STIR
Internet-Draft
Intended status: Standards Track
Expires: September 6, 2018

E. Burger
Georgetown University
March 5, 2018

Registry for Country-Specific Secure Telephone Identity (STIR) Root Certificates
draft-burger-stir-iana-cert-00

Abstract

This document defines an IANA registry that maps country codes to secure telephone identity (STIR) root certificates authorized to create signing certificates for telephone numbers under the authority of a given country. Some countries allow carriers to block unsolicited, automatically generated nuisance calls commonly known as 'robocalls.' The use of signed STIR tokens in the Session Initiation Protocol (SIP) may be useful in such scenarios to provide positive attestations as to call origin. Legacy telephone numbering resources are administrated by national policy. Unlike the market-driven use case of Web commerce, some nations may restrict the list of STIR root certificate authorities acceptable for issuing signing certificates for STIR tokens that provide attestations for their local legacy telephone numbering resources. The registry described in this document enables call recipients in a first country to validate that signaling it receives from a caller with a telephone number claiming to be in a second country conforms to the second country's policy of (1) having a limited list of STIR root certificate authorities (or not) and (2) the certificate that produced the signature over the signaling is signed by one of those authorized STIR root certificate authorities.

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 6, 2018.

Copyright Notice

Copyright (c) 2018 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.

1. Introduction

One problem that plagues some communications applications is where the caller deliberately misrepresents their identity with the intent to defraud, cause harm, or wrongfully obtain anything of value. The IETF Secure Telephone Identity Revisited (STIR) work group has developed a series of RFCs specifying the mechanisms for cryptographically signing the asserted identity and other elements in Session Initiation Protocol (SIP) [RFC3261] messages. One kind of identity used in SIP is a telephone number [E.164]. A telephone number is a string of digits, where the first one to three digits indicate a country code. The International Telecommunications Union - Telecommunications Sector (ITU-T) defines country codes and delegates the authority for numbers under a country code to the respective national communications authority for that country, as listed in E.164 Annex D [E.164D].

Section 7 of Authenticated Identity Management in the Session Initiation Protocol [RFC8224] describes the process for signing identity tokens. Correspondingly, the STIR Certificates document [RFC8226] describes the format of the signing certificate. The protocol and formats are independent of and can have uses beyond that of signing originating telephone numbers. As well, given that for the most part governments are responsible for managing the numbering resources within their country code, governmental policy may impact who is authorized to issue signing certificates and what constitutes a valid signing chain. As such, the base STIR documents defer certificate and validation policy to other documents. This document describes a registry for finding the appropriate STIR root certificate authority for a given country code for signed telephone numbers. This document neither implies nor endorses any policies for

non-E.164 number identity assertions, such as arbitrary SIP URI's. Moreover, while this document describes the STIR root certificate registry for various nation's STIR root certificates, it does not mandate any particular policy regime.

Recalling the STIR problem statement [RFC7340], the goal is to provide authenticated identity for the caller. When a SIP endpoint receives a message with a signed STIR token, that endpoint needs to know whether the signing certificate is, in fact, allowed to make assertions for that identity. It does us no good for a caller with ill intent to have a signed assertion that has a valid certificate chain to an unauthorized root. Likewise, it does us no good to use self-signed certificates to sign a SIP message, as even with some limited verification, if there is the slightest chance of an entity with nefarious intent to succeed in either spoofing or taking over the identify of a caller, experience has shown they will do so.

As mentioned above, telephone numbers are assigned by the ITU-T to national communications authorities responsible for the number space below the numeric country code. A national regulator can inform service providers under its authority which root certificate authorities are authoritative for numbers under its control. This is straightforward within a country. However, this does not work for the global, interconnected communications network. When someone in a first country calls someone in a second country, how is the service provider or end user in the second country to know who is authoritative for signing certificates in the first country?

To solve this problem, this document establishes an IANA registry of STIR root certificate authorities, indexed by country. This document also establishes an IANA registry of numeric country codes to ISO 3166-1 [ISO.3166-1.2013] alpha-2 country codes.

2. Data Model

2.1. Country Code Registry

The ITU-T publishes a list of assigned numeric country codes in E.164 Annex D [E.164D]. The International Standards Organization (ISO) publishes a list of two-character country codes in ISO 3166-1 [ISO.3166-1.2013]. The Country Code Registry maps the telephone country codes to two-letter country codes. From here on, this document refers to the former as "numeric country codes" and the latter as "ISO country codes".

Applications are expected to do a longest-match search to find the ISO country code corresponding to a numeric country code. This enables overlapping numeric country codes such as for +1 and +7. Let

us say an enclosing numeric country code, such as +7 for the Russian Federation, will specify the certificates of an enclosed numeric country code, such as +76 for Kazakhstan. It also enables overlapping countries to provide their own, distinct set of roots for the enclosed numeric country code or to specify they are not specifying any STIR root certificates.

2.2. STIR Root Certificate Registry

This registry maps ISO country codes to STIR root certificates. There can be one or more STIR root certificates per ISO country code.

2.3. Operation

If a country is participating, it ensures it has the appropriate mapping from numeric country code to ISO country code in the Country Code Registry. Then, if the country does have STIR root certificate(s) to list, it places them in the STIR Root Certificate Registry. If the country wants to indicate that it is not specifying STIR root certificates, it creates an entry in the Country Code Registry but has no entries in the STIR Root Certificate Registry.

Besides directly indicating non-participation, this model enables handling of overlapping country codes.

Take the case of an overlapping numeric country code where the enclosed numbering country uses the same roots as the enclosing numbering country. The enclosed numbering country refrains from making an entry in the Country Code Registry. For example, let us say Kazakhstan uses the same STIR root certificates as the Russian Federation. We would expect to see

Numeric	ISO
7	RF

in the Country Code Registry and

ISO	Certificate
RF	[STIR public root certificate]

in the STIR Root Certificate Registry. Calls to +76 and +77 will match +7 in the Country Codes Registry, which maps to the string RF, which maps to the shared STIR root certificate.

Take the case where Kazakhstan uses a different certificate than the Russian Federation. Then we would expect to see

Numeric	ISO
7	RF
76	KZ
77	KZ

in the Country Code Registry and

ISO	Certificate
RF	[RF's STIR public root certificate]
KZ	[KZ's STIR public root certificate]

in the STIR Root Certificate Registry.

Finally, take the case the Russian Federation specifies authorized STIR root certificate authorities, but Kazakhstan does not. Then we would see

Numeric	ISO
7	RF
76	KZ
77	KZ

in the Country Code Registry and

ISO	Certificate
RF	[RF's STIR public root certificate]

in the STIR Root Certificate Registry. Here, calls from Kazakhstan would match the +76 mapping, but applications will notice there are no KZ STIR root certificate authorities in the STIR Root Certificates Registry.

The registry indicates multiple STIR root certificate authorities by having multiple entities with the same ISO country code and different STIR root certificates in the STIR Root Certificates Registry. For example,

Numeric	ISO
1	US

in the Country Code Registry and

ISO	Certificate
US	[US STIR public root certificate authority A]
US	[US STIR public root certificate authority Z]

in the STIR Root Certificate Registry.

3. Registry Elements

3.1. Numeric Country Code

E.164 [E.164] defines the country code as a one- to three-digit string. However, there are some country codes that have different country delegations beyond the country code. For example, footnote b of E.164 Annex D [E.164D] shows 25 countries under country code +1 and two countries under country code +7. As well, country code +881, for satellite services, and codes +882 and +883, for international networks, are under the jurisdiction of various national authorities.

To distinguish the various national authorities under a given country code, the country code entry can contain these identity codes.

Currently, the longest entry can be seven digits, but this could change in the future.

Applications using this registry to find the ISO country code for a given numeric country code (and identity codes) use the longest match in the registry. A potential error condition would be if a country has not designated a mapping in the registry and another country with a shorter, overlapping numeric country code string does have a mapping. At the time of this writing, this is only possible for the overlapping country codes of +1 and +7 as well as the special use codes +881, +882, and +883.

Unfortunately, there is no easy algorithm or pattern to the identity digits (area codes) in country code +1. As of the time of the writing this document, the North American Numbering Plan Administrator (NANPA) reports that the United States has about 275 area codes assigned (including free phone and local number portability routing), Canada has 65 area codes assigned, and the various Caribbean nations have 1-4 area codes assigned each [NPAreport]. As a further complication, the freephone number space, such as +1800 and +1888, is also shared. Some countries have exclusive responsibility for some 800 number prefixes, such as +1800389 for the Bahamas and +1800271 for Trinidad.

3.2. STIR Root CA Public Key

Each country can have zero or more STIR root certificate authorities. The STIR root certificate authority is the trust anchor for STIR (SIP) PKI in the given jurisdiction. The expectation is the authority for signing the identity of a caller will be much stricter than the authority for signing the identity of, for example, a Web site. In the common Web browser situation, a Web server operator can purchase a certificate issued by one of hundreds of certificate authorities from anywhere in the world. To ensure interoperability, browser and operating system manufacturers need to include the STIR root certificates from those certificate authorities so when a user in one part of the world accesses a Web server in another part of the world that has a certificate issued by a certificate authority in yet a different part of the world, the site will validate. In the telephone number identity situation, it is expected that for the most part the individual national numbering authorities will choose a very limited set of STIR root certificate authorities who will be allowed to issue signing certificates for numbers assigned to that country.

Within a single country, it would be a relatively easy matter for the national communications regulator to impose and inform their domestic service providers who is the designated certificate authority within that country. However, given the large amount of international

telephone traffic (as an example, there were over 100,000,000,000 minutes of traffic between the U.S. and other countries in 2014, including VoIP [FCC_intl]), there is a need for service providers and users in different countries to validate that one of the proper certificate authorities for that country has issued the signing certificate.

The entry for each national STIR root certificate authority is a P7B certificate [RFC2315] that contains the public key of the STIR root certificate authority, matching the private key the STIR root certificate authority uses to sign signing keys used by its delegates, such as telecommunications service providers.

Countries that are not participating in STIR but want to avoid the shortest-match issue raised above can create an entry in the Country Code registry with no entry in the STIR Root Certificate registry.

4. Terminology

This document uses the terms "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" as RFC 2119 [RFC2119] defines them.

5. IANA Considerations

Refer to [RFC8126] for a description of IANA Considerations terms and their meanings.

5.1. Registry Policy: Expert Review

This registry is Expert Review with registry-based delegation. The integrity of a given nation’s numbering system is generally the purview of the respective national government. We do not anticipate IANA to intervene in disputes of who has the authority for entering and changing STIR root certificates. In general, IANA SHOULD validate the request is related to the recognized national authority for the country as specified in [ITU-D.Agencies], unless it is not clear who the national authority is.

> TO DO: Instead of using the RAI list, should we setup a dedicated list for dispute resolution?

5.2. Appealing Registry Decisions

IANA makes decisions based on expertise as well as guidance from the community. If a member of the community has a concern with an individual decision made by IANA with regard to the registry, the individual shall proceed as follows:

1. Attempt to resolve the concern directly with IANA.

2. If a resolution cannot be reached directly with IANA, express the concern to the community and attempt to achieve rough consensus regarding a resolution on the RAI list. The Area Directors of the IETF Real-time Applications and Infrastructure Areamay, at their discretion, attempt to guide the members of the community to rough consensus.

3. As a last resort, if a resolution cannot be reached on the RAI mailing list, appeal to the IESG for a resolution. The appellant must show that the decision made by IANA (a) was materially in error and (b) has caused material harm. In its deliberations regarding an appeal, the IESG shall weigh all the evidence presented to it and use its best judgment in determining a resolution.

5.3. Registry Elements

The STIR Root Certificate registry consists of one or more entities indicating the public keys of STIR root certificate authorities for a given country code. With around 200 countries, each of which might have one to four STIR root certificate authorities, results in a registry with a total participation of about one thousand entries. The expectation is there would be substantially fewer entries in practice.

5.3.1. Numeric Country Code

The numeric country code is a one- to eight-digit string indicating the numeric country code and optional identity digits. Identity digits are often known as an area code or city code. [E.164D] lists country codes and the identity digits when there are overlapping country codes (+1, +7, and some international codes).

IANA MUST verify the requested mapping includes a valid numeric country code as specified in E.164 Annex D.

NOTE: The conventional leading + to indicate the string identifies a country code is NOT part of the Country Code element in the registry.

5.3.2. ISO Country Code

The ISO country code is a two-character string drawn from ISO 3166-1 alpha-2 [ISO.3166-1.2013].

IANA should verify the requested mapping includes a valid two-digit country code appropriate for the requested numeric country code,

subject to the understanding that a country's numeric country code may map to an enclosing ISO country code if there is no longer match in the Country Code Registry. IANA MAY verify whether there is a need to place entries for enclosed numeric country codes if an enclosing Country Code mapping is established. This is only an issue for numeric country codes in +1, +7, +881, +882, and +883 at the time of this writing.

5.3.3. STIR Root Certificate

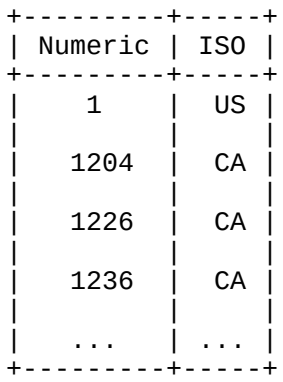
The STIR root certificate is a P7B file [RFC2315] that contains the public key of the authorized STIR root certificate that signs the certificates authorized to sign STIR signaling in the given country. There can be one or more entries in the registry for a given ISO country code to allow for multiple STIR root certificate authorities for a given country.

IANA MUST verify the certificate is valid.

5.4. Other IANA Considerations

The expectation is the relevant national authorities or their designates will keep IANA informed on updates to things such as numbering plans. This is most prominently an issue in numeric country code +1, where the numbering administrator often assigns new area codes, which could end up in different countries. Specifically, IANA has no obligation to monitor the ITU-T, North American Numbering Plan Administrator (NANPA), or other entity to keep the Country Code Registry up to date. It should be noted there is a single NANPA for the entire +1 numeric country code.

At the time of this writing, we expect both the United States and Canada to be specifying a limited set of STIR root certificate authorities. The most difficult overlap set is the overlap between Canada and the United States in the numeric country code list. As a convenience to the community we request IANA pre-populate the Country Code Registry with +1 mapped to the string US and to pre-populate the Country Code Registry with the area codes assigned to Canada with the string CA, as found in the authoritative listing of +1 area code assignments [NPAreport]. As an example, but not necessarily the normative entries:



Numeric	ISO
1	US
1204	CA
1226	CA
1236	CA
...	...

6. Security Considerations

The choice of having the STIR root certificate stored by IANA means that users accessing the certificates MUST use a source-authenticated retrieval mechanism, such as HTTPS [RFC7231]. It almost goes without saying implementers should be using the most up-to-date TLS implementation (or its successor) when retrieving registry elements from IANA. Likewise, the application resolving the URI MUST verify the domain in the certificate matches the IANA domain. The application resolving the URI MUST use DNSSEC [RFC4035] if it is available to the client. Finally, during TLS negotiation the application MUST verify the authority signing IANA's certificate matches the application's understanding of who is expected to sign IANA's certificate. At the time of this writing, that root certificate would be the DigiCert High Assurance EV Root CA.

7. Acknowledgements

Russ Housley and Sean Turner helped with the decision of registering certificates instead of URIs. Ken Carlberg and Padma Krishnaswamy of the United States Federal Communications Commission provided useful feedback in an incredibly short time period. Finally, a huge thank-you to Michelle Cotton and Kim Davies for helping normalize the registries and the procedures for populating them.

8. References

8.1. Normative References

[E.164D] International Telecommunications Union, "List of ITU-T Recommendation E.164 Assigned Country Codes", ITU-T Recommendation E.164 Annex D, 11 2011, <https://www.itu.int/dms_pub/itu-t/opb/sp/T-SP-E.164D-2016-PDF-E.pdf>.

[ISO.3166-1.2013]
International Organization for Standardization, "Codes for the representation of names of countries and their subdivisions -- Part 1: Country codes, 3rd edition", ISO Standard 3166-1, 11 2013.

[ITU-D.Agencies]
International Telecommunications Union - Development Sector, "National Telecommunication Agencies", 12 2017, <http://www.itu.int/en/ITU-D/Statistics/Pages/links/nta.aspx>.

[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>.

[RFC2315] Kaliski, B., "PKCS #7: Cryptographic Message Syntax Version 1.5", RFC 2315, DOI 10.17487/RFC2315, March 1998, <https://www.rfc-editor.org/info/rfc2315>.

[RFC4035] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Protocol Modifications for the DNS Security Extensions", RFC 4035, DOI 10.17487/RFC4035, March 2005, <https://www.rfc-editor.org/info/rfc4035>.

[RFC7231] Fielding, R., Ed. and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content", RFC 7231, DOI 10.17487/RFC7231, June 2014, <https://www.rfc-editor.org/info/rfc7231>.

[RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <https://www.rfc-editor.org/info/rfc8126>.

8.2. Informative References

[E.164] International Telecommunications Union, "The International Public Telecommunication Numbering Plan", ITU-T Recommendation E.164, 11 2010, <https://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-E.164-201011-I!!PDF-E&type=items>.

[FCC_intl] Ashton, S. and L. Blake, "2014 U.S. International Telecommunications Traffic and Revenue Data", 7 2016, <http://transition.fcc.gov/Daily_Releases/Daily_Business/2016/db0701/DOC-340121A1.pdf>.

[NPAreport] North American Numbering Plan Administrator, "NPA Database", 12 2017, <https://www.nationalnanpa.com/nanp1/npa_report.csv>.

[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>.

[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>.

[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>.

[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>.

Author’s Address

Eric W. Burger
Georgetown University
37th & O St, NW
Washington, DC 20057
USA

Email: eburger@standardstrack.com

STIR
Internet-Draft
Intended status: Standards Track
Expires: September 9, 2020

E. Burger
Georgetown University

March 8, 2020

Registry for Country-Specific Secure Telephone Identity (STIR) Trust Anchors
draft-burger-stir-iana-cert-01

Abstract

National policy defines telephone numbering governance. One area of such governance are the policies applied to the Secure Telephone Identity Credentials defined in RFC 8226. Nations have policies for the acceptable trust anchors for these credentials. This document defines an IANA registry that enables a SIP call recipient in one country to validate the signature, as defined in RFC 8224, that originates in another country useing an appropriate trust anchor for the signer's certification path, per the origination country's trust anchor policy.

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 9, 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.

1. Introduction

One problem that plagues some communications applications is a caller deliberately misrepresenting their identity with the intent to defraud, cause harm, or wrongfully obtain anything of value. The IETF Secure Telephone Identity Revisited (STIR) work group has developed a series of RFCs specifying the mechanisms for cryptographically signing the asserted identity and other elements in Session Initiation Protocol (SIP) [RFC3261] messages. One kind of identity used in SIP is an E.164 [E.164] telephone number. A telephone number is a string of digits, where the first one to three digits indicate a country code. The International Telecommunications Union - Telecommunications Sector (ITU-T) defines country codes and delegates the authority for numbers under a country code to the respective national communications authority for that country, as listed in E.164 Annex D [E.164D]. Note the country code does not itself necessarily uniquely identify a country. For example, in country codes +1 and +7, multiple countries share the country code. In the cases of +1 and +7, further digits in the E.164 number, known as national significant digits (also known as area codes in +1) further identify the country. As well, there are non-geographic services with country codes assigned to them.

Section 7 of Authenticated Identity Management in the Session Initiation Protocol [RFC8224] describes the process for signing identity tokens. Correspondingly, the STIR Certificates document [RFC8226] describes the format of the signing certificate. The protocol and formats are independent of and can have uses beyond that of signing originating telephone numbers. As well, given that for the most part governments are responsible for managing the numbering resources within their country code, governmental policy may impact who is authorized to issue signing certificates and what constitutes a valid certification path. As such, the base STIR documents defer certificate and validation policy to other documents. This document describes a registry for finding a STIR trust anchor for a given country code for signed telephone numbers. This document only enables policies for E.164 number identity assertions. Moreover, while this document describes the STIR trust anchor registry for various national STIR trust anchors, it does not mandate any particular policy regime.

Recalling the STIR problem statement [RFC7340], the goal is to provide authenticated identity for the caller. When a SIP endpoint receives a message with a signed STIR token, that endpoint needs to know whether the signing certificate is, in fact, allowed to make assertions for that identity. It does us no good for a caller with ill intent to have a signed assertion that has a valid certification path to an unauthorized trust anchor. Likewise, it does us no good to use self-signed certificates to sign a SIP message, as even with some limited verification, if there is the slightest chance of an entity with nefarious intent to succeed in either spoofing or taking over the identify of a caller, experience has shown they will do so.

As mentioned above, the ITU-T assigns telephone numbers, specifically the responsibility to assign numbers beneath a country's country code, to national communications authorities. A national regulator can inform service providers under its authority which trust anchors are authoritative for numbers under its control. This is straightforward within a country. However, this does not work for the global, interconnected communications network. When someone in a first country calls someone in a second country, how is the service provider or end user in the second country to know who is authoritative for signing certificates in the first country?

To solve this problem, this document establishes an IANA registry of STIR trust anchors, indexed by country codes.

2. Terminology

This document uses the terms "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" as RFC 2119 [RFC2119] defines them.

As noted above, a country code may not sufficiently identify a particular country. Likewise, national policy may assign different STIR trust anchors for different sets of national significant numbers (e.g., area codes). For example, while +7 generally identifies the Russian Federation, +76 and +77 identify Kazakhstan. Likewise, +1 generally identifies the North American Numbering Plan (NANP), which identifies countries by area code (the following three digits after the country code). For example, +1869 identifies Saint Kitts and Nevis while +1649 identifies Turks and Caicos. The term "country code" appearing from this point forward in this document refers to the country code and, if necessary, the subsequent digits that identify a country or region. With the exception of ITU-T country code +1, the ITU-T country code is the "country code" for the purposes of this registry. In the NANP (+1) case, this means the "country code" can be four digits long. Specifically, to identify a

specific country in the NANP, what this document terms the "country code" will be the leading +1 and the following three-digit area code.

3. STIR Trust Anchor Registry

This registry maps E.164 country codes to STIR trust anchors. There can be one or more STIR trust anchors per country code.

3.1. Numeric Country Code

E.164 [E.164] defines the country code as a one- to three-digit string. However, there are some country codes that have different country delegations beyond the country code. In these cases, we use additional digits in the number to unambituously identify a country. For example, footnote b of E.164 Annex D [E.164D] shows 25 countries under country code +1 and two countries under country code +7. As well, country code +881, for satellite services, and codes +882 and +883, for international networks, are under the jurisdiction of various national authorities.

To distinguish the various national authorities under a given country code, the country code entry can contain these identity codes. Currently, the longest entry can be seven digits, but this could change in the future. As noted above, distinguishing the appropriate certificate to use can be a matter of local policy. We suggest longest match, but be aware that local policy may dictate another policy within that jurisdiction.

3.2. STIR Trust Anchor

Each country can have zero or more STIR trust anchors. The trust anchor is a self-signed certificate [RFC5280]. The STIR trust anchor is the trust anchor for STIR (SIP) PKI in the given jurisdiction. In the common Web browser situation, a Web server operator can purchase a certificate issued by one of hundreds of certificate authorities from anywhere in the world. The expectation is the authority for signing the identity of a caller will be more strict than the authority for signing the identity of, for example, a Web site. To ensure interoperability, browser and operating system manufacturers need to include the STIR trust anchors from those certificate authorities so when a user in one part of the world accesses a Web server in another part of the world that has a certificate issued by a certificate authority in yet a different part of the world, the site will validate. In the telephone number identity situation, for the most part the individual national numbering authorities will choose a very limited set of STIR trust anchors who they will allow to issue signing certificates for numbers assigned to that country.

Within a single country, it would be a relatively easy matter for the national communications regulator to impose and inform their domestic service providers who is the designated certificate authority within that country. However, given the large amount of international telephone traffic (as an example, there were over 100,000,000,000 minutes of traffic between the U.S. and other countries in 2014, including VoIP [FCC_intl]), there is a need for service providers and users in different countries to validate that one of the proper certificate authorities for that country has issued the signing certificate.

