifies an extension to X.509 that defines a Telephony Number (TN) Authorization List that may be included by certificate authorities in certificates. This extension provides additional information that relying parties can use when validating transactions with the certificate. When a SIP request, for example, arrives at a terminating administrative domain, the calling number attested by the SIP request can be compared to the TN Authorization List of the certificate that signed the PASSport to determine if the caller is authorized to use that calling number.

Initial deployment of [RFC8226] has focused on the use of Service Provider Codes (SPCs) to attest the scope of authority of a certificate. Typically, these codes are internal telephone network identifiers such as the Operating Company Numbers (OCNs) assigned to carriers in the United States. Allocations at finer levels of granularity, to blocks of telephone numbers or even to individual numbers, are also desirable for enterprise use cases. [RFC8226] gave an overview of a certificate enrollment model based on "delegation," whereby the holder of certificate might allocate a subset of that certificates authority to another party. This specification details how delegation of authority works for STIR certificates.

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. Delegation of STIR Certificates

STIR delegate certificates are certificates containing a TNAuthList object that have been signed with the private key of a parent certificate that itself contains a TNAuthList object. The parent certificate needs to have its CA boolean set to "true", indicating that that it can sign certificates. Every STIR delegate certificate identifies its parent certificate with a standard [RFC5280] Authority Key Identifier.

The authority bestowed on the holder of the delegate certificate by the parent certificate is recorded in the delegate certificate's TNAuthList. Because STIR certificates use the TNAuthList object rather than the Subject Name for indicating the scope of their authority, traditional [RFC5280] name constraints are not directly applicable to STIR. In a manner similar to the RPKI [RFC6480] "encompassing" semantics, each delegate certificate must have a TNAuthList scope that is equal to or a subset of its parent certificate's scope: it must be "encompassed." For example, a parent certificate with a TNAuthList that attested authority for the numbering range +1-212-555-1000 through 1999 could issue a certificate to one delegate attesting authority for the range +1-212-555-1500 through 1599, and to another delegate a certificate for the individual number +1-212-555-1824.

Delegate certificates may themselves be issued with the CA boolean set to "true" so that they can serve as parent certificates to further delegates; effectively, this delegate certificate is a cross-certificate, as its issuer is not the same as its subject. In the

STIR ecosystem, certification authority certificates may be used to sign PASSporTs; this removes the need for creating a redundant end-entity certificate with an identical TNAuthList to its parent, though if for operational or security reasons certificate holders wish to do so, they may.

Parent certificates may have a TNAuthList containing one or more SPCs, one or more telephone number ranges, or both. Delegations from a parent certificate that contains only SPCs to a delegate certificate containing a telephone number or number range are permitted. Ascertaining whether or not a given telephone number belongs to the service provider identified by an SPC requires access to industry numbering databases that are outside the scope of this specification; entities that are constructing a certificate path who have access to those resources can validate those delegations.

3.1. Authentication Services Signing with Delegate Certificates

Authentication service behavior for delegate certificates is little changed from baseline STIR behavior. The same checks are performed by the authentication service, comparing the calling party number attested in call signaling with the scope of the authority of the signing certificate. Authentication services SHOULD NOT use a delegate certificate without validating that its scope of authority is encompassed by that of its parent certificate, and if that certificate in turn has its own parent, the entire certificate path should be validated.

Note that authentication services creating a PASSporT for a call signed with a delegate certificate MUST provide an "x5u" link corresponding to the entire certificate chain, rather than just the delegate certificate used to sign the call, as described in Section 4.

3.2. Verification Service Behavior for Delegate Certificate Signatures

The responsibility of a verification service validating PASSporTs signed with delegate certificates, while largely following baseline [RFC8224] and [RFC8225], requires some additional procedures. When the verification service dereferences the "x5u" parameter, it will acquire a certificate list rather than a single certificate. It MUST then validate all of the credentials in the list, identifying the parent certificate for each delegate through its AKID object.

While ordinarily, relying parties have significant latitude in path construction when validating a certificate chain, STIR assumes a more rigid hierarchical subordination model, rather than one where relying parties may want to derive their own chains to particular trust

anchors. If the certificate chain acquired from the "x5u" element of a PASSporT does not lead to an anchor that the verification service trusts, it treats the validation no differently than it would when a non-delegated certificate was issued by an untrust root; in SIP, it MAY return a 437 "Unsupported Credential" response if the call should be failed for lack of a valid Identity header.

4. Acquiring Certificate Chains in STIR

PASSporT [RFC8225] uses the "x5u" element to convey the URL where verification services can acquire the certificate used to sign a PASSporT. This value is mirrored by the "info" parameter of the Identity header when a PASSporT is conveyed via SIP. Commonly, this is an HTTPS URI.

When a STIR delegate certificate is used to sign a PASSporT, the "x5u" element in the PASSporT will contain a URI indicating where a certificate list is available. That list will be a concatenation of PEM encoded certificates of the type "application/pem-certificate-chain" defined in [I-D.ietf-acme-acme]. The list begins with the certificate used to sign the PASSporT, followed by its parent, and then any subsequent grandparents, great-grandparents, and so on. The ordering MUST conform to the AKID/SKID order chain encoded in the certs themselves. Note that ACME requires the first element in a pem-certificate-chain to be an end-entity certificate; STIR relaxes this requirement, as CA certificates are permitted to sign PASSporTs, so the first element in a pem-certificate-chain used for STIR MAY be a CA certificate.

5. ACME and Delegation

STIR deployments commonly use ACME [I-D.ietf-acme-acme] for certificate acquisition, and it is anticipated that delegate certificates as well will be acquired through an ACME interface. An entity that wishes to acquire a certificate from a particular CA will request an Authority Token [I-D.ietf-acme-authority-token] from the parent with the desired TNAuthList [I-D.ietf-acme-authority-token-tnauthlist] object. Note that if the client wishes to do further subdelegation of its own, it should request a token with the "ca" Authority Token flag set.