The entry for each national STIR trust anchor is a text certificate [RFC7468] that contains the public key of the STIR trust anchor, matching the private key the STIR trust anchor uses to sign signing keys used by its delegates, such as telecommunications service providers.

4. IANA Considerations

Refer to [RFC8126] for a description of IANA Considerations terms and their meanings.

4.1. Registry Policy: First Come First Served

This registry is First Come First Served, understanding there can be multiple trust anchors registered for a given Country Code prefix. The integrity of an originating nation's numbering system is generally the purview of the respective national government. Moreover, the integrity of a terminating network, including the accuracy of received signaling, is generally the purview of the government with jurisdiction over the terminating network. We do not anticipate IANA to intervene in disputes of who has the authority for entering and changing STIR trust anchors. In general, IANA SHOULD validate the request originates from an entity authorized by the recognized national authority for the country as specified in [ITU-D.Agencies], unless it is not clear who the national authority is. However, because it is likely the regulatory authorities in the terminating country will determine the validity of the STIR trust anchor found in the IANA registry, irrespective of the depth of vetting IANA could perform, if IANA believes the registration is not fraudulent, it SHOULD accept the registration even if it cannot positively identify or contact the appropriate national authority.

4.2. Registry Elements

The STIR Trust Anchor registry consists of one or more entities indicating the public keys of STIR trust anchors for a given country code. With around 200 countries, each of which might have one to

four STIR trust anchors, results in a registry with a total participation of about one thousand entries. The expectation is there would be substantially fewer entries in practice.

4.2.1. Numeric Country Code

The numeric country code is a one- to eight-digit string indicating the numeric country code and optional identity digits. Identity digits are often known as an area code or city code. [E.164D] lists country codes and the identity digits when there are overlapping country codes (+1, +7, and some international codes).

IANA MUST verify the requested mapping includes a valid numeric country code as specified in E.164 Annex D.

NOTE: The conventional leading + to indicate the string identifies a country code is NOT part of the Country Code element in the registry.

4.2.2. STIR Trust Anchor

The STIR trust anchor is an RFC7468 [RFC7468] text file that contains the public key of the authorized STIR trust anchor that signs the certificates authorized to sign STIR signaling in the given country. There can be one or more entries in the registry for a given ISO country code to allow for multiple STIR trust anchors for a given country.

IANA MUST verify the certificate is valid by using the provided public key in the certificate to validate the signature in the certificate.

IANA SHOULD remove a STIR trust anchor from the registry if the certificate expires.

4.2.3. Domain of Authority

For traceback and reputation purposes, IANA MUST record the validated domain of the entity that made the request to enter, delete, or modify an entry in the STIR Trust Anchor Registry. The mechanism for validating the domain is a matter of IANA policy. Mechanisms include ensuring an emailed request uses DKIM [RFC6376] with secure cryptographic algorithms [RFC8301], web requests have validated client certificates identifying the domain of the requestor, or out of band methods. Note that an unauthenticated inbound phone call is not likely to be an acceptable mechanism of identifying the domain.

4.3. Other IANA Considerations

There is the potential for a malicious actor attempting to load a trust anchor that could enable them to sign spoofed signaling. As such, IANA SHOULD note who is making the request, to sufficient detail to locate that party for referral to the relevant national authorities. For most countries, it will be the national authority itself or a clear delegate that will be making the registration. For example, in the United States, the Federal Communications Commission has delegated the governance of the STIR trust anchor to the U.S. STI-GA, administered by ATIS, which is an identifiable, incorporated entity with a fixed, physical address.

5. Security Considerations

The choice of having the STIR trust anchor stored by IANA means that users accessing the certificates MUST use a source-authenticated retrieval mechanism, such as HTTPS [RFC7231]. It almost goes without saying implementers should be using the most up-to-date TLS implementation (or its successor) when retrieving registry elements from IANA. Likewise, the application resolving the URI MUST verify the domain in the certificate matches the IANA domain. The application resolving the URI MUST use DNSSEC [RFC4035] if it is available to the client. Finally, during TLS negotiation the application MUST verify the authority signing IANA's certificate matches the application's understanding of who should sign IANA's certificate. At the time of this writing, that trust anchor would be the DigiCert High Assurance EV Root CA.

Because IANA takes no responsibility for the accuracy of any given country's STIR trust anchor entry, this document presumes the terminating provider or local authority will use local policy to determine the trustworthiness of any given entry. ATIS [ATIS-Intl] describes an example of such a local policy.

6. Acknowledgements

Russ Housley, Jim McEachern, and Sean Turner gave invaluable insight. Ken Carlberg and Padma Krishnaswamy of the United States Federal Communications Commission provided useful feedback in an incredibly short time period. Finally, a huge thank-you to Michelle Cotton and Kim Davies for helping normalize the registries and the procedures for populating them.

7. References

7.1. Normative References

[E.164D] International Telecommunications Union, "List of ITU-T Recommendation E.164 Assigned Country Codes", ITU-T Recommendation E.164 Annex D, 11 2011, <https://www.itu.int/dms_pub/itu-t/opb/sp/T-SP-E.164D-2016-PDF-E.pdf>.

[ITU-D.Agencies] International Telecommunications Union - Development Sector, "National Telecommunication Agencies", 12 2017, <http://www.itu.int/en/ITU-D/Statistics/Pages/links/nta.aspx>.

[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>.

[RFC4035] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Protocol Modifications for the DNS Security Extensions", RFC 4035, DOI 10.17487/RFC4035, March 2005, <https://www.rfc-editor.org/info/rfc4035>.

[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>.

[RFC6376] Crocker, D., Ed., Hansen, T., Ed., and M. Kucherawy, Ed., "DomainKeys Identified Mail (DKIM) Signatures", STD 76, RFC 6376, DOI 10.17487/RFC6376, September 2011, <https://www.rfc-editor.org/info/rfc6376>.

[RFC7231] Fielding, R., Ed. and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content", RFC 7231, DOI 10.17487/RFC7231, June 2014, <https://www.rfc-editor.org/info/rfc7231>.

[RFC7468] Josefsson, S. and S. Leonard, "Textual Encodings of PKIX, PKCS, and CMS Structures", RFC 7468, DOI 10.17487/RFC7468, April 2015, <https://www.rfc-editor.org/info/rfc7468>.

[RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <https://www.rfc-editor.org/info/rfc8126>.

[RFC8301] Kitterman, S., "Cryptographic Algorithm and Key Usage Update to DomainKeys Identified Mail (DKIM)", RFC 8301, DOI 10.17487/RFC8301, January 2018, <https://www.rfc-editor.org/info/rfc8301>.

7.2. Informative References

[ATIS-Intl] Alliance for Telecommunications Industry Solutions, "Mechanism for International Signature-based Handling of Asserted information using toKENs (SHAKEN)", <http://access.atis.org/apps/org/workgroup/ipnni/download.php/51306/IPNNI-2020-00032R000.docx>.

[E.164] International Telecommunications Union, "The International Public Telecommunication Numbering Plan", ITU-T Recommendation E.164, 11 2010, <https://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-E.164-201011-I!!PDF-E&type=items>.

[FCC_intl] Ashton, S. and L. Blake, "2014 U.S. International Telecommunications Traffic and Revenue Data", 7 2016, <http://transition.fcc.gov/Daily_Releases/Daily_Business/2016/db0701/DOC-340121A1.pdf>.

[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>.

[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>.

[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>.

[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>.

Author's Address

Eric W. Burger
Georgetown University
37th & O St, NW
Washington, DC 20057
USA

Email: eburger@standardstrack.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 6, 2018

E. Rescorla
Mozilla
J. Peterson
Neustar

March 5, 2018

STIR Out-of-Band Architecture and Use Cases
draft-ietf-stir-oob-02.txt

Abstract

The PASSporT format defines a token that can be carried by signaling protocols, including SIP, to cryptographically attest the identify of callers. Not all telephone calls use Internet signaling protocols, however, and some calls use them for only part of their signaling path. This document describes use cases that require the delivery of PASSporT objects outside of the signaling path, and defines architectures and semantics to provide this functionality.

Status of This Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at 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 6, 2018.

Copyright Notice

Copyright (c) 2018 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 6
5.2. Case 2: Two Smart PSTN endpoints 6
5.3. Case 3: PSTN to VoIP Call 7
5.4. Case 4: Gateway Out-of-band 7
6. Storing and Retrieving PASSporTs 8
6.1. Storage 9
6.2. Retrieval 10
7. Solution Architecture 11
7.1. Credentials and Phone Numbers 12
7.2. Call Flow 12
7.3. Security Analysis 13
7.4. Substitution Attacks 13
8. Authentication and Verification Service Behavior for Out-of-Band 14
8.1. Authentication Service 14
8.2. Verification Service 16
8.3. Gateway Placement Services 17
9. HTTPS Interface to the CPS 17
10. CPS Discovery 19
11. Credential Lookup 20
12. Acknowledgments 21
13. IANA Considerations 21
14. Security Considerations 21
15. Informative References 21
Authors' Addresses 22

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 an Identity header of SIP requests capable of carrying a PASSporT [RFC8225] object in SIP as a means to cryptographically attest that the originator of a

telephone call is authorized to use the calling party number (or, for native SIP cases, SIP URI) associated with the originator of the call. of the request.

Not all telephone calls use SIP today, however; and even those that do use SIP do not always carry SIP signaling end-to-end. Most calls from telephone numbers still traverse the Public Switched Telephone Network (PSTN) at some point. Broadly, calls fall into one of three categories:

1. One or both of the endpoints is actually a PSTN endpoint.

2. Both of the endpoints are non-PSTN (SIP, Jingle, ...) but the call transits the PSTN at some point.

3. Non-PSTN calls which do not transit the PSTN at all (such as native SIP end-to-end calls).

The first two categories represent the majority of telephone calls associated with problems like illegal robocalling: many robocalls today originate on the Internet but terminate at PSTN endpoints. However, the core network elements that operate the PSTN are legacy devices that are unlikely to be upgradable at this point to support an in-band authentication system. As such, those devices largely cannot be modified to pass signatures originating on the Internet--or indeed any inband signaling data--intact. Even if fields for tunneling arbtirary data can be found in traditional PSTN signaling, in some cases legacy elements would strip the signatures from those fields; in others, they might damage them to the point where they cannot be verified. For those first two categories above, any in-band authentication scheme does not seem practical in the current environment.

But while the core network of the PSTN remains fixed, the endpoints of the telephone network are becoming increasingly programmable and sophisticated. Landline "plain old telephone service" deployments, especially in the developed world, are shrinking, and increasingly being replaced by three classes of intelligent devices: smart phones, IP PBXs, and terminal adapters. All three are general purpose computers, and typically all three have Internet access as well as access to the PSTN. Additionally, various kinds of gateways increasingly front for legacy equipment. All of this provides a potential avenue for building an authentication system that implements stronger identity while leaving PSTN systems intact.

This capability also provides an ideal transitional technology while in-band STIR adoption is ramping up. It permits early adopters to use the technology even when intervening network elements are not yet

STIR-aware, and through various kinds of gateways it may allow providers with a significant PSTN investment to still secure their calls with STIR.

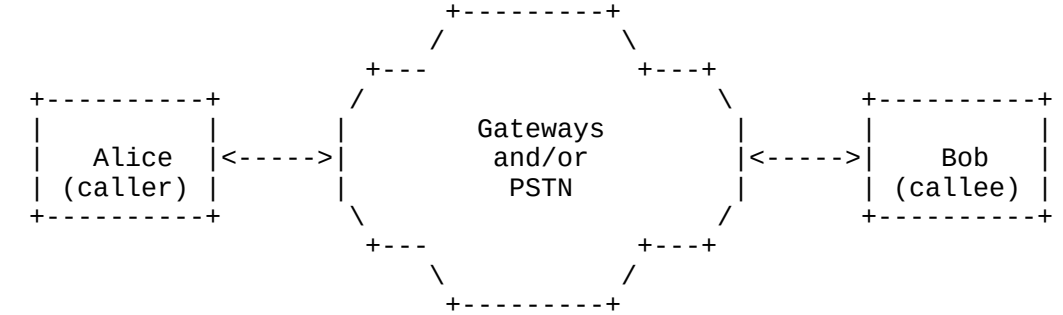
This specification therefore builds on the PASSporT [RFC8225] mechanism and the work of [RFC8224] to define a way that a PASSporT object created in the originating network of a call can reach the terminating network even when it cannot be carried end-to-end in-band in the call signaling. This relies on a new service defined in this document that permits the PASSporT object to be stored during call processing and retrieved for verification purposes.

2. Terminology

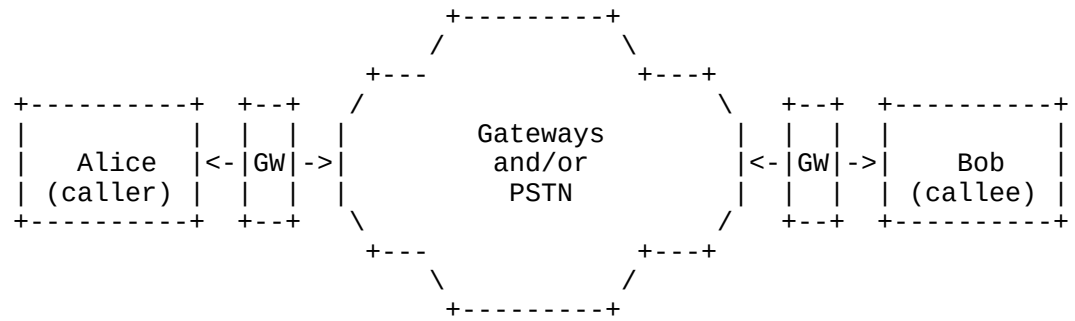
The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

3. Operating Environments

This section describes the environments in which the proposed mechanism is intended to operate. In the simplest setting, Alice is calling Bob through some set of gateways and/or the PSTN. Both Alice and Bob have smart devices which can be modified, but they do not have a clear connection between them: Alice cannot inject any data into signaling which Bob can read, with the exception of the asserted destination and origination E.164 numbers. The calling party number might originate from her own device or from the network. These numbers are effectively the only data that can be used for coordination between the endpoints.



In a more complicated setting, Alice and/or Bob may not have a smart or programmable device, but one or both of them are behind a STIR-aware gateway that can participate in out-of-band coordination, as shown below:



In such a case, Alice might have an analog connection to her gateway/ switch which is responsible for her identity. Similarly, the gateway would verify Alice's identity, generate the right calling party number information and provide that number to Bob using ordinary POTS mechanisms.

4. Dataflows

Because in these operating environments endpoints cannot pass cryptographic information to one another directly through signaling, any solution must involve some rendezvous mechanism to allow endpoints to communicate. We call this rendezvous service a "call placement service" (CPS), a service where a record of call placement, in this case a PASSporT, can be stored for future retrieval. In principle this service could communicate any information, but minimally we expect it to include a full-form PASSporT that attests the caller, callee, and the time of the call. The callee can use the existence of a PASSporT for a given incoming call as rough validation of the asserted origin of that call. (See Section 11 for limitations of this design.)

There are roughly two plausible dataflow architectures for the CPS:

The callee registers with the CPS. When the caller wishes to place a call to the callee, it sends the PASSporT to the CPS, which immediately forwards it to the callee.

The caller stores the PASSporT with the CPS at the time of call placement. When the callee receives the call, it contacts the CPS and retrieves the PASSporT.

While the first architecture is roughly isomorphic to current VoIP protocols, it shares their drawbacks. Specifically, the callee must maintain a full-time connection to the CPS to serve as a notification channel. This comes with the usual networking costs to the callee and is especially problematic for mobile endpoints. Indeed, if the

endpoints had the capabilities to implement such an architecture, they could surely just use SIP or some other protocol to set up a secure session; even if the media were going through the traditional PSTN, a "shadow" SIP session could convey the PASSporT. Thus, we focus on the second architecture in which the PSTN incoming call serves as the notification channel and the callee can then contact the CPS to retrieve the PASSporT.

5. Use Cases

The following are the motivating use cases for this mechanism. Bear in mind that just as in [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 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 reaches a gateway to the PSTN. Eventually, the call will terminate on a mobile smartphone that supports this out-of-band mechanism.

In this use case, the originating authentication service can store the PASSporT with the appropriate CPS for the target telephone number as a fallback in case SIP signaling will not reach end-to-end. When the destination mobile smartphone receives the call over the PSTN, it consults the CPS and discovers a PASSporT from the originating telephone number waiting for it. It uses this PASSporT to verify the calling party number.

5.2. Case 2: Two Smart PSTN endpoints

A call originates with an enterprise PBX that has both Internet access and a built-in gateway to the PSTN. It will immediately drop its call to the PSTN, but before it does, it provisions a PASSporT on the CPS associated with the target telephone number.

After normal PSTN routing, the call lands on a smart mobile handset that supports the STIR out-of-band mechanism. It queries the appropriate CPS over the Internet to determine if a call has been

placed to it by a STIR-aware device. It finds the PASSporT provisioned by the enterprise PBX and uses it to verify the calling party number.

5.3. Case 3: PSTN to VoIP Call

A call originates with an enterprise PBX that has both Internet access and a built-in gateway to the PSTN. It will immediate drop the call to the PSTN, but before it does, it provisions a PASSporT with the CPS associated with the target telephone number. However, it turns out that the call will eventually route through the PSTN to an Internet gateway, which will translate this into a SIP call and deliver it to an administrative domain with a STIR verification service.

In this case, there are two subcases for how the PASSporT might be retrieved. In subcase 1, the Internet gateway that receives the call from the PSTN could query the appropriate CPS to determine if the original caller created and provisioned a PASSporT for this call. If so, it can retrieve the PASSporT and, when it creates a SIP INVITE for this call, add a corresponding Identity header per [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 the IP world. In the former case, perhaps the destination endpoints

queries the CPS to retrieve the PASSporT provisioned by the first gateway. Or if the call ultimately returns to the IP world, it might be the gateway from the PSTN back to the Internet that retrieves the PASSporT from the CPS and attaches it to the new SIP INVITE it creates, or it might be the terminating administrative domain's verification service that checks the CPS when an INVITE arrives with no Identity header field. Either way the PASSporT can survive the gap in SIP coverage caused by the PSTN leg of the call.

6. 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. Broadly, the architecture described here is one focused on permitting any entity to store encrypted PASSporTs at the CPS, indexed under the caller number. PASSporTs will be encrypted with associated with the called number, so these PASSporTs may also 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 would make the CPS a less attractive target for pervasive monitoring (see [RFC7258]). Ho bolster the privacy story, prevent denial-of-service flooding of the CPS, and to complicate traffic analysis, a few additional mechanisms are also recommended.

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 and 4 in Section 5 show, it may sometimes make sense for the entity storing or retrieving PASSporTs to be an intermediary rather than a device associated with either the originating or terminating side of a call, and those intermediaries often would not have access to STIR credentials covering the telephone numbers in question. Requiring authorization based on a credential to store PASSporTs is therefore undesirable, though potentially acceptible if sufficient steps are taken to mitigate the privacy risk as described in the next section.

Furthermore, 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. As a first step, transport-level security can provide confidentiality from eavesdroppers for both the storage and retrieval of PASSporTs.

6.1. Storage

For authorizing the storage of PASSporTs, the architecture can permit some flexibility. Note that in this architecture a CPS has no way to tell if a PASSporT is valid; it simply conveys encrypted blocks that it cannot access itself. In that 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.

Note that this architecture requires clients that stores PASSporTs to have access to a public key associated with the intended called party to be used to encrypt the PASSporT. Discovering this key requires some new service that does not exist today; depending on how the CPS is architected, however, some kind of key store or repository could be implemented adjacent to it, and perhaps even incorporated into its operation. 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 therefore might need to be encrypted with multiple keys in the hopes that one will be decipherable by the relying party.

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 target number, and thereby prevent that called party from finding a valid PASSporT for an incoming call buried in a haystack of fake entries. A CPS must therefore implement 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 ths source and destination of traffic.

In order to do this, we propose the use of "blind signatures". A sender will initially authenticate to the CPS, and acquire a signed token for 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 acqusition 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 more than 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. harder to mount

6.2. Retrieval

For retrieval of PASSporTs, this architecture assumes that clients contact the CPS to send requests of the form:

Are there any current PASSporTs for calls destined to 2.222.222.2222?

As all PASSporTs stored at the CPS are encrypted with a key belonging to the intended destination, then potentially the CPS could 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 provides 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.

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 decrypt all PASSporTs retrieved from the CPS, and may find only one that is valid.

Note that in call forwarding cases, the difficulties in managing the relationship between PASSporTs with the diversion extension [I-D.ietf-stir-passport-divert] become more serious. The originating authentication service would encrypt the PASSporT with the public key of the intended destination, but when a call is forwarded, it may go to a destination that does not possess the corresponding private key. This requires special behavior on the part of the retargeting entity, and probably the CPS as well, to accommodate encrypted PASSporTs that show a secure chain of diversion. A storer could for example notify the CPS that the divert PASSporT it is storing relates to a specific PASSporT already in the CPS, but in so doing, the storer will inevitably reveal more metadata to the CPS.

7. Solution Architecture

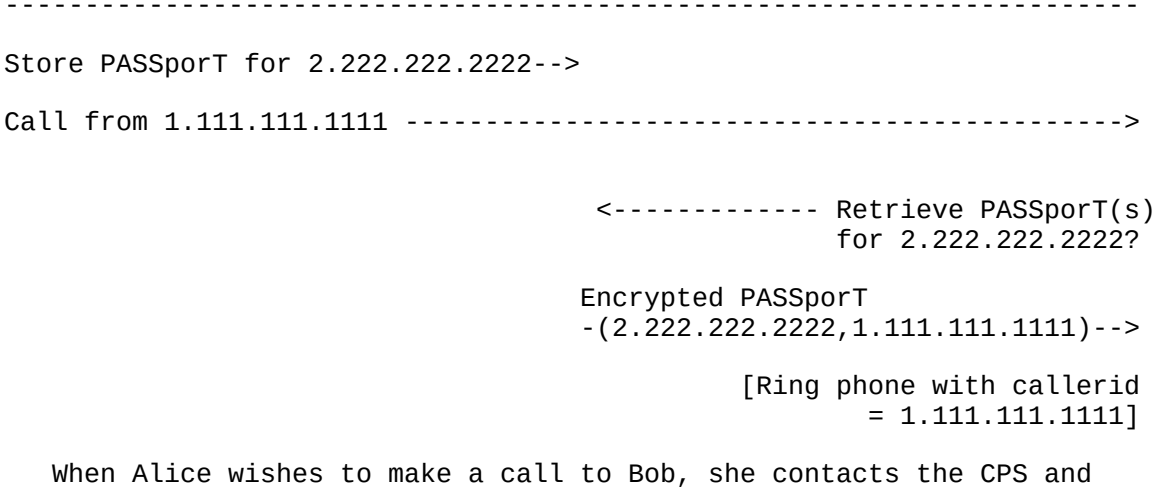
In this section, we discuss a strawman architecture for providing the service described in the previous sections. This discussion is deliberately sketchy, focusing on broad concepts and skipping over details. The intent here is merely to provide an overall architecture, not an implementable specification.

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 credentials system suitable for this purpose; the question of how an entity is determined to have control of a given number is out of scope for the current document.

7.2. Call Flow

An overview of the basic calling and verification process is shown below. In this diagram, we assume that Alice has the number +1.111.111.1111 and Bob has the number +2.222.222.2222.



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.

Once Alice has stored the PASSporT, she then places the call to Bob as usual. At this point, Bob’s phone would usually ring and display Alice’s number (+1.111.111.1111), which is informed by the existing PSTN mechanisms for relying a calling party number (i.e., the CIN field of the IAM). Instead, Bob’s phone transparently contacts the CPS and requests any current PASSporTs for calls to his number. The CPS responds with any such PASSporTs (assuming they exist). 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 then present the calling party number information as valid. Otherwise, the call is unverifiable. Note that this does not necessarily mean that the call is bogus; because we expect incremental deployment many legitimate calls will be unverifiable.

7.3. Security Analysis

The primary attack we seek to prevent is an attacker convincing the callee that a given call is from some other caller C. There are two scenarios to be concerned with:

The attacker wishes to impersonate a target when no call from that target is in progress.

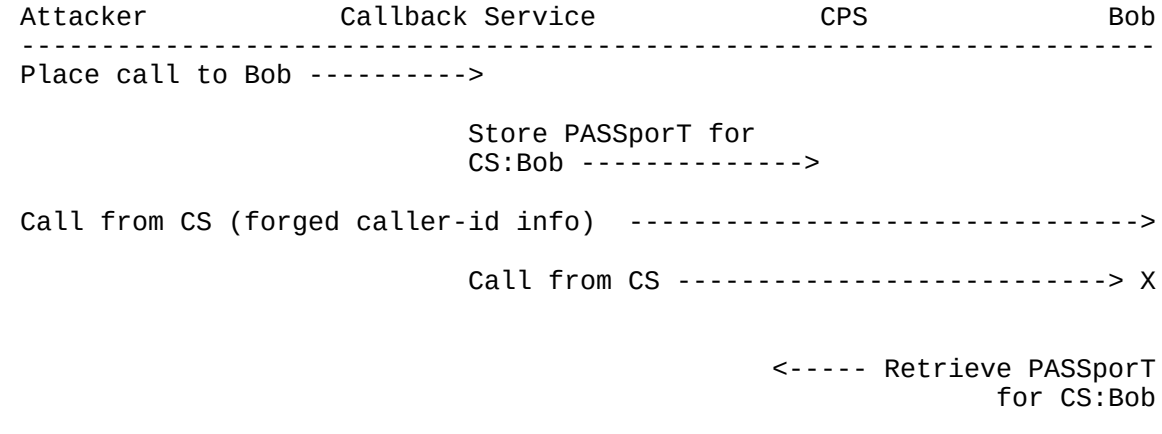
The attacker wishes to substitute himself for an existing call setup as described in Section 7.4.

If an attacker can inject fake PASSporT into the CPS or in the communication from the CPS to the callee, he can mount either attack. As PASSporTs should be digitally signed by an appropriate authority for the number and verified by the callee (see Section 7.1), this should not arise in ordinary operations. For privacy and robustness reasons, using TLS on the originating side when storing the PASSporT at the CPS is recommended.

The entire system depends on the security of the credential infrastructure. If the authentication credentials for a given number are compromised, then an attacker can impersonate calls from that number. However, that is no different from in-band [RFC8224] STIR.

7.4. Substitution Attacks

All that receipt of the PASSporT from the CPS proves to the called party is that Alice is trying to call Bob (or at least was as of very recently) - it does not prove that any particular incoming call is from Alice. Consider the scenario in which we have a service which provides an automatic callback to a user-provided number. In that case, the attacker can try to arrange for a false caller-id value, as shown below:



In order to mount this attack, the attacker contacts the Callback Service (CS) and provides it with Bob's number. This causes the CS to initiate a call to Bob. As before, the CS contacts the CPS to insert an appropriate PASSporT and then initiates a call to Bob. Because it is a valid CS injecting the PASSporT, none of the security checks mentioned above help. However, the attacker simultaneously initiates a call to Bob using forged caller-id information corresponding to the CS. If he wins the race with the CS, then Bob's phone will attempt to verify the attacker's call (and succeed since they are indistinguishable) and the CS's call will go to busy/voice mail/call waiting. Note: in a SIP environment, the callee might notice that there were multiple INVITEs and thus detect this attack.

8. 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

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 something 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 have a private key, such one corresponding to a STIR certificate, that can sign for the originating identity?

Step 3: The authentication service MUST acquire the public key of the destination, which will be used to encrypt the PASSporT. It must also discover (see Section 10) the CPS associated with the destination. The authentication service may already have the key and destination CPS cached, or may need to query a service to acquire the key. Note that per Section 6.1 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 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 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 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

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. 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 called party number 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 described in [RFC8225].

Finally, the verification service will end up with one or more valid PASSporTs corresponding to the call it has received. This document does not prescribe any particular treatment of calls that have valid PASSporTs associated with them. The handling of the message 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 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, extract a PASSporT from the Identity header, 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 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. HTTPS Interface to the CPS

The default Call Placement Service implementation 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 2.222.222.2222: [TBD - these are currently dummy values, will mock up real examples later]

eyJhbGciOiJFUzI1NiIsInR5cCI6InBhc3Nwb3J0IiwieDV1IjoiaHR0cHM6Ly9j
ZXJ0LmV4YW1wbGUub3JnL3Bhc3Nwb3J0LmNlciJ9.eyJkZXN0Ijp7InVyaSI6WyJz
aXA6YWxpY2VAZXhhbXBsZS5jb20iXX0sImlhdCI6IjE0NDMyMDgzNDUiLCJvcmlnI
jp7InRuIjoiMTIxNTU1NTEyMTIifX0.rq3pjT1hoRwakEGjHCnWSwUnshd0-zJ6F1
VOgFWSjHBr8Qjpjlk-cpFYpFYsojNCpTzO3QfPOlckGaS6hEck7w

Through some out-of-band mechanism (see Section 10) the authentication service discovers the network location of a web service that acts as the CPS for 2.222.222.2222. Through the same mechanism, we will say that it has also discovered one public key for that destination. It uses that public key to encrypt the PASSporT, resulting in the encrypted PASSporT:

```
rlWuoTpvBvWSHmV1AvVfVaE5pPV6VaOup3Ajo3W0VvjvrQI1VwbvnUE0pUZ6Yl9w
MKW0YzI4LJ1joTHho3WaY3Oup3Ajo3W0YzAypvW9rlWxMKA0Vwc7VaIlnFV6JlWm
nKN6LJkcL2INMKuuoKOfMF5wo20vKK0fVzyuqPV6VwR0AQZlZQtmAQHvYPWipzyaV
wc7VaEhVwbvZGVkAGH1AGRlZGVvsK0ed3cwG1ubEjnxRTwUPaJFjHafuq0-mW6S1
IBtSJFwUOe8Dwcwyx-pcSLcSLfbwAPcGmB3DsCBypxTnF6uRpx7j
```

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/2.222.222.2222/ppts HTTP/1.1
Host: cps.example.com
Content-Type: application/passport

rlWuoTpvBvWSHmV1AvVfVaE5pPV6VaOup3Ajo3W0VvjvrQI1VwbvnUE0pUZ6Yl9w
MKW0YzI4LJ1joTHho3WaY3Oup3Ajo3W0YzAypvW9rlWxMKA0Vwc7VaIlnFV6JlWm
nKN6LJkcL2INMKuuoKOfMF5wo20vKK0fVzyuqPV6VwR0AQZlZQtmAQHvYPWipzyaV
wc7VaEhVwbvZGVkAGH1AGRlZGVvsK0ed3cwG1ubEjnxRTwUPaJFjHafuq0-mW6S1
IBtSJFwUOe8Dwcwyx-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 interrogates its CPS to ask for the set of incoming calls for its telephone number (2.222.222.2222).

```
GET /cps/2.222.222.2222/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/2.222.222.2222/ppts/ppt1/ which yields:

```
HTTP/1.1 200 OK
Content-Type: application/passport
Link: <https://cps.example.com/cps/2.222.222.2222/ppts>

rlWuoTpvBvWSHmV1AvVfVaE5pPV6VaOup3Ajo3W0VvjvrQI1VwbvnUE0pUZ6Yl9w
MKW0YzI4LJ1joTHho3WaY3Oup3Ajo3W0YzAypvW9rlWxMKA0Vwc7VaIlnFV6JlWm
nKN6LJkcL2INMKuuoKOfMF5wo20vKK0fVzyuqPV6VwR0AQZlZQtmAQHvYPWipzyaV
wc7VaEhVwbvZGVkAGH1AGRlZGVvsK0ed3cwG1ubEjnxRTwUPaJFjHafuq0-mW6S1
IBtSJFwUOe8Dwcwyx-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.

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 in terms of a single CPS, having a significant fraction of all telephone calls result in storing and retrieving PASSporTs at a single monolithic CPS has obvious scaling problems, and would as well allow the CPS to gather metadata about a very wide set of callers and callees. These issues can be alleviated by operational models with a federated CPS; any service discovery mechanism for out-of-band STIR should enable federation of the CPS function.

Some service discovery possibilities under consideration include the following:

If a credential lookup service is already available (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.peterson-modern-teri] protocol, could also serve for this application.

Another possibility is to use a single distributed service for this function. VIPR [I-D.rosenberg-dispatch-vipr-overview] proposed a RELOAD [RFC6940] usage for telephone numbers to help direct calls to enterprises on the Internet. It would be possible to describe a similar RELOAD usage to identify the CPS where calls for a particular telephone number should be stored. One advantage that the STIR architecture has over VIPR is that it assumes a credential system that proves authority over telephone numbers; those credentials could be used to determine whether or not a CPS could legitimately claim to be the proper store for a given telephone number.

Future versions of this specification will identify suitable service discovery mechanisms for out-of-band STIR.

11. Credential Lookup

In order to encrypt a PASSporT (see Section 6.1), the caller needs access to the callee's credentials (specifically their public key). This requires some sort of directory/lookup system. This document does not specify any particular scheme, but a list of requirements would be something like:

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

Have endpoints pre-fetch credentials for potential counterparties (e.g., their address book or the entire database).

Have caching servers in the user's network that proxy their fetches and thus conceal the relationship between the user and the credentials they are fetching.

Clearly, there is a privacy/timeliness tradeoff in that getting up-to-date knowledge about credential validity requires contacting the credential directory in real-time (e.g., via OCSP). This is somewhat mitigated for the caller's credentials in that he can get short-term credentials right before placing a call which only reveals his calling rate, but not who he is calling. Alternately, the CPS can verify the caller's credentials via OCSP, though of course this

requires the callee to trust the CPS's verification. This approach does not work as well for the callee's credentials, but the risk there is more modest since an attacker would need to both have the callee's credentials and regularly poll the database for every potential caller.

We consider the exact best point in the tradeoff space to be an open issue.

12. Acknowledgments

The ideas in this document come out of discussions with Richard Barnes and Cullen Jennings. We'd also like to thank Robert Sparks for helpful suggestions.

13. IANA Considerations

This memo includes no request to IANA.

14. Security Considerations

This entire document is about security, but the detailed security properties depend on having a single concrete scheme to analyze.

15. Informative References

[I-D.ietf-stir-passport-divert]
Peterson, J., "PASSporT Extension for Diverted Calls", draft-ietf-stir-passport-divert-01 (work in progress), October 2017.

[I-D.peterson-modern-teri]
Peterson, J., "An Architecture and Information Model for Telephone-Related Information (TeRI)", draft-peterson-modern-teri-03 (work in progress), July 2017.

[I-D.rosenberg-dispatch-vipr-overview]
Rosenberg, J., Jennings, C., and M. Petit-Huguenin, "Verification Involving PSTN Reachability: Requirements and Architecture Overview", draft-rosenberg-dispatch-vipr-overview-04 (work in progress), October 2010.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <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>.

[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>.

[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>.

Authors' Addresses

Eric Rescorla
Mozilla

Email: ekr@rtfm.com

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

Email: jon.peterson@neustar.biz

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 identify 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.

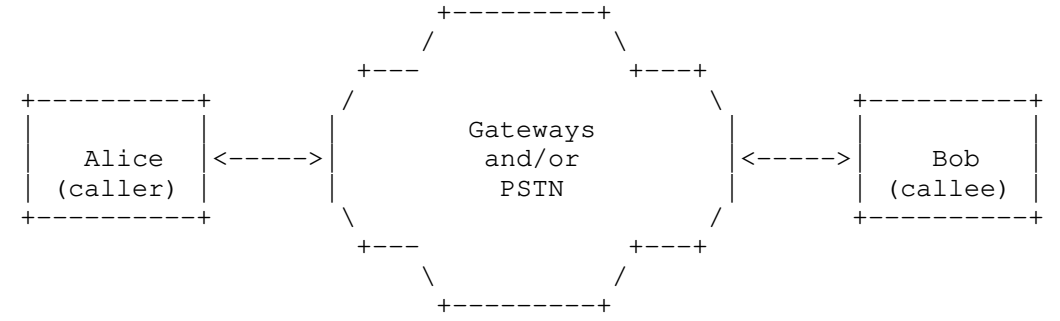
2. Terminology

TThe 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.

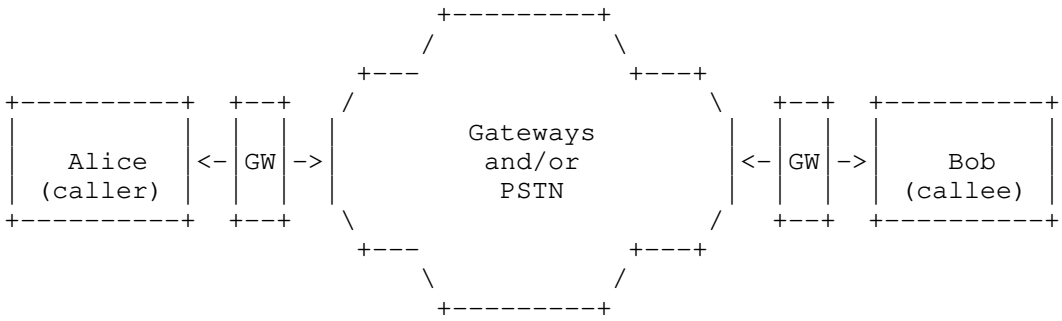
3. Operating Environments

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

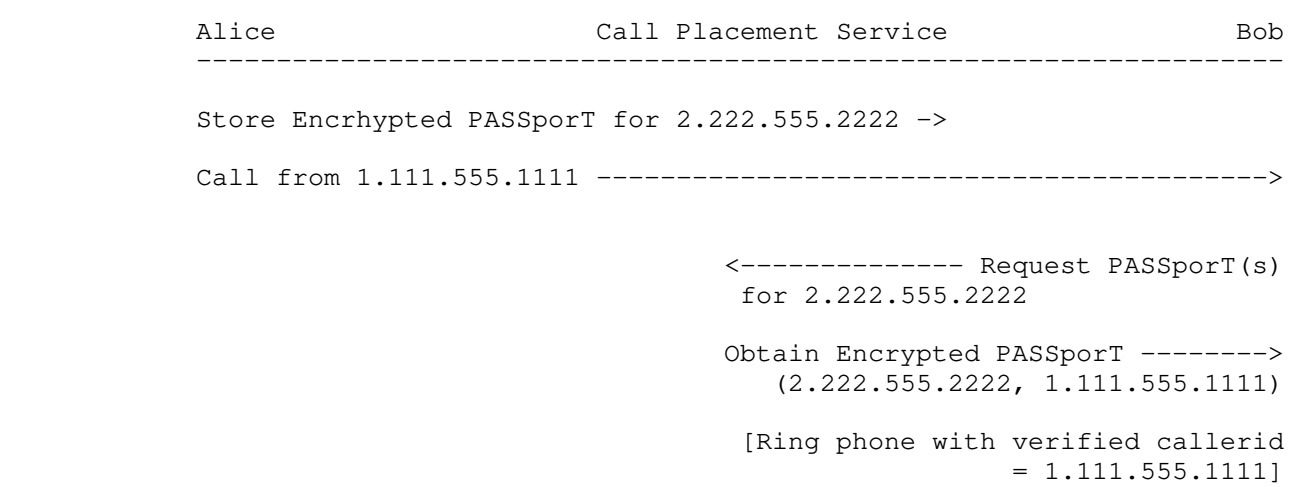
In this section, we discuss a high-level architecture for providing the service described in the previous sections. This discussion is deliberately sketchy, focusing on broad concepts and skipping over details. The intent here is merely to provide an overall architecture, not an implementable specification. 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.

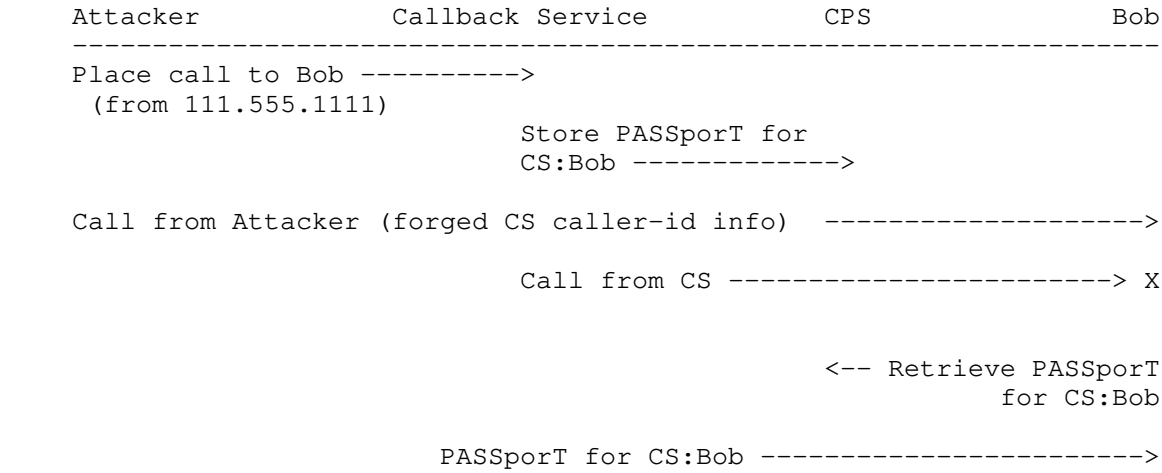
If an attacker can inject fake PASSporTs into the CPS or in the communication from the CPS to the callee, he can mount either attack. As PASSporTs should be digitally signed by an appropriate authority for the number and verified by the callee (see Section 7.1), this should not arise in ordinary operations. Any attacker who is aware of calls in progress can attempt to mount a race to subtitute 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:



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.

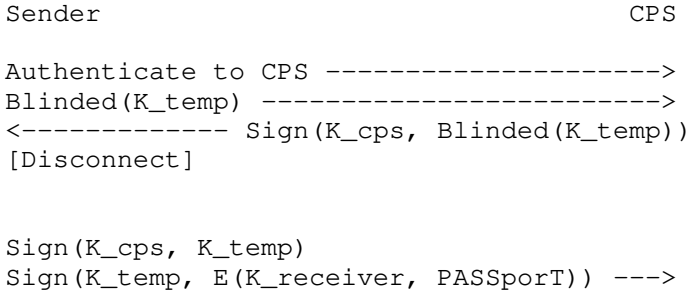
In order to prevent a passive attacker from using traffic analysis or similar means to learn precisely when a call is placed, it is essential that the connection between the caller and the CPS be encrypted as recommended above. Authentication services could store dummy PASSporTs at the CPS at random 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:



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):


```
eyJhbGciOiJFUzI1NiIsInR5cCI6InBhc3Nwb3J0IiwieDV1IjoiaHR0cHM6Ly9
jZXJ0LmV4YW1wbGUub3JnL3Bhc3Nwb3J0LmNlciJ9.eyJkZXN0Ijp7InRuIjpbI
jIyMjI1NTUyMjIyIl19LCJpYXQiOiIxNTgzMjUxODEwIiwib3JpZyI6eyJ0biI6
IjExMTE1NTUxMTExIn19.pnij4IlLHoR4vxID0u3CT1e9Hq4xLngZUTv45Vbxmd
3IVyZug4KOSa378yfP4x6twY0KTdiDypsereS438ZHaQ
```


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:


```
rlWuoTpvBvWSHmV1AvVfVaE5pPV6VaOup3Ajo3W0VvjvrQI1VwbvnUE0pUZ6Yl9w
MKW0YzI4LJ1joTHho3WaY3Oup3Ajo3W0YzAypvW9rlWxMKA0Vwc7VaIlnFV6JlWm
nKN6LJkcL2INMKuuoKOfMF5wo20vKK0fVzyuqPV6VwR0AQZlZQtmAQHvYPWipzyaV
wc7VaEhVwbvZGVkAGH1AGRlZGVvsK0ed3cwG1ubEjnxRTwUPaJFjHafuq0-mW6S1
IBtSJFwUOe8Dwcwyx-pcSLcSLfbwAPcGmB3DsCBypxTnF6uRpx7j
```


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

rlWuoTpvBvWSHmV1AvVfVaE5pPV6VaOup3Ajo3W0VvjvrQI1VwbvnUE0pUZ6Yl9w
MKW0YzI4LJ1joTHho3WaY3Oup3Ajo3W0YzAypvW9rlWxMKA0Vwc7VaIlnFV6JlWm
nKN6LJkcL2INMKuuoKOfMF5wo20vKK0fVzyuqPV6VwR0AQZlZQtmAQHvYPWipzyaV
wc7VaEhVwbvZGVkAGH1AGRlZGVvsK0ed3cwG1ubEjnxRTwUPaJFjHafuq0-mW6S1
IBtSJFwUOe8Dwcwyx-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>

rlWuoTpvBvWSHmV1AvVfVaE5pPV6VaOup3Ajo3W0VvjvrQI1VwbvnUE0pUZ6Yl9w
MKW0YzI4LJ1joTHho3WaY3Oup3Ajo3W0YzAypvW9rlWxMKA0Vwc7VaIlnFV6JlWm
nKN6LJkcL2INMKuuoKOfMF5wo20vKK0fVzyuqPV6VwR0AQZlZQtmAQHvYPWipzyaV
wc7VaEhVwbvZGVkAGH1AGRlZGVvsK0ed3cwG1ubEjnxRTwUPaJFjHafuq0-mW6S1
IBtSJFwUOe8Dwcwyx-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 mitigiate 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
Intended status: Informational
Expires: September 6, 2018

J. Peterson
Neustar

March 5, 2018

PASSporT Extension for Diverted Calls
draft-ietf-stir-passport-divert-02.txt

Abstract

This document extends PASSporT, which conveys cryptographically-signed information about the people involved in personal communications, 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.

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 6, 2018.

Copyright Notice

Copyright (c) 2018 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. PASSporT 'div' Claim 3
3.1. Nesting the original PASSporT in 'div' 5
4. Using 'div' in SIP 6
4.1. Authentication Service Behavior 6
4.2. Verification Service Behavior 6
5. 'div' and Redirection 7
6. Extending 'div' to work with Service Logic Tracking 8
7. Acknowledgments 9
8. IANA Considerations 9
9. Security Considerations 9
10. Informative References 9
Author's Address 11

1. Introduction

PASSporT [RFC8225] is a token format based on JWT [RFC7519] for conveying cryptographically-signed information about the people involved in personal communications; it is used with STIR [RFC8224] 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 originally 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 as 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 fact that the original called party number is not always the number to which a call is delivered. The address in the To header field value of SIP requests is not supposed to change, accordingly to baseline [RFC3261], as it is the Request-URI that is supposed to be updated when a call is retargeted, but practically speaking some operational environments do alter the To header field. There are a number of potential ways for intermediaries to indicate that such a forwarding operating has taken place. 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 can both capture 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 original called number in PASSporT no longer reflects the destination to which a call is likely to be delivered. Verification services and the relying parties who make authorization decisions about communications may use this indication to confirm that a legitimate retargeting of the call has taken place, rather than a cut-and-paste attack.

2. Terminology

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

3. PASSporT ’div’ Claim

This specification defines a new JSON Web Token claim for "div" 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 (without removing the original PASSporT). Note that a new PASSporT is only necessary when the canonical form of the "dest" identifier (per the canonicalization procedures in [RFC8224] Section 8) changes due to this retargeting. "div" is typically populated with a destination address found in the "dest" field of PASSporT received by the retargeting entity, though it may include other elements as well, including a copy of the original PASSporT. These new PASSporT generated by retargeting entities MUST include the "div" PASSporT type, and an "x5u" field pointing to a credential that the

retargeting entity controls. The new PASSporT header will look as follows:

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

A PASSporT claims object containing "div" is populated with a modification of the original token before the call was retargeted: at a high level, the original identifier for the called party in the "dest" array will become the "div" claim in the new PASSporT. If the "dest" array of the original PASSporT contains multiple identifiers, the retargeting entity MUST select only one them to occupy the "div" field in the new PASSporT. and in particular, 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" array of the new PASSporT. The "orig" value MUST be copied into the new PASSporT from the original PASSporT received by the retargeting entity. The retargeting entity SHOULD retain the "iat" value 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 extension claims should be copied from the original PASSporT to the "div" PASSporT.