The entity then presents that Authority Token to a certificate authority to acquire a STIR delegate certificate. ACME returns an "application/pem-certificate-chain" object with suitable for publishing as an HTTPS resource for retrieval with the PASSporT "x5u" mechanism as discussed in Section 4. If the CSR presented to the ACME server is for a certificate with the CA boolean set to "true", then the ACME server makes a policy decision to determine whether or

not it is appropriate to issue that certificate to the requesting entity. In most ACME cases, that policy decision will be made based on the "ca" flag in the Authority Token.

6. IANA Considerations

This document contains no actions for the IANA.

7. Privacy Considerations

[TBD.]

8. Security Considerations

This document is entirely about security. For further information on certificate security and practices, see [RFC5280], in particular its Security Considerations.

9. Acknowledgments

We would like to thank Richard Barnes, Chris Wendt, Dave Hancock, Russ Housley, and Sean Turner for key input to the discussions leading to this document.

10. References

10.1. Normative References

[I-D.ietf-acme-acme]

Barnes, R., Hoffman-Andrews, J., McCarney, D., and J. Kasten, "Automatic Certificate Management Environment (ACME)", draft-ietf-acme-acme-18 (work in progress), December 2018.

[I-D.ietf-acme-authority-token]

Peterson, J., Barnes, M., Hancock, D., and C. Wendt, "ACME Challenges Using an Authority Token", draft-ietf-acme-authority-token-01 (work in progress), October 2018.

[I-D.ietf-acme-authority-token-tnauthlist]

Wendt, C., Hancock, D., Barnes, M., and J. Peterson, "TNAuthList profile of ACME Authority Token", draft-ietf-acme-authority-token-tnauthlist-01 (work in progress), October 2018.

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://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, <<https://www.rfc-editor.org/info/rfc3261>>.
- [RFC7044] Barnes, M., Audet, F., Schubert, S., van Elburg, J., and C. Holmberg, "An Extension to the Session Initiation Protocol (SIP) for Request History Information", RFC 7044, DOI 10.17487/RFC7044, February 2014, <<https://www.rfc-editor.org/info/rfc7044>>.
- [RFC7519] Jones, M., Bradley, J., and N. Sakimura, "JSON Web Token (JWT)", RFC 7519, DOI 10.17487/RFC7519, May 2015, <<https://www.rfc-editor.org/info/rfc7519>>.
- [RFC8224] Peterson, J., Jennings, C., Rescorla, E., and C. Wendt, "Authenticated Identity Management in the Session Initiation Protocol (SIP)", RFC 8224, DOI 10.17487/RFC8224, February 2018, <<https://www.rfc-editor.org/info/rfc8224>>.
- [RFC8225] Wendt, C. and J. Peterson, "PASSporT: Personal Assertion Token", RFC 8225, DOI 10.17487/RFC8225, February 2018, <<https://www.rfc-editor.org/info/rfc8225>>.
- [RFC8226] Peterson, J. and S. Turner, "Secure Telephone Identity Credentials: Certificates", RFC 8226, DOI 10.17487/RFC8226, February 2018, <<https://www.rfc-editor.org/info/rfc8226>>.

10.2. Informative References

- [ATIS-0300251] ATIS Recommendation 0300251, "Codes for Identification of Service Providers for Information Exchange", 2007.
- [DSS] National Institute of Standards and Technology, U.S. Department of Commerce | NIST FIPS PUB 186-4, "Digital Signature Standard, version 4", 2013.

- [RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, DOI 10.17487/RFC5280, May 2008, <<https://www.rfc-editor.org/info/rfc5280>>.
- [RFC5806] Levy, S. and M. Mohali, Ed., "Diversion Indication in SIP", RFC 5806, DOI 10.17487/RFC5806, March 2010, <<https://www.rfc-editor.org/info/rfc5806>>.
- [RFC6480] Lepinski, M. and S. Kent, "An Infrastructure to Support Secure Internet Routing", RFC 6480, DOI 10.17487/RFC6480, February 2012, <<https://www.rfc-editor.org/info/rfc6480>>.
- [RFC7340] Peterson, J., Schulzrinne, H., and H. Tschofenig, "Secure Telephone Identity Problem Statement and Requirements", RFC 7340, DOI 10.17487/RFC7340, September 2014, <<https://www.rfc-editor.org/info/rfc7340>>.
- [X.509] ITU-T Recommendation X.509 (10/2012) | ISO/IEC 9594-8, "Information technology - Open Systems Interconnection - The Directory: Public-key and attribute certificate frameworks", 2012.
- [X.520] ITU-T Recommendation X.520 (10/2012) | ISO/IEC 9594-6, "Information technology - Open Systems Interconnection - The Directory: Selected Attribute Types", 2012.
- [X.680] ITU-T Recommendation X.680 (08/2015) | ISO/IEC 8824-1, "Information Technology - Abstract Syntax Notation One: Specification of basic notation".
- [X.681] ITU-T Recommendation X.681 (08/2015) | ISO/IEC 8824-2, "Information Technology - Abstract Syntax Notation One: Information Object Specification".
- [X.682] ITU-T Recommendation X.682 (08/2015) | ISO/IEC 8824-2, "Information Technology - Abstract Syntax Notation One: Constraint Specification".
- [X.683] ITU-T Recommendation X.683 (08/2015) | ISO/IEC 8824-3, "Information Technology - Abstract Syntax Notation One: Parameterization of ASN.1 Specifications".

Author's Address

Jon Peterson
Neustar, Inc.

Email: jon.peterson@team.neustar