So, for an original PASSporT 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 with "div" would look as follows:

```
{ "orig":{"tn":"12155551212"},
  "dest":{"tn":"12155551214"},
  "iat":1443208345,
  "div":{"tn":"121555551213"} }
```

Note that the "div" claim may contain other elements than just a destination, including a copy of the original PASSporT (see Section 3.1). 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.

3.1. Nesting the original PASSporT in 'div'

For some use cases, rather than having multiple unconnected PASSporTS associated with a single call, it makes more sense to nest the PASSporTs, explicitly relating two PASSporTs to one another. For example, when storing a PASSporT with "div" at a Call Placement Service (CPS) for STIR out-of-band [I-D.ietf-stir-oob] scenarios, clients MUST include an "opt" element within "div". "opt" contains the full form of the original PASSporT from which the "div" was generated. If the diverting entity originally received that PASSporT encrypted, it MUST decrypt it before storing it in "opt." The entire "div" PASSporT would than be signed and re-encrypted normally for storage at an out-of-band Call Placement Service (CPS).

A "div" PASSporT containing the "opt" would look as follows:

```
{ "orig":{"tn":"12155551212"},
  "dest":{"tn":"12155551214"},
  "iat":1443208345,
  "div":{"tn":"121555551213",
  "opt":"eyJhbGciOiJFUzI1NiIsInR5cCI6InBhc3Nwb3J0IiwieDV1I \
  joiaHR0cHM6Ly9jZXJ0LmV4YW1wbGUub3JnL3Bhc3Nwb3J0LmNlciJ9.eyJ
  kZXN0Ijp7InVyaSI6WyJzaXA6YWxpY2VAZXhhbXBsZS5jb20iXX0sImlhdC \
  I6IjE0NDMyMDgzNDUiLCJvcmlnIjp7InRuIjoiMTIxNTU1NTEyMTIifX0.r \
  q3pjT1hoRwakEGjHCnWSwUnshd0-zJ6F1VOgFWSjHBr8Qjpjlk-cpFYpFYs \
  ojNCpTzO3QfPOlckGaS6hEck7w"} }
```

The "opt" extension is RECOMMENDED for use within in-band SIP use cases as well. The alternative, having multiple Identity headers in a SIP request, could be confusing for some verification services. However, nested PASSporTs could result in lengthy Identity headers, and some operational experience is needed to ascertain how viable multiple layers of nesting will be.

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 using protocols of 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 containing the "div" PASSporT type to an SIP request that already contains at least one Identity header field; it MUST NOT add a "div" request to an INVITE that contains no other Identity headers fields. Note that the authentication service doing so does not remove or replace any existing Identity header fields, it simply adds a new one. When adding an Identity header field with a PASSporT object containing a "div" claim, SIP authentication services MUST also add a "ppt" parameter to that Identity header with a value of "div". The resulting compact form Identity header field to add to the message might look as follows:

```
Identity: ..sv5CTo05KqpSmtHt3dcEiO/1CWTSZtnG3iV+1nmurLXV/HmtyNS7Ltrg9dlxkWzo
    eU7d7OV8HweTTDobV3itTmgPwCFjaEmMyEI3d7SyN21yNDo2ER/Ovgtw0Lu5csIp
    pPqOg1uXndzHbG7mR6Rl9BnUhHufVRbp51Mn3w0gfUs=; \
    info=<https://biloxi.example.org/biloxi.cer>;alg=ES256;ppt="div"
```

A SIP authentication service typically will derive the new value of "dest" 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 no specific interaction between the "div" mechanism and those SIP header fields. 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 (like History-Info) that record retargeting will likely not require a "div" PASSporT.

4.2. 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 "div" value of "ppt" is as follows.

In order to use the "div" extension, 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" necessary refers to an earlier PASSporT already in the message. In particular, the verification service must find a PASSporT associated

with the call, one created earlier, that contains a "dest" claim with a value equivalent to the "div" claim in the current 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. 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 ultimate destination; this can help the verification service to determine that original PASSporT in the call was not simply used in a cut-and-paste attack. This will help relying parties to make 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.

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 divert PASSporTs to their originals.

5. '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 its "dest" value designates a different original destination. Any intermediary that assigns a new target to a request could choose to redirect with a 3xx response code instead of retargeting. In ordinary operations, a redirection poses no difficult for the operations of baseline STIR: when the UAC receives the 3xx response, it will initiate a new request to the new target (typically 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 requirement 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 to attest that 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; and thus the redirecting UAS needs to include one nested PASSporT per new target. 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 nested "div" PASSporT per each PASSporT in the original request per each Contact URI in the 3xx. Often that may mean just one "div" PASSporT, but for some deployment scenarios, it could require an impractical number of combinations.

STIR-aware intermediaries that redirect requests MAY therefore convey one or more PASSporTs in the backwards direction within Identity headers. This document consequently updates [RFC8224] to permit carrying Identity headers in SIP 300-class responses. It is left to authentication services to determine which Identity headers should be copied into any new requests resulting from the redirection, if any: use of these Identity headers 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.

6. 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 may appear in "div" corresponding to the History-Info header field index parameter value. So for a History-Info header with an index value of "1.2.1", the claims object of the corresponding PASSporT with "div" might look like:

```
{ "orig":{"tn":"12155551212"},
  "dest":{"tn":"12155551214"},
  "iat":1443208345,
  "div":{"tn":"121555551213",
         "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".

7. Acknowledgments

We would like to thank Robert Sparks for contributions to this document.

8. IANA Considerations

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

Claim Name: "div"

Claim Description: New Target of a Call

Change Controller: IESG

Specification Document(s): [RFCThis]

9. 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 the original 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.

10. Informative References

[I-D.ietf-stir-oob]
Rescorla, E. and J. Peterson, "STIR Out-of-Band Architecture and Use Cases", draft-ietf-stir-oob-01 (work in progress), October 2017.

[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>.

[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>.

[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>.

[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>.

[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>.

Author's Address

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

Email: jon.peterson@neustar.biz

Network Working Group
Internet-Draft
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 operating 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":"121555551213"},
  "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 \
 jpbIjEyMTU1NTUxMjE0Il19LCJkaXYiOnsidG4iOiIxMjE1NTU1NTEyMTMifSwiaWF \
 0IjoxNDQzMjA4MzQ1LCJvcmlnIjp7InRuIjoiMTIxNTU1NTEyMTIifX0.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:

```
Identity:eyJhbGciOiJFUzI1NiIsInBwdCI6ImRpdiIsInR5cCI6InBhc3Nwb3J0I \
iwieDV1IjoiaHR0cHM6Ly93d3cuZXhhbXBsZS5jb20vY2VydC5jZXIifQ.eyJkZXN0 \
Ijp7InRuIjpbIjEyMTU1NTUxMjE0Il19LCJkaXYiOnsidG4iOiIxMjE1NTU1NTEyMT \
MifSwiaWF0IjoxNDQzMjA4MzQ1LCJvcmlnIjp7InRuIjoiMTIxNTU1NTEyMTIifX0. \
xBHWipDEEJ8a6TsdX6xUXAnblsFiGUiAxwLiv0HLC9IICj6eG9jQd6WzeSSjHRBwxm \
ChHhVIiMTSqIlk3yCNkg; \
info=<https://www.example.com/cert.cer>;ppt="div"
```

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 in 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":"eyJhbGciOiJFUzI1NiIsInR5cCI6InBhc3Nwb3J0IiwieDV1IjoiaHR0c \
   HM6Ly93d3cuZXhhbXBsZS5jb20vY2VydC5jZXIifQ.eyJkZXN0Ijp7InRuIjpbIj \
   EyMTU1NTUxMjEzIl19LCJpYXQiOjE0NDMyMDgzNDUsIm9yaWciOnsidG4iOiIxMj \
   E1NTU1MTIxMiJ9fQ.1bEzkzcNbKvgz4QoMx0_DJ2T8qFMDC1sPqHPXl1WvbauzRJ \
   RvYlZqQ0qgGTlS8tJ_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:eyJhbGciOiJFUzI1NiIsInBwdCI6ImRpdi1vIiwidHlwIjoicGFzc3Bvc \
nQiLCJ4NXUiOiJodHRwczovL3d3dy5leGFtcGxlLmNvbS9jZXJ0LmNlciJ9.eyJkZX \
N0Ijp7InRuIjoiMTIxNTU1NTEyMTQifSwiZGl2Ijp7InRuIjoiMTIxNTU1NTUxMjEz \
In0sImlhdCI6MTQ0MzIwODM0NSwib3B0IjoiZXlKaGJHY2lPaUpGVXpJMU5pSXNJbl \
I1Y0NJNkluQmhjM053YjNKMElpd2llRFYxSWpvaWFIUjBjSE02THk5M2QzY3VaWGho \
YlhCc1pTNWpiMjB2WTJWeWRDNWpaWElpZlEuZXlKa1pYTjBJanA3SW5SdUlqcGJJak \
V5TVRVMU5UVXhNakV6SWwxOUxDSnBZWFFpT2pFME5ETXlNRGd6TkRVc0ltOXlhV2Np \
T25zaWRHNGlPaUl4TWpFMU5UVTFNVEl4TWlKOWZRLjFiRXpremNOYkt2Z3o0UW9NeD \
BfREoyVDhxRk1EQzFzUHFIUFhsMVd2YmF1elJKUnZZbFpxUTBxZ0dUbFM4dEpfd1hq \
VmUwN1ozd3ZEcmRBcEhoaFl3Iiwib3JpZyI6eyJ0biI6IjEyMTU1NTUxMjEyIn19.C \
HeA9wRnthl7paMe6rP0TARpmFCXjmi_vF_HRz2O_oulB_R-G9xZNiLVvmvHv4gk6LI \
LaDV2y2VtHTLIEgmHig; \
info=<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 fo 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":"121555551213",
         "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-----
MFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAEmzGM1VsO+3IqbMF54rQMaYKQftO4
hUYm9wv5wutLgEd9FsiTy3+4+Wa2O7pffOXPC0QzO+yD8hGEXGP/2mZo6w==
-----END PUBLIC KEY-----

-----BEGIN EC PRIVATE KEY-----
MHcCAQEEIFKCsFZ4Wsw3ZpBxgc4Z0sOjaXDdMk07Ny1fKg6OntAkoAoGCCqGSM49
AwEHoUQDQgAEmzGM1VsO+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

stir
Internet-Draft
Intended status: Standards Track
Expires: September 6, 2018

C. Wendt
Comcast
M. Barnes
MLB@Realtime Communications

March 05, 2018

PASSporT SHAKEN Extension (SHAKEN)
draft-ietf-stir-passport-shaken-01

Abstract

This document extends PASSporT, which is a token object that conveys cryptographically-signed information about the participants involved in communications, to include information defined as part of the SHAKEN specification from ATIS (Alliance for Telecommunications Industry Solutions) and the SIP Forum IP-NNI Joint Task Force. These extensions provide a level of confidence in the correctness of the originating identity for a telephone network that has communications coming from both STIR participating originating communications as well as communications that does not include STIR information.

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 6, 2018.

Copyright Notice

Copyright (c) 2018 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 2
3. Overview of 'shaken' PASSporT extension 3
4. PASSporT 'attest' Claim 3
5. PASSporT 'origid' Claim 4
6. Example 4
7. Using 'shaken' in SIP 5
8. IANA Considerations 5
8.1. JSON Web Token claims 5
8.2. PASSporT Types 6
9. Acknowledgements 6
10. References 6
10.1. Normative References 6
10.2. Informative References 7
Authors' Addresses 7

1. Introduction

The SHAKEN [ATIS-1000074] specification defines a framework for using STIR protocols including PASSporT [RFC8225], RFC4474bis [RFC8224] and the STIR certificate framework [RFC8226] for implementing the cryptographic validation of an authorized originator of telephone calls using SIP. Because the current telephone network contains both VoIP and TDM/SS7 originated traffic, there are many scenarios that need to be accounted for where PASSporT signatures may represent either direct or indirect call origination scenarios. The SHAKEN [ATIS-1000074] specification defines levels of attestation of the origination of the call as well as an origination identifier that can help create a unique association with the origination of calls from various parts of the VoIP or TDM telephone network. This document specifies these indicators as a specified PASSporT extension.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3. Overview of 'shaken' PASSporT extension

The SHAKEN framework is designed to use PASSporT [RFC8225] as a method of asserting the telephone number calling identity. In addition to the PASSporT base claims, there are two additional claims that have been defined for the needs of a service provider to signal information beyond just the telephone identity. First, in order to help bridge the transition of the state of the current telephone network which has calls with no authentication and non-SIP [RFC3261] signaling not compatible with the use of PASSporT and Secure Telephone Identity (STI) in general, there is an attestation claim. This provides three levels of attestation, including a full attestation when the service provider can fully attest to the calling identity, a partial attestation, when the service provider originated a telephone call but can not fully attest to the calling identity, and a gateway attestation which is the lowest level of attestation and represents the service provider receiving a call from a non PASSporT or STI supporting telephone gateway.

The second claim is a unique origination identifier that should be used by the service provider to identify different sources of telephone calls to support a traceback mechanism that can be used for enforcement and identification of a source of illegitimate calls.

The next two sections define these new claims.

4. PASSporT 'attest' Claim

This indicator allows for both identifying the service provider that is vouching for the call as well as clearly indicating what information the service provider is attesting to. The 'attest' claim can be one of the following three values, 'A', 'B', or 'C' as defined in [ATIS-1000074].

'A' represents 'Full Attestation' where the signing provider MUST satisfy all of the following conditions:

- Is responsible for the origination of the call onto the IP based service provider voice network.
- Has a direct authenticated relationship with the customer and can identify the customer.
- Has established a verified association with the telephone number used for the call.

'B' represents 'Partial Attestation' where the signing provider MUST satisfy all of the following conditions:

- Is responsible for the origination of the call onto its IP-based voice network.
- Has a direct authenticated relationship with the customer and can identify the customer.
- Has NOT established a verified association with the telephone number being used for the call.

'C' represents 'Gateway Attestation' where the signing provider MUST satisfy all of the following conditions:

- Is the entry point of the call into its VoIP network.
- Has no relationship with the initiator of the call (e.g., international gateways)

5. PASSporT 'origid' Claim

The purpose of the unique origination identifier is to assign an opaque identifier corresponding to the service provider-initiated calls themselves, customers, classes of devices, or other groupings that a service provider might want to use for determining things like reputation or trace back identification of customers or gateways. The value of 'origid' claim is a UUID as defined in [RFC4122]. SHAKEN isn't prescriptive in the exact usage of origid other than the UUID format as a globally unique identifier representing the originator of the call to whatever granularity the PASSporT signer determines is sufficient for the ability to trace the original origination point of the call. There will likely be best practices documents that more precisely guide it's usage in real deployments.

6. Example

```
Protected Header
{
   "alg":"ES256",
   "typ":"passport",
   "ppt":"shaken",
   "x5u":"https://cert.example.org/passport.cer"
}
Payload
{
   "attest":"A"
   "dest":{"uri":["sip:alice@example.com"]}
   "iat":"1443208345",
   "orig":{"tn":"12155551212"},
   "origid":"123e4567-e89b-12d3-a456-426655440000"
}
```

7. Using 'shaken' in SIP

The use of the 'shaken' PASSporT type and the claims 'attest' and 'origid' are formally defined in [ATIS-1000074] for usage in SIP [RFC3261] aligned with the use of the identity header defined in [RFC8224]. The carriage of the 'attest' and 'origid' values are in the full PASSporT token included in the identity header as specified in [ATIS-1000074].

8. IANA Considerations

8.1. JSON Web Token claims

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

Claim Name: "attest"

Claim Description: Attestation level as defined in SHAKEN framework

Change Controller: IESG

Specification Document(s): [RFCThis]

Claim Name: "origid"

Claim Description: Originating Identifier as defined in SHAKEN framework

Change Controller: IESG

Specification Document(s): [RFCThis]

8.2. PASSporT Types

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

9. Acknowledgements

The authors would like to thank those that helped review and contribute to this document including specific contributions from Jon Peterson, Russ Housley, and Andrew Jurczak. The authors would like acknowledge the work of the ATIS/SIP Forum IP-NNI Task Force to develop the concepts behind this document.

10. References

10.1. Normative References

[ATIS-1000074] ATIS/SIP Forum NNI Task Group, "Signature-based Handling of Asserted information using toKENs (SHAKEN)", January 2017.

[RFC4122] Leach, P., Mealling, M., and R. Salz, "A Universally Unique IDentifier (UUID) URN Namespace", RFC 4122, DOI 10.17487/RFC4122, July 2005, <https://www.rfc-editor.org/info/rfc4122>.

[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

[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>.

Authors' Addresses

Chris Wendt
Comcast
One Comcast Center
Philadelphia, PA 19103
USA

Email: chris-ietf@chriswendt.net

Mary Barnes
MLB@Realtime Communications

Email: mary.ietf.barnes@gmail.com

STIR
Internet-Draft
Intended status: Standards Track
Expires: September 11, 2019

C. Wendt
Comcast
M. Barnes
iconectiv

March 10, 2019

PASSporT SHAKEN Extension (SHAKEN)
draft-ietf-stir-passport-shaken-08

Abstract

This document extends PASSporT, which is a token object that conveys cryptographically-signed information about the participants involved in communications. The extension is defined, corresponding to the SHAKEN specification, to provide both a specific set of levels-of-confidence in the correctness of the originating identity for a SIP based Communication Service Provider (CSP) telephone network originated call as well as an identifier that allows the CSP to uniquely identify the origin of the call within its network.

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 11, 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. Overview of 'shaken' PASSporT extension 3
4. PASSporT 'attest' Claim 4
5. PASSporT 'origid' Claim 4
6. Example "shaken" PASSporT 4
7. Using 'shaken' in SIP 4
8. Order of Claim Keys 4
9. Security Considerations 5
10. Privacy Considerations 5
11. IANA Considerations 6
11.1. JSON Web Token claims 6
11.2. PASSporT Types 6
12. Acknowledgements 6
13. References 7
13.1. Normative References 7
13.2. Informative References 7
Authors' Addresses 8

1. Introduction

The Signature-based Handling of Asserted information using toKENs (SHAKEN) [ATIS-1000074] specification defines a framework for using Secure Telephone Identity Revisited (STIR) protocols including PASSporT [RFC8225], SIP Authenticated Identity Management [RFC8224] and the STIR certificate framework [RFC8226] for implementing the cryptographic validation of an authorized originator of telephone calls using SIP. Because the current telephone network contains both VoIP and TDM/SS7 originated traffic, there are many scenarios that need to be accounted for where PASSporT signatures may represent either direct or indirect call origination scenarios. The SHAKEN [ATIS-1000074] specification defines levels of attestation of the origination of the call as well as an origination identifier that can help create a unique association between the origin of a particular call to the point in the VoIP or TDM telephone network the call came from to identify, for example, either a customer or class of service that call represents. This document specifies these values as claims to extend the base set of PASSporT claims.

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.

In addition, the following terms are used in this document:

- Verified association: is typically defined as an authenticated relationship between a customer and a device that initiated a call on behalf of that customer, for example, a subscriber account with a specific SIM card or set of SIP credentials.

- PASSporT: Defined in [RFC8225] is a JSON Web Token [RFC7519] defined specifically for securing the identity of an initiator of personal communication. This document defines a specific extension to PASSporT.

3. Overview of 'shaken' PASSporT extension

The SHAKEN framework is designed to use PASSporT [RFC8225] as a method of asserting the telephone number calling identity. In addition to the PASSporT base claims, there are two additional claims that have been defined for the needs of a service provider to signal information beyond just the telephone identity. First, in order to help bridge the transition of the state of the current telephone network which has calls with no authentication and non-SIP [RFC3261] signaling not compatible with the use of PASSporT and Secure Telephone Identity (STI) in general, there is an attestation claim. This provides three levels of attestation, including a full attestation when the service provider can fully attest to the calling identity, a partial attestation, when the service provider originated a telephone call but can not fully attest to the calling identity, and a gateway attestation which is the lowest level of attestation and represents the service provider receiving a call from a non-PASSporT and non-STI supporting telephone gateway.

The second claim is a unique origination identifier that should be used by the service provider to identify different sources of telephone calls to support a traceback mechanism that can be used for enforcement and identification of a source of illegitimate calls.

The use of the compact form of PASSporT is not specified in this document and is not specified for use in SHAKEN [ATIS-1000074].

The next two sections define these new claims.

4. PASSporT 'attest' Claim

This indicator allows for both identifying the service provider that is vouching for the call as well as clearly indicating what information the service provider is attesting to. The 'attest' claim can be one of the following three values: 'A', 'B', or 'C'. These values correspond to 'Full Attestation', 'Partial Attestation', and 'Gateway Attestation', respectively. See [ATIS-1000074] for the definitions of these three levels of attestation.

5. PASSporT 'origid' Claim

The purpose of the 'origid' claim is described in [ATIS-1000074]. The value of 'origid' claim is a UUID as defined in [RFC4122]. Please refer to Section 10 for a discussion of the privacy considerations around the use of this value.

6. Example "shaken" PASSporT

Protected Header

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

Payload

```
{
   "attest":"A"
   "dest":{"tn":["12155550131"]}
   "iat":"1443208345",
   "orig":{"tn":"12155550121"},
   "origid":"123e4567-e89b-12d3-a456-426655440000"
}
```

7. Using 'shaken' in SIP

The use of the 'shaken' PASSporT type and the claims 'attest' and 'origid' are formally defined in [ATIS-1000074] for usage in SIP [RFC3261] aligned with the use of the identity header field defined in [RFC8224].

8. Order of Claim Keys

The order of the claim keys MUST follow the rules of [RFC8225] Section 9; the claim keys MUST appear in lexicographic order. Therefore, the claim keys discussed in this document appear in the PASSporT Payload in the following order,

- attest
- dest
- iat
- orig
- origid

9. Security Considerations

This document defines a new PASSporT [RFC8225] extension. The considerations related to the security of the PASSporT object itself are the same as those described in [RFC8225].

[RFC8224] defines how to compare the values of the "dest", "orig" and "iat" claims against fields in a SIP containing a PASSporT as part of validating that request. The values of the new "attest" and "origid" claims added by this extension are not used in such a validation step. They are not compared to fields in the SIP message. Instead, they simply carry additional information from the signer to the consumer of the PASSport. This new information shares the same integrity protection and non-repudiation properties as the base claims in the PASSporT.

10. Privacy Considerations

As detailed in [RFC3261] Section 26, SIP messages inherently carry identifying information of the caller and callee. The addition of STIR cryptographically attests that the signing party vouches for the information given about the callee, as is discussed in the Privacy Considerations of [RFC8224].

SHAKEN [ATIS-1000074] furthermore adds an 'origid' value to the STIR PASSporT, which is an opaque unique identifier representing an element on the path of a given SIP request. This identifier is generated by an originating telephone service provider to identify where within their network (e.g. a gateway or particular service element) a call was initiated; 'origid' can facilitate forensic analysis of call origins when identifying and stopping bad actors trying to spoof identities or make fraudulent calls.

The opacity of the 'origid' claim value is intended to minimize exposure of information about the origination of calls labelled with an 'origid' value. It is therefore RECOMMENDED that implementations generate a unique 'origid' value per call in such a way that only the generator of the 'origid' can determine when two 'origid' values

represent the same or different elements. If deployed systems instead use a common or related 'origid' for service elements in their network, the potential for discovering patterns through correlation of those calls exists. This could allow a recipient of calls to, for instance, learn that a set of callers are using a particular service or coming through a common gateway. It is expected that SHAKEN PASSporTs are shared only within an [RFC3324] trust domain and will be stripped before calls exit that trust domain, but this information still could be used by analytics on intermediary and terminating systems to reveal information that could include geographic location and even device-level information, depending on how the 'origid' is generated.

11. IANA Considerations

11.1. JSON Web Token claims

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

Claim Name: "attest"

Claim Description: Attestation level as defined in SHAKEN framework

Change Controller: IESG

Specification Document(s): [RFCThis]

Claim Name: "origid"

Claim Description: Originating Identifier as defined in SHAKEN framework

Change Controller: IESG

Specification Document(s): [RFCThis]

11.2. PASSporT Types

This specification requests that the IANA add a new entry to the Personal Assertion Token (PASSporT) Extensions registry for the type "shaken" which is specified in [RFCThis].

12. Acknowledgements

The authors would like to thank those that helped review and contribute to this document including specific contributions from Jon Peterson, Russ Housley, Robert Sparks, and Andrew Jurczak. The

authors would like to acknowledge the work of the ATIS/SIP Forum IP-NNI Task Force to develop the concepts behind this document.

13. References

13.1. Normative References

[ATIS-1000074] ATIS/SIP Forum IP-NNI Task Group, "Signature-based Handling of Asserted information using toKENs (SHAKEN)", January 2017, <https://access.atis.org/apps/group_public/download.php/32237/ATIS-1000074.pdf>.

[RFC4122] Leach, P., Mealling, M., and R. Salz, "A Universally Unique IDentifier (UUID) URN Namespace", RFC 4122, DOI 10.17487/RFC4122, July 2005, <https://www.rfc-editor.org/info/rfc4122>.

[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>.

13.2. Informative 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>.

[RFC3323] Peterson, J., "A Privacy Mechanism for the Session Initiation Protocol (SIP)", RFC 3323, DOI 10.17487/RFC3323, November 2002, <https://www.rfc-editor.org/info/rfc3323>.

[RFC3324] Watson, M., "Short Term Requirements for Network Asserted Identity", RFC 3324, DOI 10.17487/RFC3324, November 2002, <https://www.rfc-editor.org/info/rfc3324>.

[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>.

Authors' Addresses

Chris Wendt
Comcast
One Comcast Center
Philadelphia, PA 19103
USA

Email: chris-ietf@chriswendt.net

Mary Barnes
iconectiv

Email: mary.ietf.barnes@gmail.com

STIR
Internet-Draft
Intended status: Standards Track
Expires: August 5, 2018

R. Singh
Vencore Labs
M. Dolly
AT&T
S. Das
Vencore Labs
A. Nguyen
Office of Emergency Communication/DHS

February 01, 2018

PASSporT Extension for Resource-Priority Authorization
draft-ietf-stir-rph-03

Abstract

This document extends the Secure Telephone Identity Revisited (STIR) Personal Assertion Token (PASSporT) specification defined in [I-D.ietf-stir-passport] to allow the inclusion of cryptographically signed assertions of authorization for the values populated in the Session Initiation Protocol (SIP) 'Resource-Priority' header field, which is used for communications resource prioritization.

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 August 5, 2018.

Copyright Notice

Copyright (c) 2018 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. PASSporT 'rph' Claim 3
4. 'rph' in SIP 5
4.1. Authentication Service Behavior 5
4.2. Verification Service Behavior 5
5. Further Information Associated with 'Resource-Priority' . . . 6
6. IANA Considerations 6
6.1. PASSporT Extension Claims Registration 6
6.2. 'rph' Types 7
7. Security Considerations 7
7.1. Avoidance of replay and cut and paste attacks 7
7.2. Solution Considerations 7
7.3. Acknowledgements 8
8. References 8
8.1. Normative References 8
8.2. Informative References 9
Authors' Addresses 9

1. Introduction

PASSporT [I-D.ietf-stir-passport] is a token format based on JSON Web Token (JWT) [RFC7519] for conveying cryptographically signed information about the identities involved in personal communications; it is used with STIR [I-D.ietf-stir-rfc4474bis] to convey a signed assertion of the identity of the participants in real-time communications established via a protocol like SIP [RFC3261]. This specification extends PASSporT to allow cryptographic-signing of the SIP 'Resource-Priority' header field defined in [RFC4412].

[RFC4412] defines the SIP 'Resource-Priority' header field for communications Resource Priority. As specified in [RFC4412], the 'Resource-Priority' header field may be used by SIP user agents [RFC3261], including Public Switched Telephone Network (PSTN) gateways and terminals, and by SIP proxy servers, to influence prioritization afforded to communication sessions, including PSTN calls. However, the SIP 'Resource-Priority' header field could be spoofed and abused by unauthorized entities.

The STIR architecture [RFC7340] assumes that an authority on the originating side of a call provides a cryptographic assurance of the validity of the calling party number in order to prevent impersonation attacks. The STIR architecture allows extensions that can be utilized by authorities supporting real-time communication services using the 'Resource-Priority' header field to cryptographically sign the SIP 'Resource-Priority' header field and convey assertion of the authorization for 'Resource-Priority'. For example, the authority on the originating side verifying the authorization of a particular communication for 'Resource-Priority' can use a PASSPorT claim to cryptographically sign the SIP 'Resource-Priority' header field and convey an assertion of the authorization for 'Resource-Priority'. This will allow a receiving entity (including entities located in different network domains/boundaries) to verify the validity of assertions authorizing 'Resource-Priority'. Cryptographically signed SIP 'Resource-Priority' header fields will allow a receiving entity to verify and act on the information with confidence that the information has not been spoofed or compromised.

This specification documents an optional extension to PASSporT and the associated STIR mechanisms to provide a function to sign the SIP 'Resource-Priority' header field. This PASSporT object is used to provide attestation of a calling user authorization for priority communications. This is necessary in addition to the PASSporT object that is used for calling user telephone number attestation. How the optional extension to PASSporT is used for real-time communications supported using SIP 'Resource-Priority' header field is outside the scope of this document.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119] and in RFC 8174 [RFC8174].

3. PASSporT 'rph' Claim

This specification defines a new JSON Web Token claim for "rph", which provides an assertion for information in SIP 'Resource-Priority' header field.

The creator of a PASSporT object adds a "ppt" value of "rph" to the header of a PASSporT object, in which case the PASSporT claims MUST contain a "rph" claim, and any entities verifying the PASSporT object will be required to understand the "ppt" extension in order to process the PASSporT in question. A PASSPorT header with the "ppt" included will look as follows:

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

The "rph" claim will provide an assertion of authorization, "auth", for information in the SIP 'Resource-Priority' header field (i.e., Resource-Priority = "Resource-Priority": r-value, where r-value= "namespace "." priority value") based on [RFC4412]. Specifically, the "rph" claim includes assertion of the priority-level of the user to be used for a given communication session. The value of the "rph" claim is an Object with one or more keys. Each key is associated with a JSON Array. These arrays contain Strings that correspond to the r-values indicated in the SIP 'Resource-Priority' header field.

The following is an example "rph" claim for a SIP 'Resource-Priority' header field with a r-value ="namespace "." priority value" of "ets.0" and with another r-value= "namespace "." priority value" of "wps.0".

```
 {
  "orig":{"tn":"12155550112"},
  "dest":{["tn":"12125550113"]},
  "iat":"1443208345",
  "rph":{"auth":["ets.0", "wps.0"]}
 }
```

After the header and claims PASSporT objects have been constructed, their signature is generated normally per the guidance in [I-D.ietf-stir-passport] using the full form of PASSPorT. The credentials (e.g., authority responsible for authorizing Resource-Priority) used to create the signature must have authority over the namespace of the "rph" claim and there is only one authority per claim. The authority MUST use its credentials (i.e., CERT) associated with the specific service supported by the SIP namespace in the claim. If r-values are added or dropped by the intermediaries along the path, intermediaries must generate a new "rph" header and sign the claim with its own authority.

The use of the compact form of PASSporT is not specified in this document.

4. 'rph' in SIP

This section specifies SIP-specific usage for the "rph" claim in PASSporT.

4.1. Authentication Service Behavior

The Authentication Service will create the "rph" claim using the values discussed in section 3 based on [RFC4412]. The construction of "rph" claim follows the steps described in Section 4 of [I-D.ietf-stir-rfc4474bis].

The resulting Identity header for "rph" might look as follows(backslashes shown for line folding only):

```
Identity:eyJhbGciOiJFUzI1NiIsInBwdCI6InJwaCIsInR5cCI6InBhc3Nwb3J0\
IiwieDV1IjoiaHR0cHM6Ly93d3cuZXhhbXBsZS5jb20vY2VydC5jZXIifQo.eyJkZ\
XN0Ijp7WyJ0biI6IjEyMTI1NTUwMTEzIl19LCJpYXQiOiIxNDQzMjA4MzQ1Iiwib3\
JpZyI6eyJ0biI6IjEyMTU1NTUwMTEyIn0sInJwaCI6eyJhdXRoIjpbImV0cy4wIiw\
id3BzLjAiXX19Cg.s37S6VC8HM6Dl6YzJeQDsrZcwJ0lizxhUrA7f_98oWBHvo-cl\
-n8MIhoCr18vYYFy3blXvs3fslM_oos2P2Dyw;info=<https://www.example.\
org/cert.cer>;alg=ES256;ppt=rph
```

A SIP authentication service typically will derive the value of "rph" from the 'Resource-Priority' header field based on policy associated with service specific use of the "namespace "." priority value" for r-values based on [RFC4412]. The authentication service derives the value of the PASSPorT claim by verifying the authorization for 'Resource-Priority' (i.e., verifying a calling user privilege for 'Resource-Priority' based on its identity) which might be derived from customer profile data or from access to external services.

[RFC4412] allows multiple "namespace "." priority value" pairs, either in a single SIP 'Resource-Priority' header field or across multiple SIP 'Resource-Priority' headers. An authority is responsible for signing all the content of a SIP 'Resource-Priority' header field for which it has the authority.

4.2. Verification Service Behavior

[I-D.ietf-stir-rfc4474bis] Section 6.2 Step 5 requires that specifications defining "ppt" values describe any additional verifier behavior. The behavior specified for the "ppt" values of "rph" is as follows:

The verification service MUST extract the value associated with the "auth" key in a full form PASSPorT with a "ppt" value of "rph". If the signature validates, then the verification service can use the

value of the "rph" claim as validation that the calling party is authorized for 'Resource-Priority' as indicated in the claim. This value would in turn be used for priority treatment in accordance with local policy for the associated communication service. If the signature validation fails, the verification service should infer that the calling party is not authorized for 'Resource-Priority' as indicated in the claim. In such cases, the priority treatment for the associated communication service is handled as per the local policy.

In addition, [I-D.ietf-stir-rfc4474bis] Section 6.2 Step 4 requires "iat" value in "rph" claim to be verified.

The behavior of a SIP UA upon receiving an INVITE containing a PASSporT object with a "rph" claim will largely remain a matter of implementation policy for the specific communication service. In most cases, implementations would act based on confidence in the veracity of this information.

5. Further Information Associated with 'Resource-Priority'

There may be additional information about the calling party or the call that could be relevant to authorization for 'Resource-Priority'. This may include information related to the device subscription of the caller, or to any institutions that the caller or device is associated with, or even categories of institutions. All of these data elements would benefit from the secure attestations provided by the STIR and PASSporT frameworks. The specification of the "rph" claim could entail the optional presence of one or more such additional information fields.

A new IANA registry has been defined to hold potential values of the "rph" array; see Section 6.2. The definition of the "rph" claim may have one or more such additional information field(s). Details of such "rph" claim to encompass other data elements are left for future version of this specification.

6. IANA Considerations

6.1. PASSporT Extension Claims Registration

This document registers a new "ppt" value for the "Personal Assertion Token (PASSporT) Extensions" table.

- Claim Name: "rph"
- Claim Description: Resource Priority Header Authorization

- Change Controller: IESG

- Specification Document(s): Section 3 of [RFCThis]

6.2. 'rph' Types

This specification also requests that the IANA creates a new registry for "rph" types. Each registry entry must contain two fields: the name of the "rph" type and the specification in which the type is described. This registry is to be initially populated with a single value for "auth" which is specified in [RFCThis]. Registration of new "rph" types shall be under the specification required policy.

7. Security Considerations

The security considerations discussed in [I-D.ietf-stir-rfc4474bis] in Section 10 are applicable here.

7.1. Avoidance of replay and cut and paste attacks

The PASSporT extension with a "ppt" value of "rph" MUST only be sent with SIP INVITE when 'Resource-Priority' header field is used to convey the priority of the communication as defined in [RFC4412]. To avoid the replay, and cut and paste attacks, the procedures described in Section 10.1 of [I-D.ietf-stir-rfc4474bis] MUST be followed.

7.2. Solution Considerations

The use of extension to PASSporT tokens with "ppt" value "rph" based on the validation of the digital signature and the associated certificate requires consideration of the authentication and authority or reputation of the signer to attest to the identity being asserted. The following considerations should be recognized when using PASSporT extension with "ppt" value of "rph":

- An authority (signer) is only allowed to sign the content of a SIP 'Resource-Priority' header field for which it has the right authority. The authority that signs the token MUST have a secure method for authentication of the end user or the device.

- The verification of the signature MUST include means of verifying that the signer is authoritative for the signed content of the resource priority namespace in the PASSporT.

7.3. Acknowledgements

We would like to thank STIR members, ATIS/SIP Forum Task Force on IPNNI members, and the NS/EP Priority Services community for contributions to this problem statement and specification. We would also like to thank David Hancock and Ning Zhang for their valuable inputs.

8. References

8.1. Normative References

[I-D.ietf-stir-passport]
Wendt, C. and J. Peterson, "Personal Assertion Token (PASSporT)", February 2017.

[I-D.ietf-stir-rfc4474bis]
Peterson, J., Jennings, C., Rescorla, E., and C. Wendt, "Authenticated Identity Management in the Session Initiation Protocol (SIP)", February 2017.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, DOI 10.17487/RFC3261, June 2002, <http://www.rfc-editor.org/info/rfc3261>.

[RFC4412] Schulzrinne, H. and J. Polk, "Communications Resource Priority for the Session Initiation Protocol (SIP)", RFC 4412, DOI 10.17487/RFC4412, February 2006, <http://www.rfc-editor.org/info/rfc4412>.

[RFC7519] Jones, M., Bradley, J., and N. Sakimura, "JSON Web Token (JWT)", RFC 7519, DOI 10.17487/RFC7519, May 2015, <http://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, <http://www.rfc-editor.org/info/rfc8174>.

8.2. Informative References

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

Authors' Addresses

Ray P. Singh
Vencore Labs
150 Mount Airy Road
New Jersey, NJ 07920
USA

Email: rsingh@vencorelabs.com

Martin Dolly
AT&T
200 Laurel Avenue
Middletown, NJ 07748
USA

Email: md3135@att.com

Subir Das
Vencore Labs
150 Mount Airy Road
New Jersey, NJ 07920
USA

Email: sdas@vencorelabs.com

An Nguyen
Office of Emergency Communication/DHS
245 Murray Lane, Building 410
Washington, DC 20528
USA

Email: an.p.nguyen@HQ.DHS.GOV

STIR
Internet-Draft
Intended status: Standards Track
Expires: November 25, 2018

R. Singh
Vencore Labs
M. Dolly
AT&T
S. Das
Vencore Labs
A. Nguyen
Office of Emergency Communication/DHS

May 24, 2018

PASSporT Extension for Resource Priority Authorization
draft-ietf-stir-rph-06

Abstract

This document extends the PASSporT (Personal Assertion Token) specification defined in [RFC8225] to allow the inclusion of cryptographically signed assertions of authorization for the values populated in the 'Session Initiation Protocol (SIP) Resource-Priority' header field, which is used for communications resource prioritization.

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 November 25, 2018.

Copyright Notice

Copyright (c) 2018 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. PASSporT 'rph' Claim 3
4. 'rph' in SIP 5
4.1. Authentication Service Behavior 5
4.2. Verification Service Behavior 5
5. Further Information Associated with 'Resource-Priority' . . . 6
6. IANA Considerations 6
6.1. JSON Web Token Claims 6
6.2. PASSporT Types 7
7. Security Considerations 7
7.1. Avoidance of replay and cut and paste attacks 7
7.2. Solution Considerations 7
7.3. Acknowledgements 8
8. References 8
8.1. Normative References 8
8.2. Informative References 9
Authors' Addresses 9

1. Introduction

PASSporT [RFC8225] is a token format based on JSON Web Token (JWT) [RFC7519] for conveying cryptographically signed information about the identities involved in personal communications. PASSporT with STIR [RFC8224] provides a mechanism by which an authority on the originating side of a call via a protocol like SIP [RFC3261] can provide a cryptographic assurance of the validity of the calling party telephone number in order to prevent impersonation attacks.

[RFC4412] defines the 'SIP Resource-Priority' header field for communications 'Resource-Priority'. As specified in [RFC4412], the 'SIP Resource-Priority' header field may be used by SIP user agents [RFC3261] (including Public Switched Telephone Network (PSTN) gateways and SIP proxy servers) to influence prioritization afforded to communication sessions including PSTN calls (e.g., to manage scarce network resources during network congestion scenarios). However, the 'SIP Resource-Priority' header field could be spoofed and abused by unauthorized entities, the threat models and use cases of which are described in [RFC7375] and [RFC7340], respectively.

Compromise of the 'SIP Resource-Priority' header field [RFC4412] could lead to misuse of network resource (i.e., during congestion scenarios) resulting in impacts to the application services supported using the 'SIP Resource-Priority' header field.

[RFC8225] allows extensions by which an authority on the originating side verifying the authorization of a particular communication for 'SIP Resource-Priority' can use a PASSPorT claim to cryptographically sign the 'SIP Resource-Priority' header field and convey assertion of the authorization for 'Resource-Priority'. Signed 'SIP Resource-Priority' header field will allow a receiving entity (including entities located in different network domains/boundaries) to verify the validity of assertions authorizing 'Resource-Priority' and to act on the information with confidence that the information has not been spoofed or compromised.

This specification documents an extension to PASSporT and the associated STIR mechanisms to provide a function to cryptographically sign the 'SIP Resource-Priority' header field. This PASSporT object is used to provide attestation of a calling user authorization for priority communications. This is necessary in addition to the PASSporT object that is used for calling user telephone number attestation. How this extension to PASSporT is used for real-time communications supported using 'SIP Resource-Priority' header field is outside the scope of this document. In addition, the PASSPorT extension defined in this document is intended for use in environments where there are means to verify that the signer of the 'SIP Resource-Priority' header field is authoritative.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119] and in RFC 8174 [RFC8174].

3. PASSporT 'rph' Claim

This specification defines a new JSON Web Token claim for "rph", which provides an assertion for information in 'SIP Resource-Priority' header field.

The creator of a PASSporT object adds a "ppt" value of "rph" to the header of a PASSporT object, in which case the PASSporT claims MUST contain a "rph" claim, and any entities verifying the PASSporT object will be required to understand the "ppt" extension in order to process the PASSporT in question. A PASSPorT header with the "ppt" included will look as follows:

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

The "rph" claim will provide an assertion of authorization, "auth", for information in the 'SIP Resource-Priority' header field based on [RFC4412] and the syntax is:

```
{
Resource-Priority = "Resource-Priority" : r-value,
r-value= namespace  "."  r-priority
}
```

Specifically, the "rph" claim includes an assertion of the priority-level of the user to be used for a given communication session. The value of the "rph" claim is an Object with one or more keys. Each key is associated with a JSON Array. These arrays contain Strings that correspond to the r-values indicated in the 'SIP Resource-Priority' header field.

The following is an example "rph" claim for a 'SIP Resource-Priority' header field with one r-value of "ets.0" and with another r-value of "wps.0":

```
 {
  "orig":{"tn":"12155550112"},
  "dest":{["tn":"12125550113"]},
  "iat":1443208345,
  "rph":{"auth":["ets.0", "wps.0"]}
 }
```

After the header and claims PASSporT objects have been constructed, their signature is generated normally per the guidance in [RFC8225] using the full form of PASSPorT. The credentials (i.e., Certificate) used to create the signature must have authority over the namespace of the "rph" claim and there is only one authority per claim. The authority MUST use its credentials associated with the specific service supported by the resource priority namespace in the claim. If r-values are added or dropped by the intermediaries along the path, intermediaries must generate a new "rph" header and sign the claim with its own authority.

The use of the compact form of PASSporT is not specified in this document.

4. 'rph' in SIP

This section specifies SIP-specific usage for the "rph" claim in PASSporT.

4.1. Authentication Service Behavior

The Authentication Service will create the "rph" claim using the values discussed in section 3 of this document that are based on [RFC4412]. The construction of "rph" claim follows the steps described in Section 4.1 of [RFC8224].

The resulting Identity header for "rph" might look as follows(backslashes shown for line folding only):

```
Identity:eyJhbGciOiJFUzI1NiIsInBwdCI6InJwaCIsInR5cCI6InBhc3Nwb3J0\
IiwieDV1IjoiaHR0cHM6Ly93d3cuZXhhbXBsZS5jb20vY2VydC5jZXIifQo.eyJkZ\
XN0Ijp7WyJ0biI6IjEyMTI1NTUwMTEzIl19LCJpYXQiOiIxNDQzMjA4MzQ1Iiwib3\
JpZyI6eyJ0biI6IjEyMTU1NTUwMTEyIn0sInJwaCI6eyJhdXRoIjpbImV0cy4wIiw\
id3BzLjAiXX19Cg.s37S6VC8HM6Dl6YzJeQDsrZcwJ0lizxhUrA7f_98oWBHvo-cl\
-n8MIhoCr18vYYFy3blXvs3fslM_oos2P2Dyw;info=<https://www.example.\
org/cert.cer>;alg=ES256;ppt="rph"
```

A SIP authentication service will derive the value of "rph" from the 'SIP Resource-Priority' header field based on policy associated with service specific use of the "namespace "." r-priority" for r-values based on [RFC4412]. The authentication service derives the value of the PASSPorT claim by verifying the authorization for 'SIP Resource-Priority' (i.e., verifying a calling user privilege for 'Resource-Priority' based on its identity) which might be derived from customer profile data or from access to external services.

[RFC4412] allows multiple "namespace "." priority value" pairs, either in a single 'SIP Resource-Priority' header field or across multiple 'SIP Resource-Priority' headers. An authority is responsible for signing all the content of a 'SIP Resource-Priority' header field for which it has the authority.

4.2. 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 "rph" is as follows:

The verification service MUST extract the value associated with the "auth" key in a full form PASSPorT with a "ppt" value of "rph". If the signature validates, then the verification service can use the value of the "rph" claim as validation that the calling party is

authorized for 'SIP Resource-Priority' as indicated in the claim. This value would in turn be used for priority treatment in accordance with local policy for the associated communication service. If the signature validation fails, the verification service should infer that the calling party is not authorized for 'SIP Resource-Priority' as indicated in the claim. In such cases, the priority treatment for the associated communication service is handled as per the local policy of the verifier. In such scenarios, 'SIP Resource-Priority' header field SHOULD be stripped from SIP request and the network entities should treat the call as an ordinary call.

In addition, [RFC8224] Section 6.2 Step 4 requires the "iat" value in "rph" claim to be verified.

The behavior of a SIP UA upon receiving an INVITE containing a PASSporT object with a "rph" claim will largely remain a matter of implementation policy for the specific communication service. In most cases, implementations would act based on confidence in the veracity of this information.

5. Further Information Associated with 'Resource-Priority'

There may be additional information about the calling party or the call that could be relevant to authorization for 'SIP Resource-Priority'. This may include information related to the device subscription of the caller, or to any institutions that the caller or device is associated with, or even categories of institutions. All of these data elements would benefit from the secure attestations provided by the STIR and PASSporT frameworks. The specification of the "rph" claim could entail the optional presence of one or more such additional information fields applicable to 'SIP Resource-Priority'.

A new IANA registry has been defined to hold potential values of the "rph" array; see Section 6.2. The definition of the "rph" claim may have one or more such additional information field(s). Details of such "rph" claim to encompass other data elements are left for future version of this specification.

6. IANA Considerations

6.1. JSON Web Token Claims

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

- Claim Name: "rph"

- Claim Description: Resource Priority Header Authorization
- Change Controller: IESG
- Specification Document(s): Section 3 of [RFCThis]

6.2. PASSporT Types

This specification also requests that the IANA creates a new entry to the PASSporT Types registry for the type "rph" which is specified in [RFCThis]. In addition, another registry needs to be created in which each entry must contain two fields: the name of the "rph" type and the specification in which the type is described. This registry is to be initially populated with a single value for "auth" which is specified in [RFCThis]. Registration of new "rph" types shall be under the specification required policy.

7. Security Considerations

The security considerations discussed in [RFC8224] in Section 12 are applicable here.

7.1. Avoidance of replay and cut and paste attacks

The PASSporT extension with a "ppt" value of "rph" MUST only be sent with SIP INVITE when ’Resource-Priority’ header field is used to convey the priority of the communication as defined in [RFC4412]. To avoid replay, and cut and paste attacks, the recommendations provided in Section 12.1 of [RFC8224] MUST be followed.

7.2. Solution Considerations

Using extensions to PASSporT tokens with a "ppt" value of "rph" requires knowledge of the authentication, authorization, and reputation of the signer to attest to the identity being asserted, including validating the digital signature and the associated certificate chain to a trust anchor. The following considerations should be recognized when using PASSporT extensions with a "ppt" value of "rph":

- A signer is only allowed to sign the content of a ’SIP Resource-Priority’ header field for which it has the proper authorization. Before signing tokens, the signer MUST have a secure method for authentication of the end user or the device being granted a token.

- o The verification of the signature MUST include means of verifying that the signer is authoritative for the signed content of the resource priority namespace in the PASSporT.

7.3. Acknowledgements

We would like to thank STIR WG members, ATIS/SIP Forum Task Force on IPNNI members, and the NS/EP Priority Services community for contributions to this problem statement and specification. We would also like to thank David Hancock and Ning Zhang for their valuable inputs.

8. References

8.1. 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, <http://www.rfc-editor.org/info/rfc2119>.

[RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, DOI 10.17487/RFC3261, June 2002, <http://www.rfc-editor.org/info/rfc3261>.

[RFC4412] Schulzrinne, H. and J. Polk, "Communications Resource Priority for the Session Initiation Protocol (SIP)", RFC 4412, DOI 10.17487/RFC4412, February 2006, <http://www.rfc-editor.org/info/rfc4412>.

[RFC7519] Jones, M., Bradley, J., and N. Sakimura, "JSON Web Token (JWT)", RFC 7519, DOI 10.17487/RFC7519, May 2015, <http://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, <http://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, <http://www.rfc-editor.org/info/rfc8224>.

[RFC8225] Wendt, C. and J. Peterson, "PASSporT:Personal Assertion Token", RFC 8225, DOI 10.17487/RFC8225, February 2018, <http://www.rfc-editor.org/info/rfc8225>.

8.2. Informative References

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

[RFC7375] Peterson, J., "Secure Telephone Identity Threat Model", RFC 7375, DOI 10.17487/RFC7375, October 2014, <http://www.rfc-editor.org/info/rfc7375>.

Authors' Addresses

Ray P. Singh
Vencore Labs
150 Mount Airy Road
New Jersey, NJ 07920
USA

Email: rsingh@vencorelabs.com

Martin Dolly
AT&T
200 Laurel Avenue
Middletown, NJ 07748
USA

Email: md3135@att.com

Subir Das
Vencore Labs
150 Mount Airy Road
New Jersey, NJ 07920
USA

Email: sdas@vencorelabs.com

An Nguyen
Office of Emergency Communication/DHS
245 Murray Lane, Building 410
Washington, DC 20528
USA

Email: an.p.nguyen@HQ.DHS.GOV

Network Working Group
Internet-Draft
Intended status: Informational
Expires: September 6, 2018

J. Peterson
Neustar
C. Wendt
Comcast

March 5, 2018

Connected Identity for STIR
draft-peterson-stir-rfc4916-update-00.txt

Abstract

The SIP Identity header conveys cryptographic identity information about the originators of SIP requests. The Secure Telephone Identity Revisited (STIR) framework however provides no means for determining the identity of the called party in a traditional telephone calling scenario. This document updates prior guidance on the "connected identity" problem to reflect the changes to SIP Identity that accompanied STIR, and considers a revised problem space for connected identity as a means of detecting calls that have been retargeted to a party impersonating the intended destination.

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 6, 2018.

Copyright Notice

Copyright (c) 2018 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. Connected Identity Problem Statement for STIR 3
4. Authorization Policy for Callers 4
5. Pre-Association with Destinations 5
6. Updates to RFC4916 5
7. Acknowledgments 5
8. IANA Considerations 6
9. Security Considerations 6
10. Informative References 6
Authors' Addresses 7

1. Introduction

The Session Initiation Protocol (SIP) [RFC3261] initiates sessions, and as a step in establishing sessions, it exchanges information about the parties at both ends of a session. Users review information about the calling party, for example, to determine whether to accept communications initiated by a SIP, in the same way that users of the telephone network assess "Caller ID" information before picking up calls. This information may sometimes be consumed by automata to make authorization decisions.

STIR [RFC8224] provides a cryptographic assurance of the identity of calling parties in order to prevent impersonation, which is a key enabler of unwanted robocalls, swatting, vishing, voicemail hacking, and similar attacks (see [RFC7340]). There also exists a related problem: the identity of the party who answers a call can differ from that of the initial called party for various reasons such as call forwarding, call distribution and call pick-up. It can potentially be difficult to determine why a call reaches a target other than the one originally intended, and whether the party ultimately reached by the call is one that the caller should trust

[RFC4916] allowed a mid-dialog request, such as an UPDATE [RFC3311], to convey what is commonly called "connected identity" information--that is, the identity of the connected user--in either direction within the context of an existing INVITE-initiated dialog. In an update to the original [RFC3261] behavior, [RFC4916] allowed that UPDATE to alter the From header field value for requests in the

backwards direction: previously [RFC3261] required that the From header field values sent in requests in the backwards direction reflect the To header field value of the dialog-forming request, for various backwards-compatibility reasons. In other words, if Alice sent a dialog-forming request to Bob, then under the original [RFC3261] rules, even if that dialog-forming request reached Carol, Carol would still be required to put Bob's identity in the From header field value in any mid-dialog requests in the backwards direction. [RFC4916] furthermore created the "from-change" option tag to negotiate this capability during dialog establishment.

[RFC4916] was created to work with the original SIP Identity [RFC4474] mechanism, as that mechanism only allowed requests to be signed, but not responses. Since a mid-dialog request in the backwards direction can be signed with Identity like any other SIP request, this created a practical problem: Carol, say, would not be able to furnish a key to sign for Bob's identity, if Carol wanted to sign requests in the backwards direction.

This specification updates [RFC4916] to reflect the changes to the SIP Identity header as defined in [RFC4474] made by [RFC8224], and the revised problem space of STIR.

2. Terminology

In this document, the key words "MAY", "MUST, "MUST NOT", "SHOULD", and "SHOULD NOT", are to be interpreted as described in [RFC2119].

3. Connected Identity Problem Statement for STIR

The STIR problem statement [RFC7340] enumerates robocalling, voicemail hacking, vishing, and swatting as problems with the modern telephone number that are enabled, or abetted, by impersonation: by the ability of a calling party to arbitrarily set the identity that will be rendered to end users to identify the caller.

Today, sophisticated adversaries can redirect calls on the PSTN to destinations other than the intended called party. For some call centers, like those associated with financial institutions, healthcare, and emergency services, an attacker could hope to gain valuable information about people or to prevent some classes of important services.

Moreover, on the Internet, the lack of any centralized or even federated routing system for telephone numbers has resulted in deployments where the routing of calls is arbitrary: calls to a telephone numbers might be unceremoniously dumped on a PSTN gateway, they might be sent to a default intermediary that makes forwarding

decisions based on a local flat file, various mechanisms like private ENUM might be consulted, or routing might be determined in some other, domain specific way. While the MODERN framework hopes to foster a more credible story about how to establish authority for telephone numbers on the Internet, in the interim, there are numerous attack surfaces that an adversary could explore to attempt to redirect calls to a particular number to someplace other than the intended destination.

[RFC4916] rightly observed that once a SIP call has been answered, the called party can be replaced by a different party with a different identity due to call transfer, call park and retrieval, and so on. In some cases, due to the presence of a back-to-back user agent, it can be effectively impossible for the calling party to know that this has happened. The problem statement considered for STIR focuses solely on call setup, and whether or not media from the connected party should be rendered to the caller when a dialog has been established. This specification does not consider further any threats that arise from a substitution of the called party.

4. Authorization Policy for Callers

In traditional telephone call, the called party receives an alerting signal and can make a decision about whether or not to pick up a phone. They may have access to displayed information, like "Caller ID", to help them arrive at an authorization decision. The situation is more complicated for callers, however: callers typically expect to be connected to the proper destination and are often holding telephones in a position that would not enable them to see displayed information, if any were available for them to review--and moreover, their most direct response to a security breach would be to hang up the call they were in the middle of placing.

While this specification will not prescribe any user experience associated with placing a call, it assumes that callers have some authorization posture that will result in the right thing happening when the connected identity is not expected. This is analogous to a situation where SRTP negotiation fails because the keys exchanges at the media layer do not match fingerprints exchanged at the signaling layer: when a user requests confidentiality services, and they are available, media should not be exchanged. Thus we assume that users have a way in their interface to require this criticality, on a per-call basis, or perhaps on a per-destination basis. Similarly, users will not always place calls where the connected identity is crucial--but when they do, they should have a way to tell their devices that the call should not be completed if it arrives at an unexpected party.

Ultimately, authorization policy for called parties is difficult to set, as calls can end up at unexpected places for legitimate reasons. Some work has been done to make sure that secure diversion works with STIR, in for example [I-D.ietf-stir-passport-divert]. Those indications can be consumed by on the terminating side by verification services to determine that a call has reached its eventual destination for the right reasons. There is currently no way to expose similar information to the calling party however: only if redirection is used (SIP 3XX responses) instead of retargeting will the originating side participate in setting a new destination for calls.

Future versions of this specification will explore ways that the results of mechanisms like [I-D.ietf-stir-passport-divert] could be communicated back to the originating authentication service.

5. Pre-Association with Destinations

Any connected identity mechanism will work best if the user knows before initiating a call that security services are supported by the destination side. Not every institution that a user wants to connect to securely will support STIR and connected identity out of the gate.

Future versions of this specification will explore how the security features of destinations can be discovered before calls are set up so that calling parties can make more informed authorization decisions. This may reuse mechanisms defined by [I-D.ietf-stir-oob].

6. Updates to RFC4916

[TBD - ways that UPDATEs in the backwards direction can carry additional information in support of the above]

In general, the guidance of RFC4916 remains valid for RFC8224.

The deprecation of the Identity-Info header has a number of implications for RFC4916; all of the protocol examples need to be updated to reflect that.

7. Acknowledgments

We would like to thank YOU for your contributions to this specification.

8. IANA Considerations

This memo includes no request to IANA.

9. Security Considerations

TBD.

10. Informative References

[I-D.ietf-modern-problem-framework]
Peterson, J. and T. McGarry, "Modern Problem Statement, Use Cases, and Framework", draft-ietf-modern-problem-framework-03 (work in progress), July 2017.

[I-D.ietf-stir-oob]
Rescorla, E. and J. Peterson, "STIR Out-of-Band Architecture and Use Cases", draft-ietf-stir-oob-01 (work in progress), October 2017.

[I-D.ietf-stir-passport-divert]
Peterson, J., "PASSporT Extension for Diverted Calls", draft-ietf-stir-passport-divert-01 (work in progress), October 2017.

[I-D.peterson-modern-teri]
Peterson, J., "An Architecture and Information Model for Telephone-Related Information (TeRI)", draft-peterson-modern-teri-03 (work in progress), July 2017.

[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>.

[RFC3311] Rosenberg, J., "The Session Initiation Protocol (SIP) UPDATE Method", RFC 3311, DOI 10.17487/RFC3311, October 2002, <https://www.rfc-editor.org/info/rfc3311>.

[RFC4474] Peterson, J. and C. Jennings, "Enhancements for Authenticated Identity Management in the Session Initiation Protocol (SIP)", RFC 4474, DOI 10.17487/RFC4474, August 2006, <https://www.rfc-editor.org/info/rfc4474>.

[RFC4916] Elwell, J., "Connected Identity in the Session Initiation Protocol (SIP)", RFC 4916, DOI 10.17487/RFC4916, June 2007, <https://www.rfc-editor.org/info/rfc4916>.

[RFC7159] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", RFC 7159, DOI 10.17487/RFC7159, March 2014, <https://www.rfc-editor.org/info/rfc7159>.

[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>.

[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>.

Authors' Addresses

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

Email: jon.peterson@neustar.biz

Chris Wendt
Comcast
One Comcast Center
Philadelphia, PA 19103
USA

Email: chris-ietf@chriswendt.net

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 13, 2022

J. Peterson
Neustar
C. Wendt
Comcast

July 12, 2021

Connected Identity for STIR
draft-peterson-stir-rfc4916-update-04

Abstract

The SIP Identity header conveys cryptographic identity information about the originators of SIP requests. The Secure Telephone Identity Revisited (STIR) framework however provides no means for determining the identity of the called party in a traditional telephone calling scenario. This document updates prior guidance on the "connected identity" problem to reflect the changes to SIP Identity that accompanied STIR, and considers a revised problem space for connected identity as a means of detecting calls that have been retargeted to a party impersonating the intended destination, as well as the spoofing of mid-dialog or dialog-terminating events by intermediaries or third parties.

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 13, 2022.

Copyright Notice

Copyright (c) 2021 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. Connected Identity Problem Statement for STIR 4
4. Connected Identity in Provisional Dialogs 5
5. Connected Identity in Mid-Dialog and Dialog-Terminating Requests 6
6. Interaction with Divert PASSporT 7
7. Authorization Policy for Callers 7
8. Pre-Association with Destinations 8
9. Connected Identity and Conferencing 9
10. Examples 9
11. Updates to RFC4916 9
12. Acknowledgments 9
13. IANA Considerations 9
14. Security Considerations 10
15. References 10
15.1. Informative References 10
15.2. Informative References 11
Authors' Addresses 12

1. Introduction

The Session Initiation Protocol (SIP) [RFC3261] initiates sessions, and as a step in establishing sessions, it exchanges information about the parties at both ends of a session. Users review information about the calling party, for example, to determine whether to accept communications initiated by a SIP, in the same way that users of the telephone network assess "Caller ID" information before picking up calls. This information may sometimes be consumed by automata to make authorization decisions.

STIR [RFC8224] provides a cryptographic assurance of the identity of calling parties in order to prevent impersonation, which is a key enabler of unwanted robocalls, swatting, vishing, voicemail hacking, and similar attacks (see [RFC7340]). There also exists a related problem: the identity of the party who answers a call can differ from that of the initial called party for various innocuous reasons such as call forwarding, but in certain network environments, it is

possible for attackers to hijack the route of a called number and direct it to a resource controlled by the attacker. It can potentially be difficult to determine why a call reached a target other than the one originally intended, and whether the party ultimately reached by the call is one that the caller should trust. The lack of mutual authentication of parties moreover makes it possible for outside attackers to inject forged messages (e.g. BYE) into a SIP session.

The property of providing identity in the backwards direction of a call is here called "connected identity." Previous work on connected identity focused on fixing the core semantics of SIP. [RFC4916] allowed a mid-dialog request, such as an UPDATE [RFC3311], to convey identity in either direction within the context of an existing INVITE-initiated dialog. In an update to the original [RFC3261] behavior, [RFC4916] allowed that UPDATE to alter the From header field value for requests in the backwards direction: previously [RFC3261] required that the From header field values sent in requests in the backwards direction reflect the To header field value of the dialog-forming request, for various backwards-compatibility reasons. Under the original [RFC3261] rules, if Alice sent a dialog-forming request to Bob, then even if Bob's SIP service forwarded that dialog-forming request to Carol, Carol would still be required to put Bob's identity in the From header field value in any mid-dialog requests in the backwards direction.

One of the original motivating use cases for [RFC4916] was the use of connected identity with the SIP Identity [RFC4474] header field. While a mid-dialog request in the backwards direction (e.g. UPDATE) can be signed with Identity like any other SIP request, forwarded requests would not be signable without the ability to change the mid-dialog From header field value: Carol, say, would not be able to furnish a key to sign for Bob's identity, if Carol wanted to sign requests in the backwards direction. Carol would however be able to sign for her own identity in the From header field value, if mid-dialog requests in the backwards direction were permitted to vary from the original To header field value.

With the obsolence of [RFC4474] by [RFC8224], this specification updates [RFC4916] to reflect the changes to the SIP Identity header and the revised problem space of STIR. It also explores some new features that would be enabled by connected identity for STIR, including the use of connected identity to prevent route hijacking and to notify callers when an expected called party has successfully been reached. This document also addresses concerns about applying [RFC4916] connected identity to STIR discussed in the SIPBRANDY framework [RFC8862].

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. Connected Identity Problem Statement for STIR

The STIR problem statement [RFC7340] enumerates robocalling, voicemail hacking, vishing, and swatting as problems with the modern telephone network that are enabled, or abetted, by impersonation: by the ability of a calling party to arbitrarily set the telephone number that will be rendered to end users to identify the caller.

Today, sophisticated adversaries can redirect calls on the PSTN to destinations other than the intended called party. For some call centers, like those associated with financial institutions, healthcare, and emergency services, an attacker could hope to gain valuable information about people or to prevent some classes of important services. Moreover, on the Internet, the lack of any centralized or even federated routing system for telephone numbers has resulted in deployments where the routing of calls is arbitrary: calls to telephone numbers might be unceremoniously dumped on a PSTN gateway, they might be sent to a default intermediary that makes forwarding decisions based on a local flat file, various mechanisms like private ENUM [RFC6116] might be consulted, or routing might be determined in some other, domain-specific way. In short, there are numerous attack surfaces that an adversary could explore to attempt to redirect calls to a particular number to someplace other than the intended destination.

Another motivating use case for connected identity is mid-dialog requests, including BYE. The potential for an intermediary to generate a forged BYE in the backwards direction has always been built-in to the stateful dialog management of SIP. For example, there is a class of mobile fraud attacks ("short-stopping") that rely on intermediary networks making it appear as if a call has terminated to one side, while maintaining that the call is still active to the other, in order to create a billing discrepancy that could be pocketed by the intermediary. If BYE requests in both directions of a SIP dialog could be authenticated with STIR, just like dialog-forming requests, then another impersonation vector leading to fraud in the telephone network could be shut down.

There are however practical limits to what securing the signaling can achieve. [RFC4916] rightly observed that once a SIP call has been

answered, the called party can be replaced by a different party (with a different identity) due to call transfer, call park and retrieval, and so on. In some cases, due to the presence of a back-to-back user agent, it can be effectively impossible for the calling party to know that this has happened. The problem statement considered for STIR focuses solely on signaling, not whether media from the connected party should be rendered to the caller when a dialog has been established. This specification does not consider further any threats that arise from a substitution of media.

4. Connected Identity in Provisional Dialogs

[RFC4916] identified a means of sending Identity header field values in the backwards direction before a final response to a dialog has been received by the UAC. It relied on the use of the UPDATE method to send the connected identity in the backwards direction after the UAS had received and responded to a PRACK [RFC3262] from the UAC, which would in turn have been triggered by a provisional 1xx response sent earlier by the UAC. [RFC4916] permits the From header field of the UPDATE to change the address of record of the recipient: if the original INVITE had been sent with a To header field value of "sip:bob@example.com", the UAS in its UPDATE could set the From header field value to "sip:carol@example.com." For STIR, this is a very important property, as Carol might not even possess a credential that can legitimately sign for Bob.

Per [RFC3262], UAC's that require connected identity MUST thus send a Require header field with the option tag 100rel in INVITEs in addition to an Identity header field value containing a PASSporT. UAS's that support this mechanism will first send a Require header field with the option tag 100rel in 1xx class responses to INVITEs that they receive, along with the necessary RSeq header field. The UAC will send a PRACK when it receives the reliable 1xx response from the UAS; the UAS, upon receiving a PRACK, responds with a 200 OK. At this point, the terminating UA is free to send an UPDATE [RFC3311] request in the backwards direction to the originating UA. This update will contain an Identity header, with a PASSporT that signs for the connected identity in its "orig" claim, which typically corresponds to the From header field value of the UPDATE request. If the PASSporT is valid, the originating UA will respond with an OK, and may perform any behaviors associated with the updated identity (see Section 7). Even if connected identity is not required by the originator of an INVITE request, it can still be solicited if available by sending the 100rel option tag in a Supported header field when sending an INVITE with an Identity header, which will trigger the preceding flow if the UAS supports connected identity.

Usually, the updated Identity reflects a changed to the From header field value. But in many operating environments, the From header field value does not contain the identity of the caller that has been asserted by the network, which is instead conveyed by the P-Asserted-Identity header field [RFC3325]. The contents of PAID are not used for dialog matching, and so in environments where PAID is used, it can be altered more dynamically. However, in order for the connected identity and a PASSporT to be conveyed in the backwards direction, a provisional dialog still needs to be established, and an UPDATE sent: in this case, it will be the UDPATE that contains the connected identity in its P-Asserted-Identity header field value, and that value will be signed by the PASSporT in its "orig" claim.

5. Connected Identity in Mid-Dialog and Dialog-Terminating Requests

The use of the connected identity mechanism here specified is not limited to provisional dialog requests. Once a dialog has been established with connected identity, any re-INVITEs from either the originating and terminating side, as well as any BYE requests, MUST contain Identity headers with valid PASSporTs based on the current To/From header field values for the dialog. This prevents third-parties from spoofing any mid-dialog requests in order to redirect media or similarly interfere with communications, as well as preventing denial of service teardowns by attackers.

Theoretically, any SIP requests in a dialog could be signed in this fashion, though it is unclear how valuable it would be for some (e.g. OPTIONS). Requests with specialized payloads such as INFO or MESSAGE, however, would require additional specification for how integrity protection for their bodies could be implemented. Some work has been done toward that for MESSAGE (see [I-D.peterson-stir-messaging]. This specification thus does not mandate PASSporTs for any requests sent in a dialog other than INVITE, UPDATE, and BYE.

It might seem tempting to require that, if an INVITE has been with an Identity header containing a PASSporT, any CANCEL request received for the dialog initiated by that INVITE must also contain an Identity header with a PASSporT. However, CANCEL requests can also sent be sent by stateful proxy servers engaged in parallel forking; for example, when branches need to be canceled because a final response has been received from a UAS. It is however REQUIRED by this specification that if a UAC sends a CANCEL for its own PASSporT-protected INVITE request, that it include an Identity header with a valid PASSporT in the CANCEL. UAS policy will have to determine the instances where it will accept unsigned CANCEL requests for a dialog initiated with a signed INVITE.

6. Interaction with Divert PASSporT

Many of the use cases that motivate connected identity involve retargeting: when a call acquires a new target (in its Request-URI) during transit, the terminating side needs a way to express to the originating side which destination the INVITE reached. In STIR, the "div PASSporT type [RFC8946] was created to securely record when a call was retargeted from one destination to another. Those "div" PASSporTs can be consumed on the terminating side by verification services to determine that a call has reached its eventual destination for the right reasons.

As specified in [RFC8946], the only way those diversion PASSporTs will be seen by the calling party is if redirection is used (SIP 3XX responses) instead of retargeting; while some network policies may want to conceal service logic from the originating party, sending redirections in the backwards direction is the only current defined way for secure indications of redirection to be revealed to the calling party. That in turn would allow the calling user agent to have a strong assurance that legitimate entities in the call path caused the request to reach a party that the caller did not anticipate.

This specification introduces another alternative. When per Section 4 the terminating side sends an UPDATE with an Identity header containing a PASSporT for its current From (or PAID) header field value, it MAY include in Identity header field values any "div" PASSporTs it received in the INVITE that initiated this provisional dialog. These "div" PASSporTs will enable the originating side to receive a secure assurance that the call is being fielded by the proper recipient per the routing of the call. Note however that "div" is not universally supported, and thus calls may be retargeted with generating a "div" PASSporT, so sending those PASSporTs in the backwards direction cannot be mandated. Also note that this will potentially reveal service logic to the called party.

7. Authorization Policy for Callers

In a traditional telephone call, the called party receives an alerting signal and can make a decision about whether or not to pick up a phone. They may have access to displayed information, like "Caller ID", to help them arrive at an authorization decision. The situation is more complicated for callers, however: callers typically expect to be connected to the proper destination and are often holding telephones in a position that would not enable them to see displayed information, if any were available for them to review--and moreover, their most direct response to a security breach would be to hang up the call they were in the middle of placing.

While this specification will not prescribe any user experience associated with placing a call, it assumes that callers might have some way to a set an authorization posture that will result in the right thing happening when the connected identity is not expected. This is analogous to a situation where SRTP negotiation fails because the keys exchanges at the media layer do not match fingerprints exchanged at the signaling layer: when a user requests confidentiality services, and they are unavailable, media should not be exchanged. Thus we assume that users have a way in their interface to require this criticality, on a per-call basis, or perhaps on a per-destination basis, that would cause their user agent to send the INVITE with a Require for 100rel. Similarly, users will not always place calls where the connected identity is crucial--but when they do, they should have a way to tell their devices that the call should not be completed if it arrives at an unexpected party.

Ultimately, authorization policy for called parties is difficult to set, as calls can end up at unexpected places for legitimate reasons. Some work has been done to make sure that secure diversion works with STIR, as described in Section 6.

8. Pre-Association with Destinations

Any connected identity mechanism will work best if the user knows before initiating a call that connected identity is supported by the destination side. Not every institution that a user wants to connect to securely will support STIR and connected identity out of the gate.

The user interface of modern smartphones support an address book from which users select telephone numbers to dial. Even when dialing a number manually, the interface frequently checks the address book and will display to users any provisioned name for the target of the call if one exists. Similarly, when clicking on a telephone number viewed on a web page, or similar service, smartphones often prompt users approve the access to the outbound dialer. These sorts of decision points, when the user is still interacting with the user interface, provide an opportunity to form a pre-association with the destination, and potentially even to exchange STIR PASSporTs in order to validate whether or not the expected destination can be reached securely. Again, this is probably most meaningful for contacting financial, government, or emergency services, for cases where reaching an unintended destination may have serious consequences.

The establishment of media-less dialogs has long been specified as a component of third-party call control in SIP [RFC3375], in which an INVITE is sent with no SDP. Similar media-less dialogs have been proposed for certain automata per [RFC5552]. In the STIR context, a media-less dialog is established by sending an INVITE with an

Identity header but no SDP. STIR-aware UAS's that support this specification, upon receiving an INVITE with no SDP, carrying a PASSporT with a 100rel in the Require header field value, SHOULD follow the mechanism described in Section 4 to send a provisional response and then an UPDATE carrying a PASSporT in the backwards direction. The PASSporT received in the backwards direction could be rendered to the originating user to help them decide if they want to place the call.

[More TBD. In some environments, that may require the use of mechanisms defined by [RFC8816].]

9. Connected Identity and Conferencing

The establishment of connected identity when there are more than two parties in a session will depend on the conferencing mechanism used.

[More TBD.]

10. Examples

[TBD: Revise RFC4916 examples to show new Identity header present in UPDATE and in a backwards-direction BYE.]

11. Updates to RFC4916

[TBD - ways that UPDATEs in the backwards direction can carry additional information in support of the above]

In general, the guidance of RFC4916 remains valid for RFC8224.

The deprecation of the Identity-Info header has a number of implications for RFC4916; all of the protocol examples need to be updated to reflect that.

12. Acknowledgments

We would like to thank YOU for your contributions to this specification.

13. IANA Considerations

This memo includes no request to IANA.

14. Security Considerations

TBD.

15. References

15.1. Informative 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>.

[RFC3262] Rosenberg, J. and H. Schulzrinne, "Reliability of Provisional Responses in Session Initiation Protocol (SIP)", RFC 3262, DOI 10.17487/RFC3262, June 2002, <https://www.rfc-editor.org/info/rfc3262>.

[RFC3311] Rosenberg, J., "The Session Initiation Protocol (SIP) UPDATE Method", RFC 3311, DOI 10.17487/RFC3311, October 2002, <https://www.rfc-editor.org/info/rfc3311>.

[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>.

[RFC3375] Hollenbeck, S., "Generic Registry-Registrar Protocol Requirements", RFC 3375, DOI 10.17487/RFC3375, September 2002, <https://www.rfc-editor.org/info/rfc3375>.

[RFC4474] Peterson, J. and C. Jennings, "Enhancements for Authenticated Identity Management in the Session Initiation Protocol (SIP)", RFC 4474, DOI 10.17487/RFC4474, August 2006, <https://www.rfc-editor.org/info/rfc4474>.

[RFC4916] Elwell, J., "Connected Identity in the Session Initiation Protocol (SIP)", RFC 4916, DOI 10.17487/RFC4916, June 2007, <https://www.rfc-editor.org/info/rfc4916>.

[RFC7159] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", RFC 7159, DOI 10.17487/RFC7159, March 2014, <https://www.rfc-editor.org/info/rfc7159>.

[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>.

[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>.

[RFC8396] Peterson, J. and T. McGarry, "Managing, Ordering, Distributing, Exposing, and Registering Telephone Numbers (MODERN): Problem Statement, Use Cases, and Framework", RFC 8396, DOI 10.17487/RFC8396, July 2018, <https://www.rfc-editor.org/info/rfc8396>.

[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>.

[RFC8862] Peterson, J., Barnes, R., and R. Housley, "Best Practices for Securing RTP Media Signaled with SIP", BCP 228, RFC 8862, DOI 10.17487/RFC8862, January 2021, <https://www.rfc-editor.org/info/rfc8862>.

[RFC8946] Peterson, J., "Personal Assertion Token (PASSporT) Extension for Diverted Calls", RFC 8946, DOI 10.17487/RFC8946, February 2021, <https://www.rfc-editor.org/info/rfc8946>.

15.2. Informative References

[I-D.peterson-stir-messaging]
Peterson, J. and C. Wendt, "Messaging Use Cases and Extensions for STIR", draft-peterson-stir-messaging-01 (work in progress), February 2021.

[RFC5552] Burke, D. and M. Scott, "SIP Interface to VoiceXML Media Services", RFC 5552, DOI 10.17487/RFC5552, May 2009, <https://www.rfc-editor.org/info/rfc5552>.

[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>.

Authors' Addresses

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

Email: jon.peterson@team.neustar

Chris Wendt
Comcast
One Comcast Center
Philadelphia, PA 19103
USA

Email: chris-ietf@chriswendt.net

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 2, 2018

J. Rosenberg
C. Jennings
Cisco Systems
March 1, 2018

Bootstrapping STIR Deployments with Self-Signed Certs and Callbacks
draft-rosenberg-stir-callback-00

Abstract

Robocalling has become an increasing problem in the Public Switched Telephone Network (PSTN). A partial remedy for it is the provision of an authenticated caller ID in the PSTN, which today is lacking. Secure Telephone Identity Revisited (STIR) provides this through the usage of signed payloads in Session Initiation Protocol (SIP) calls. However, STIR deployment requires a global certificate system which allows for worldwide issuance of certifications that attest to which numbers a provider is responsible for. Such a system is likely to take years to rollout. To accelerate STIR deployment, this draft proposes a technique wherein STIR can be used without certificates that attest to number ownership. This is done through a combination of self-signed certificates, reverse callbacks and cached validations.

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 2, 2018.

Copyright Notice

Copyright (c) 2018 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. Problem Statement 2
2. Terminology 3
3. Overview of Operation 4
4. Interactions with RFC 8226 8
5. SS7 Interactions 9
6. Formal Protocol Specification 9
6.1. Originating Agent Behavior 9
6.1.1. On Receipt of incoming INVITE 9
6.1.2. On Receipt of a Verifying INVITE 10
6.2. Terminating Agent Behavior 10
6.2.1. On Receipt of Incoming INVITE 10
6.2.2. On Receipt of a Response to the Verifying INVITE . . 11
6.2.3. On expiration of the timer 12
7. Security Considerations 12
7.1. Attacks from the Calling Agent 12
7.2. Attacks from the Called Agent 13
7.3. Attacks from the agent receiving the Verifying INVITE . . 13
8. IANA Considerations 13
8.1. sip-verify Option Tag 14
8.2. Response Code 471 14
8.3. Response Code 472 14
8.4. Verify-Call Header 14
9. Acknowledgments 15
10. References 15
10.1. Normative References 15
10.2. Informative References 15
Authors' Addresses 16

1. Problem Statement

Robocalling has become an increasing problem in the Public Switched Telephone Network (PSTN). Efforts to prevent it - such as the do-not-call list - have so far proven ineffective. Recently, robocallers have gotten even more crafty, and are tailoring the caller ID of incoming calls to match the area codes and exchanges of

the recipients in order to increase the likelihood that targets pick up the phone.

Part of the reason robocalling is possible is that the PSTN doesn’t provide a way to authenticate caller ID. This problem has gotten worse through the deployment of the Session Initiation Protocol (SIP) [RFC3261] along with widespread availability of APIs (as an example, Twilio), which allow third parties to easily, at low cost, place calls with desired caller IDs to anywhere in the world.

To remedy this, the Secure Telephony Identity (STIR) working group has undertaken to provide a way for e2e authenticated caller ID in SIP-based networks [RFC8224] [RFC8225] [RFC8226]. The core concept is to enable a signature over the SIP INVITE, the signature covering key SIP fields including the From header field containing the caller ID. The signature uses a certificate which is signed by an entity to whom the target has a trust chain, and more importantly, the certificate claims as part of its structure, the phone numbers that the calling party is permitted to claim.

The primary challenge to deployment of STIR is the certification process. It requires a global certification system which can issue certificates to providers across the world, and furthermore, has the processes and database accesses required to assert the set of phone numbers owned by any carrier using the system. This is likely to require coordination amongst telcos, governments, regulators, and telco providers across the globe. Its scope of complexity is similar to ENUM [RFC2916] , which required a similar global infrastructure. ENUM was never successfully deployed.

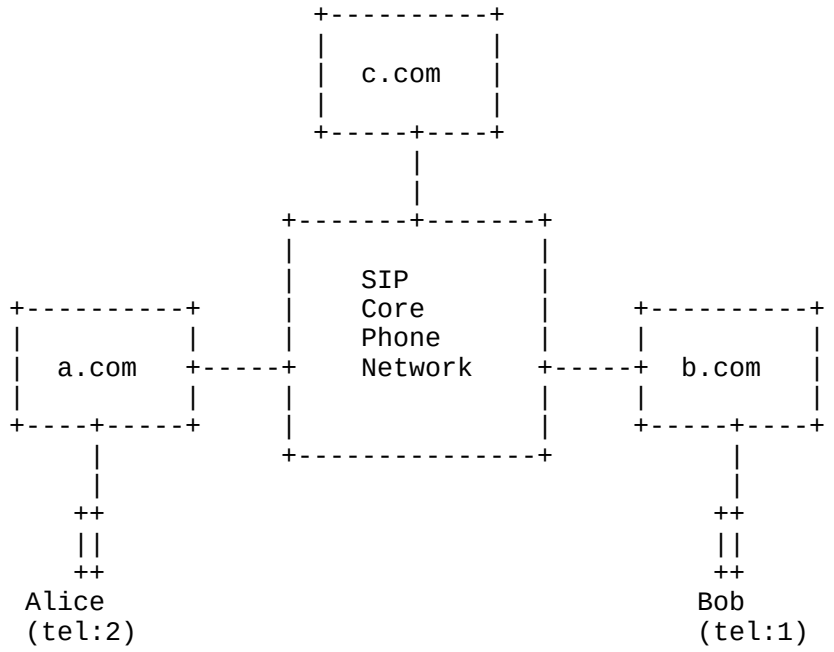
This document proposes a way to accelerate STIR deployments by relaxing the need for any such certification authority. It works with traditional self-signed certificates, and requires only that the calling domain and receiving domain support the protocol defined in this specification. This makes it much easier to deploy. If and when certificates with number ownership are deployed, they can easily co-exist with this proposal, phasing it out over time.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3. Overview of Operation

Consider the following reference architecture:



Alice and Bob are telephone subscribers with phone numbers 2 and 1 respectively, using service providers a.com and b.com respectively. These two providers are connected to each other over a SIP network, which provides routing of calls between providers. A key assumption in this proposal is that this core network accurately routes calls to a specific number in a way which attackers cannot circumvent easily. It also assumes that sufficient portions of this core phone network are now SIP based, enabling delivery of SIP extension values between the originating and terminating providers. This second constraint is identical to in-band STIR. Note however that this proposal does not require SIP to the endpoints; it only assumes SIP between the originating and terminating call agents. While those agents could be SIP proxies or B2BUA, they could also be traditional circuit switched agents with SIP interfaces. We refer to this generically as a call agent.

Alice places a call to Bob’s telephone number. It arrives at Alice’s agent - the calling agent. The calling agent has a self-signed certificate (the solution also works with traditional domain based

certificates). Alice's agent uses this certificate to sign the INVITE as specified in [RFC8224] and [RFC8225]. The INVITE includes a Supported header field with the value stir-callback.

This passes through the core SIP network, which ultimately delivers the call to the receiving agent based on traditional SIP routing logic.

When the call arrives at Bob's agent, it verifies the signature per [RFC8224]. Bob's agent maintains a cache, called the validation cache, which is a mapping from caller IDs to public keys. When the call arrives, Bob checks whether the caller ID matches an entry in the cache. If there is no match - which is the case for the first call from this caller ID - Bob's agent performs a verifying callback to check the validity of the caller ID.

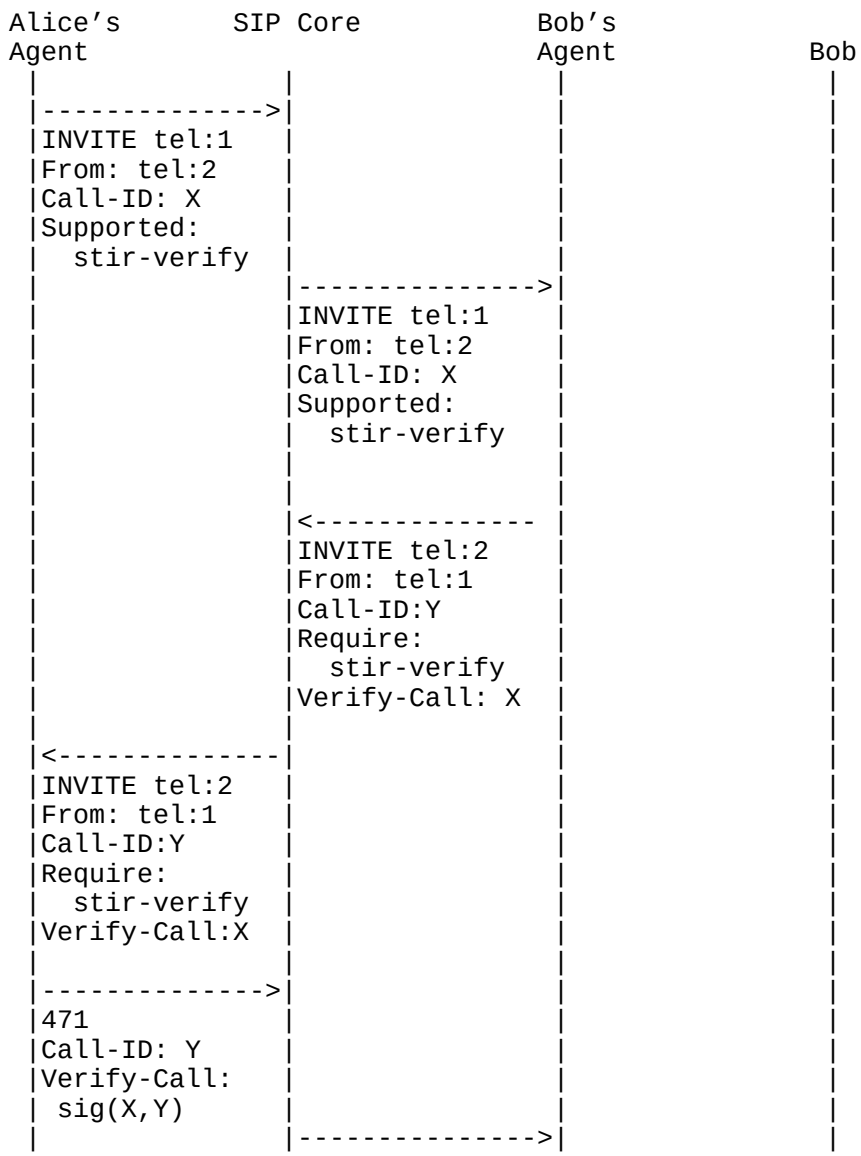
To perform this callback, Bob's agent holds onto the incoming INVITE from Alice, and generates a completely separate INVITE, targeted back towards the number from the incoming caller ID. The verifying INVITE includes a Require header field with the value stir-callback. It also includes SDP, though the contents of this SDP are not relevant as they will never be used. The verifying INVITE also includes the Verify-Call header field. This header field is populated with value taken from the Identity header field of the incoming INVITE from Alice.

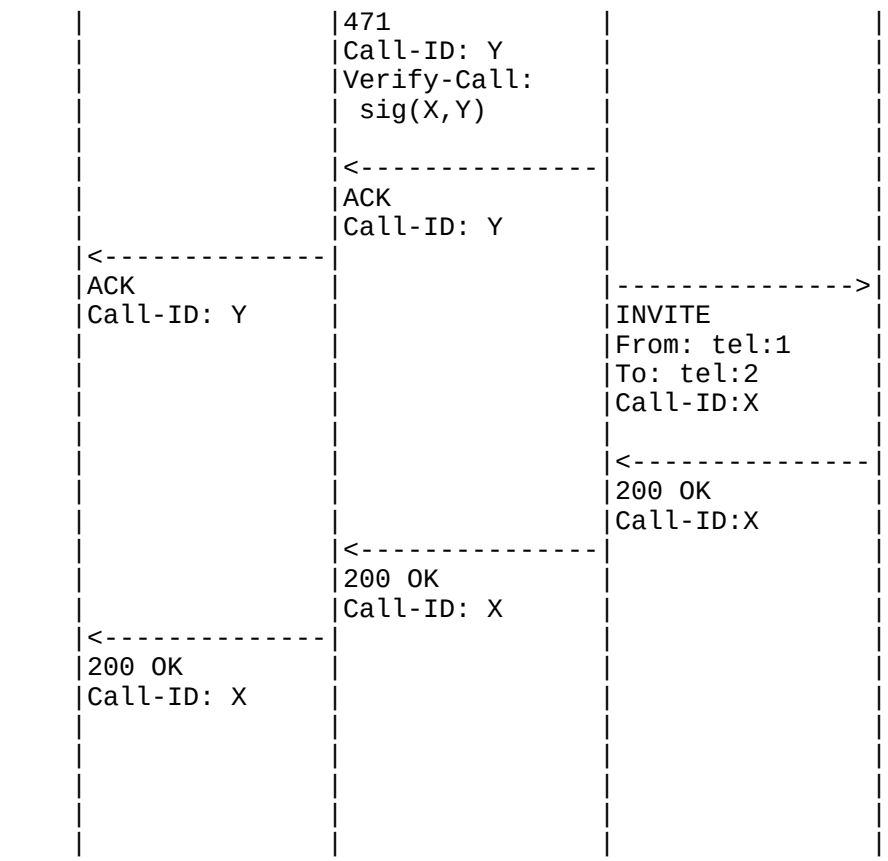
The SIP core network will route the verifying INVITE towards the agent which owns Alice's number. There are three possible cases to consider.

1. The CallerID was correct. In this case, the verifying INVITE will return to one of Alice's call agents. The agent sees the presence of the Require: stir-callback header field. This tells the agent that this is not actually a real call to be completed towards Alice, but rather, a verifying callback to check that Alice's agent really meant to place the original call. As such, Alice's agent extracts the certifcate and signature values from the Verify-Call header field, and checks if they reprsent a valid certificate for signatures from Alice. If it is correct, Alice's agent rejects the INVITE with a 471 response code. This is a new response code which means the call itself should not proceed, but the receiving agent recognizes the the information in the Verify-Call header field as valid. Alice's agent creates a signature over the Call-ID in the incoming INVITE as well as the value in the Verify-Call header field, and includes this signature in the response, in the Verify-Call header field. When this error code reaches Bob's agent, Bob's agent verifies the signature using the public key from the inbound INVITE. Once this has verified,

Bob's agent knows that the caller-ID in the original INVITE was valid. Bob's agent adds the caller-ID to its cache of validated numbers and associates it with the public key from the certificate. Any future calls with this certificate and caller ID from that source will be trusted and not require the verifying callback.

The sequence diagram for this case:





1. Alice’s agent presented a false caller ID, and the agent which owns that false caller ID supports this extension. The verifying INVITE will route through the SIP core but arrive at a different agent, that of c.com. That agent supports the stir-verify option tag. However, when goes to validate the values from the Verify-Call header field, it will fail. In that case, it rejects the INVITE with a 472 response code. This is another new response code, which means the call itself should not proceed, and furthermore, the receiving agent did not recognize the information in the Verify-Call header field as valid. When Bob’s agent receives this, it rejects the incoming INVITE with a 472 as well, informing Alice’s agent that it rejected the call due to an invalid caller ID.

2. Alice’s agent presented a false caller ID, and the agent which owns that false caller ID does not support this extension. When the verifying INVITE arrives at c.com’s agent, it will reject the INVITE as normal with a 420 response code due to the presence of

the unsupported Require option tag. This is routed back to Bob's agent. The receipt of a 420 could signify a malicious caller ID, but could also indicate that there was an intermediate PSTN gateway in the SIP core, in which case the caller ID could be authentic. In this case, Bob's agent MAY complete the call towards the caller.

Each agent builds its own cache of validated certificates for caller ID values. These caches do not need to be shared between providers; they are purely localized to a single administrative entity. The cache entries are invalidated based on the lifetime of the certificate, or through the receipt of an incoming INVITE whose caller ID matches a cache entry, but with a different public key in the certificate. This can happen legitimately due to a number port. In such a case, the receiving agent removes the cache entry and re-performs the validation callback.

Open Issue: Should a new public key invalidate previos ones or should multiple public keys for same caller ID be allowed.

The design proposed here uses an INVITE in the reverse direction, rather than an OPTIONS request or another extension, to maximize the probability that the verifying call actually traverses the SIP core. The significant number of SBCs and other entities which are not likely to pass OPTIONS or non-INVITE requests makes this the best approach for success. It also ensures that the same policy that would be use to route a real call, routes the verifying call.

The presence of the Require header field in the verifying INVITE is critical to the operation of the solution. It prevents the verifying INVITE from actually ringing a real phone, which would be quite annoying.

4. Interactions with RFC 8226

This mechanism provides a technique for deploying STIR prior to the availability of RFC 8226 certificates. It also works nicely in conjunction with incremental deployment of RFC 8226.

In the case where an originating agent supports both this specification and RFC 8226, it would use the RFC 8226 certificates which cryptographically assure its ownership of the number in the From header field. When this is received at the terminating agent, if that agent supports both RFC 8226 and this specification, it first checks for the presence of the RFC 8226 certificate. If present and valid, it proceeds with the call and no verifying callback is required. If the certificate is RFC 8226 compliant but the number

does not match the one in the From header field, or there was no RFC 8226 certificate present, the verifying INVITE is generated.

The consequence of this co-existence is that the volume of verifying callbacks decreases as RFC 8226 is deployed, and the overall system provides verified caller ID the entire time.

5. SS7 Interactions

In reality, significant portions of the PSTN traffic between carriers remain powered by SS7 and not SIP. If that happens, the verifying INVITE might hit an SS7 gateway which is not an agent acting on behalf of Alice.

There are two subcases. In one case, the SS7 gateway does not support this extension. When that happens, the INVITE is rejected with a 420. As described above, Bob's agent will pass the call to Bob. If however the SS7 gateway does support this extension, it still rejects the request with a 420 error code. This is because the overall system - the PSTN - does not support the extension and the call cannot be passed through the PSTN.

TODO: consider specifying an SS7 gateway function and corresponding SS7 extension; this extension needs only a single bit to pass through the SS7 network, and two bits in the call rejection message. It is worth noting that SS7 extensions may be needed to pass the PASSporT information. Need to investigate if that is possible.

6. Formal Protocol Specification

This specification defines behavior for two entities - an originating agent and a terminating agent.

An entity acting as an originating or terminating agent can be a proxy or a B2BUA. However, it MUST be the registrar of record for the user on whose behalf it operates.

6.1. Originating Agent Behavior

6.1.1. On Receipt of incoming INVITE

When an originating agent is acting as an outbound proxy on behalf of the user and receives an outbound INVITE from a user (no Require header field with a value of stir-verify), it MUST include a Supported header field in the INVITE with a value of stir-verify. It MUST add an entry to a table, the pending transactions table.

Furthermore, the originating agent MUST follow the procedures defined in [RFC8224] and [RFC8225] to compute a passport and create a signature over it. It MAY utilize either a self-signed certificate or a traditional domain based certificate.

6.1.2. On Receipt of a Verifying INVITE

When an originating agent receives an INVITE with a Require header field containing the value stir-verify, it MUST examine the INVITE for the presence of a Verify-Call header field. If this header field is not present, the originating agent MUST reject the INVITE with a 400 error code. If the header field is present, the agent extracts the value there, and checks that it represets a valid PASSporT signature using any self singned certificates for the caller ID.

If it is valid, it MUST reject the incoming INVITE with a response code of 471. If it is not valid, it MUST reject the incoming INVITE with a 472 response code.

A response with a 471 response code MUST contain a signature, placed into the Verify-Call header field in the response. This signature is computed by taking the caller ID from the incoming INVITE, concatenating it with the value present in the Verify-Call header field, and then using that as an input to the signature function. TODO: provide detailed spec on signature function.

Open Issue: is this signature in 471 needed?

6.2. Terminating Agent Behavior

6.2.1. On Receipt of Incoming INVITE

When a terminating agent receives an incoming request for a user on whose behalf it operates, it checks for the existence of the Supported header field with a value of stir-verify. If not present, the agent SHOULD pass the call to the targeted user. If present, the agent behaves as follows.

The agent SHOULD maintain a validation cache. This cache is indexed by E.164 number, and contains as a value the public key of the certificate for the agent that was validated as being authoritative for that number.

The agent extracts the number from the From header field of the incoming INVITE. It performs the validation processing defined in [RFC8224] to verify the signature. Once validated, it checks the value of the From header field against the cache.

If there is a matching cache entry, and the public key in the cache entry matches that of the certificate, the agent SHOULD forward the original INVITE towards the called party.

If there is a matching cache entry, but the public key in the cache entry does not match that of the certificate, the agent MUST invalidate the cache entry and proceed as if there was no match.

If there was no matching entry in the cache, the agent constructs a new INVITE header field. The Request-URI and To header field of this INVITE MUST match that of the From header field from the incoming INVITE. The From header field MUST be set to the value from the To header field in the incoming INVITE. The request MUST contain a Require header field with value stir-verify. The request MUST contain any valid SDP offer [RFC3264]. This request MUST then be sent towards the request URI in the same way it would have been sent had it been received from its own user.

The agent sets a timer, with a RECOMMENDED value of 5 seconds. This represents the maximum amount of time the agent will wait for a response to the verifying INVITE before passing the call onwards to the the target of the incoming call.

6.2.2. On Receipt of a Response to the Verifying INVITE

If the terminating agent receives a 471 response to the verifying INVITE, it MUST look for the presence of a Verify-Call header field in the response. If not present, the original INVITE is rejected with a 472, and it MUST NOT add an entry to its validation cache. The signature from this Verify-Call header field is verified, and checked to match against the public key used in the incoming INVITE. If not valid, the original INVITE is rejected with a 472, and it MUST NOT add an entry to its validation cache. If the signature is valid, It SHOULD add an entry to its validation cache. This cache is indexed by the caller ID present in the From header field of the original INVITE. Its value is the public key from the certificate in the incoming INVITE.

If the terminating agent receives a 472 response to the verifying INVITE, it MUST NOT add an entry to its validation cache. It SHOULD reject the original INVITE with a 472 error response. If the terminating agent receives a 420 response to the verifying INVITE, it MUST NOT add an entry to its validation cache. It SHOULD forward the original INVITE towards the called party.

6.2.3. On expiration of the timer

If the 5 second timer fires before a response has been received to the verifying INVITE, the agent SHOULD CANCEL the verifying INVITE. It SHOULD forward the original INVITE towards the called party.

7. Security Considerations

The primary purpose of this specification is to improve the security of caller ID in the public SIP-based phone network. We can consider three actors in the system, and examine malicious behavior from each. These actors are the caller, the callee, and the agent receiving the verifying INVITE.

7.1. Attacks from the Calling Agent

The primary attack the caller can launch is to place a call with a faked caller ID. Preventing this attack is the primary purpose of this specification. This specification prevents it under the assumption that the SIP core network provides forward routability, and therefore, the caller ID is valid if the agent that placed the call, would also receive a call placed towards that callerID. This relationship is verified with the signature over the callerID in both INVITE requests.

It is possible in this system for the calling agent to lie about the callerID, but for the fake caller ID to be associated with the number space owned by that agent. In that case, the calling agent can verify its own faked caller ID. However, since the originating agent is in purview of the usage of its own numbers, there is little that can be done to solve this attack, and in many regards it is not an attack. As an example, outbound call center calls frequently "lie" about the caller ID by placing the company main number in the callerID. Since both are owned by the same administrative entity, this is an acceptable use case.

In a different attack, the calling agent is malicious. It doesn’t lie about its callerID in the outbound INVITE. However, when the verifying call arrives, the calling agent rejects it with a 472, indicating that the caller ID was faked. The only affect of this action would have is to cause the verify call placed by the calling agent to be rejected, and therefore seems to serve no purpose.

An additional consideration is whether the mechanism specified here can be used as a denial of service attack. Consider a malicious originating agent which purposefully inserts a fake caller ID, not to be delivered to the called party, but to trigger a verifying INVITE to the agent which actually owns that phone number. Indeed, based on

this specification, the terminating agent will in fact generate such an INVITE. However, since the attacker must emit a single INVITE in order to cause the terminating agent to generating a single INVITE, there is no amplification possible.

7.2. Attacks from the Called Agent

Consider the case where the called agent is malicious. The calling agent A is not malicious, and places a legitimate call with a valid caller ID (tel:2) to agent B. Agent B places a new call (not a verifying call) to a third agent, agent C, using the same Call-ID as the incoming INVITE it just received, and claims the caller ID tel:2. When agent C places a verifying call for this caller ID, tel:2, it will be routed back to agent A. In this case, because there is in fact a valid call in progress from agent A with that caller ID, the verifying call will succeed. This will cause agent C to believe that agent A legitimately owns the caller ID tel:2, and agent C now caches the certificate from agent B. Agent B is now free, at will to place calls towards agent C with the fake caller ID.

This is prevented through the usage of the signatures in the 471 response codes. In this attack, the signature used by A to sign the response will use its own public certificate. This will not match the one used in the inbound INVITE from B to C which triggered the verifying call. Therefore, B will reject the incoming INVITE and will not update its validation cache.

7.3. Attacks from the agent receiving the Verifying INVITE

In the case where the caller is malicious, and so is the agent receiving the verifying INVITE, it is possible (even without collusion) that the agent receiving the verifying INVITE responds with a 471 to the verifying INVITE, even though it doesn’t actually own the number in question. It might do this in an attempt to pollute the cache of the called agent with an invalid entry.

This is prevented through the usage of signatures in the 471 response. Since the agent receiving the verifying INVITE is not the same as the calling agent, and there is no collusion in which private keys are shared, the signature in the 471 will not match that of the incoming INVITE. This will cause the incoming INVITE to be rejected, and no valid cache entry is added.

8. IANA Considerations

This specification registers a new SIP option code and two new response codes.

8.1. sip-verify Option Tag

This section registers a new SIP option-tag, sip-verify. The required information for this registration, as specified in RFC 3261, is:

Name: sip-verify

Description: This option code indicates support for verification of caller ID using a verifying INVITE. When present in a Supported header field, if informs the recipient that it can, and should, generate a verifying INVITE to confirm the caller ID. When present in a Require header field, it tells the receiving agent that the purpose of the INVITE is to validate that a prior call had been placed, and that the INVITE should not actually be passed to the target of the INVITE.

8.2. Response Code 471

This section registers a new SIP response code, 471. The required information for this registration, as specified in RFC 3261, is:

RFC Number: NOTE TO RFC-EDITOR: replace with the RFC number of this specification.

Response Code Number: 471

Default Reason Phrase: Caller ID Verified

8.3. Response Code 472

This section registers a new SIP response code, 472. The required information for this registration, as specified in RFC 3261, is:

RFC Number: NOTE TO RFC-EDITOR: replace with the RFC number of this specification.

Response Code Number: 472

Default Reason Phrase: Caller ID Not Verified

8.4. Verify-Call Header

TODO

9. Acknowledgments

Thanks for Richard Barnes for identifying the attacks described in the Security Considerations section.

10. References

10.1. 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>.

[RFC3264] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)", RFC 3264, DOI 10.17487/RFC3264, June 2002, <https://www.rfc-editor.org/info/rfc3264>.

[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>.

10.2. Informative References

[RFC2916] Faltstrom, P., "E.164 number and DNS", RFC 2916, DOI 10.17487/RFC2916, September 2000, <https://www.rfc-editor.org/info/rfc2916>.

[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>.

[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>.

Authors' Addresses

Jonathan Rosenberg
Cisco Systems

Email: jdrosen@jdrosen.net

Cullen Jennings
Cisco Systems

Email: fluffy@iii.ca

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 3, 2018

J. Rosenberg
C. Jennings
Cisco Systems
March 2, 2018

SIPCoin: A Cryptocurrency for Preventing RoboCalling on the PSTN
draft-rosenberg-stir-sipcoin-00

Abstract

Robocalling has become an increasing problem in the Public Switched Telephone Network (PSTN). While techniques like verified caller ID can help reduce its impact, ultimately robocalling will continue until economically it is no longer viable. This document proposes a new type of cryptocurrency, called SIPCoin, which is used to create a tax - in the form of computation - that must be paid before placing an inter-domain call on the SIP-based public telephone network. SIPCoin maintains complete anonymity of calls, is non-transferable between users avoiding its usage as an exchangeable currency, causes minimal increase call setup delays, and makes use of traditional certificate authority trust chains to validate proofs of work. SIPCoin is best used in concert with whitelist based techniques to minimize costs on known valid callers.

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 3, 2018.

Copyright Notice

Copyright (c) 2018 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. Problem Statement 3
2. Reference Architecture 4
3. Terminology 5
4. Requirements 5
5. Applicability of Traditional Cryptocurrencies 7
6. Applicability of Challenge Based Solutions 8
7. Overview of SIPCoin 8
7.1. SIPCoin Roles 9
7.2. Creation and Maintenance of the Self Ledger 9
7.3. Transaction Types 11
7.3.1. Create Transaction 11
7.3.2. Burn Transaction 11
7.4. Closing Ledger Pages 12
7.5. Server Validation 13
7.6. Constructing Burn Receipts 14
8. Usage of SIPCoin with SIP 15
9. Deployment Considerations 16
9.1. Enterprise SIP Trunks 16
9.2. Inter-Carrier Trunks 17
9.3. Consumer provider to Mobile Phone 17
9.4. Target Model 18
10. Governance 18
11. Economic Analysis and Parameter Tuning 18
11.1. Cost Targets 18
11.2. Impact of Compute Variability 20
11.3. Load Analysis on the CAs 20
12. Alternative Consensus Techniques 21
13. Security Considerations 21
13.1. Creating Additional SIPCoin 21
13.2. Burning a SIPCoin Multiple Times 22
14. IANA Considerations 23
15. Acknowledgments 23
16. References 23
16.1. Normative References 23
16.2. Informative References 23
Authors' Addresses 23

1. Problem Statement

Robocalling (also known as SPAM, voice SPAM, and so on) has become an increasing problem in the Public Switched Telephone Network (PSTN). Efforts to prevent it - such as the do-not-call list - have so far proven ineffective. Recently, robocallers have gotten even more crafty, and are tailoring the caller ID of incoming calls to match the area codes and exchanges of the recipients in order to increase the likelihood that targets pick up the phone.

This problem is not new, and ultimately the techniques for its prevention have been known for some time. [RFC5039] outlines a number of techniques for prevention of SPAM in Session Initiation Protocol (SIP) [RFC3261] based systems.

Ultimately, SPAM calls are a matter of economics. Each call costs the spammer a certain amount of money to perform. However, a small fraction of calls produce a successful result, generating economic returns. As long as the profit is positive, spammers will continue and will likely work around legal hurdles, blacklists, reputation systems, black lists, and so on. Consequently, the only true way to end robocalling is to use economics - to make it no longer profitable.

This can be achieved in two ways. One is by the exchange of actual monies across all access and peering points in the public telephone network. As the telephone network continues to grow, this becomes increasingly difficult. Furthermore, it only requires a single point of failure at one peering point, and calls have a way to enter the network. Indeed, this is exactly why we see robocalling today despite the fact that monies are in fact exchanged within the PSTN.

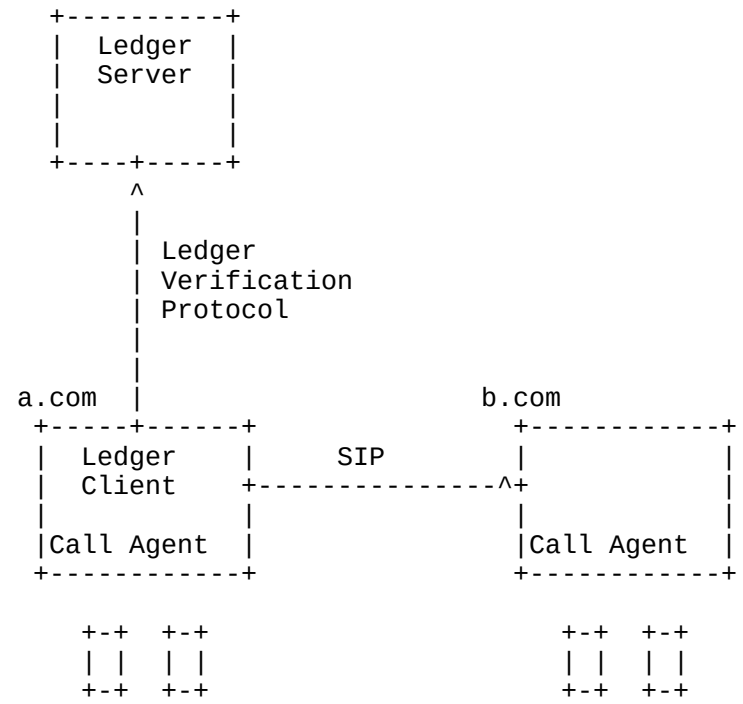
An alternative solution is to use computational puzzles, as described in Section 3.9 of [RFC5039]. The original concept described there is the a callee passes a computation test back to the caller, which performs it, and then passes the results towards the callee. This suffers from two problems. One, described in the document, is that there is high variability in the computation capabilities of individual calling devices and systems. Secondly, performing the computation at call initiation time increases call setup delays. This increase is likely to be large, owing to the amount of computation required to act as an economic disincentive.

Consequently, the problem to be solved is to provide a system that requires callers to demonstrate a proof of work towards callees in a way which does not suffer these problems. Fortunately, in the intervening years since the publication of [RFC5039], blockchain technology was invented, and along with it, a wealth of

cryptocurrencies (BitCoin, Ethereum, etc). The goal is to apply these technologies in a way to solve the unique requirements of the problem at hand.

2. Reference Architecture

The reference architecture for SIPCoin is:



In this architecture, users associated with one call agent (representing a.com) wish to communicate with users associated with a different agent, reachable through b.com, using the Session Initiation Protocol (SIP) [RFC3261]. The b.com agent wishes to gate incoming calls based on proof of computational work provided by the a.com call agent. To perform this, the a.com agent implements the client component of the Ledger Verification Protocol (LVP). In LVP, clients - in this case embedded into the call agent - perform hashing operations, and maintain a self-generated ledger of transactions. To validate pages in the ledger, the ledger client accesses a ledger server through LVP. Through this protocol, the ledger client can obtain information to include int the SIP INVITE. A call agent will typically implement many instances of the ledger client, since each instance has an upper bound on the amount of calls per second it can perform.

In this architecture, there are two call agent roles - the generating agent and the receiving agent. Though, in the picture as shown, they represent the registrar of record for the caller and callee respectively, this need not be the case. Rather, the two roles can be implemented at differing paths along the actual call setup, and indeed occur multiple times along the call. Later sections in this document map the architecture to recommended points of physical implementation.

3. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

Generating Agent
: The SIP proxy, user agent or B2BUA which wishes to demonstrate proof of work in order to pass a call downstream towards a receiving agent which will ultimately validate the proof of work.

Receiving Agent
: The SIP proxy, user agent or B2BUA which will only accept incoming calls under demonstration of proof of work.

4. Requirements

- Unlimited Participants: The system must allow for an unlimited number of call agents to participate. New agents should be able to come and go on demand. This allows the system to extend to agents representing carriers, enterprises, home networks, and so on.

- Low Latency: The system should not significantly increase the call setup delay for calls. This is a big constraint, since it means that proof-of-work computations must be performed in advance of placing the actual call. One to two seconds is acceptable, but not more than that.

- Privacy Protection: There must not be any sharing of logs of calls, personally identifiable information (PII), phone numbers, or similar information. Sharing includes passing this information between entites which would otherwise not have access to it, or storing it in some kind of ledger.

- Non-Transferrable: Any currency used for placing calls must be limited in scope to only allow placing of calls, and not be transferrable amongst participants in the system, or exchangeable for traditional or crypto currencies. This is a significant

requirement since it rules out all existing cryptocurrencies by definition. Why is this requirement important for this use case?

* Enable small players: SIP was designed to enable an open interconnection amongst anyone on the Internet. A SIP domain can be a single device supporting a single user. It can be a home network. It can be a small business. It can be a large enterprise. It can be a small telco, a large telo, or a massive global provider. In order to enable the most open access possible, barriers to entry must be small. Consequently, we want to retain the property of SIP that a two person domain can install an open source SIP server, and be off and able to make calls. Transferability would mean that the currency has real value, and thus to operate a system, the agent must be able to connect to currency exchange systems, payment processing platforms, and so on, in order to obtain the currency before being able to place the first call. This makes it difficult for small players to participate.

* Fraud: The entire purpose of this system is to prevent fraudulent entities from placing calls into the global SIP network. If it was based on transferrable cryptocurrencies, it would likely be susceptible to fraud and thus benefit the very entities we are trying to stop.

* Managed Costs: Today’s cryptocurrencies have highly variable exchange rates, sufficiently variable that they are difficult to use as a payment vehicle, and even more difficult to use for microtransactions. However, that is exactly the opposite of our case - we require high volume, extremely low cost microtransactions, at a price point which hits a particular operating point that is just high enough to make it unprofitable for spammers yet not overly expensive for real callers. Consequently, by tying the cost strictly to the price of computation, we reduce (though certainly do not fully eliminate!) the risks of highly variable currency and allow for relatively low cost microtransactions.

o Non Privileged: The system should not require centralized entities to have access to telecom databases or other information which requires governmental or regulatory access. This constraint in the system makes it incrementally deployable without waiting for the centralized bureaucracy of telco operations. Any centralized capabilities must be an easy incremental add to existing services (e.g., a change to current cerificate authorities).

- o Phone Numbers or SIP URI: The system should not require phone numbers to operate. It should work with traditional domain-based SIP URI as tell as tel URI phone numbers.

- o Predictable Cost: The system must enable a call agent to perform a certain amount of computation and be able to predict the amount of calling which it can perform for a given amount of computation performed in advance of the call. Without this property, an agent runs a risk it cannot service real-time requests for calls from its users because it doesn't have enough crypto currency. This property is related to the non-transferability requirement; if the crypto currencies were transferrable, an agent could instantly purchase crypto currency to place a call. Without transferability, predictable computation is required to ensure the ability to place a call.

- o Managed Governance: Since adjustments will need to be made in the computational costs required, the system must support a managed governance model under the authority of a standards body, such as the IETF or ITU.

5. Applicability of Traditional Cryptocurrencies

One immediate question is - why not just use Bitcoin or one of the other crypto currencies? This would be easy to do. Each SIP INVITE would contain a reference to a transaction that passes the required costs from the caller to the callee.

Putting aside for a moment the non-transferability requirement - which rules out all existing cryptocurrency - other requirements make Bitcoin and similar cryptocurrencies non viable.

Firstly, they fail on the privacy requirement. Usage of Bitcoin would require transactions in the ledger to identify the caller and called parties, thus leaking information about who is calling who.

Secondly, the systems do not provide predictable or managed costs, which are essential for this application. The cost of Bitcoin is highly variable, and subject to (sometimes wild) market swings. These costs cannot be managed by any consensus organization, and indeed the cost may collapse entirely, completely destroying the benefit of the system.

Finally, Bitcoin is too slow. It, and similar cryptocurrencies, rely on ledgers which post infrequently, causing transactions to take minutes or even hours to eventually post and be verified. This system requires a transaction - the spending of a coin to place a

call - to happen fast enough that it can be spent by the caller, and verified by the callee, within one to two seconds.

6. Applicability of Challenge Based Solutions

The second question to ask is - why not just have the callee challenge the caller to perform a computational puzzle at time of call setup, and the caller returns the results?

The primary problem with this class of solution is the time it takes to perform enough computation to serve as an economic disincentive for placing spam calls. To get a general feel for the costs using modern compute, consider Amazon EC2 on demand pricing. For a middle of the road compute optimized node - say - the c4.large instance - as of February 25, 2018, Amazon is charging USD 10 cents per hour (.0027 cents per second) of computation for an instance in US East. We can imagine that our goal for disincentivizing an attacker is somewhere between a .1 cent per call, and perhaps as high as a 10 cents per call, this would require computation on this particular instance type of between 37 seconds (for .1 cent of cost) and 1.01 hours (for one dollar).

Of course, modern Bitcoin mining no longer uses CPUs or even GPUs for that matter, but rather ASICs. Though these can perform far more computation per unit time interval than a CPU for specialized hashing. However, the raw cost per hour of operation - regardless of the amount of computation that can be performed - is the question at hand for analyzing the viability of a challenge/response approach. ASIC and GPU based systems are higher cost per hour to operate due largely to their scarcity. [[OPEN ISSUE: hmm, not sure this argument works owing to asymmetry issues]]

37 seconds - and certainly one hour - is far too long to wait before a call can be forwarded to the called party. For this reason, this class of technique does not work. The solution requires the performance of the computation ahead of the call.

[[TODO: go through all EC2 instance types, price out a more normalized compute cost - dollars per Ghz per hour. Such a metric normalizes against number of CPUs as well as variations in the performance of the CPUs.]]

7. Overview of SIPCoin

This section provides an overview of SIPCoin, a new cryptocurrency used for placing SIP calls over the global SIP network.

SIPCoin differs from Bitcoin significantly in that it does not rely on completely decentralized trust. Rather, it bootstraps itself on the existing certification authorities which power the modern web. As such, the system has two distinct actors - clients, and servers. Clients are entities which perform computation in order to create SIPCoins, and then "burn" those coins in order to place a call. Consequently, SIPCoin supports only two types of transactions - a "create" transaction which creates a Bitcoin through the solution of computational puzzles, and then a "burn" transaction which destroys a coin by binding it to a particular SIP call. Since the create and burn transactions are localized - they affect only the client itself - there is never a need for sharing of the ledger. Consequently, clients actually maintain their own ledgers for these transactions, as described below. A client needs to provide proof that it has burned a token; that proof is performed with a different object - a Burn Receipt - constructed by the client using data returned from the server.

7.1. SIPCoin Roles

Clients are uniquely identified by their public key. There is no need for a certificate to be associated with the public/private key pair. Indeed, typically a single administrative entity - such as a telco operator - would have hundreds or thousands of clients, each with its unique public/private keypair. An administrative entity can create and destroy client instances at will, without any centralized configuration or provisioning.

Servers - typically run by, or co-resident with certificate authorities - are responsible for verification of ledger pages created by clients, and issuing of data needed by clients to construct burn receipts for coins that are verifiably burned on the ledger. The protocol puts the burden of storage of all ledger information entirely in the hands of clients, such that servers require a tiny amount of storage per client. Since servers are run by certificate authorities, their verification of ledger pages and issuance of data to construct burn receipts relies on their private keying material, trusted by all other actors.

7.2. Creation and Maintenance of the Self Ledger

Each client is responsible for maintenance of a ledger of its own create and burn transactions, the only two types of transactions permitted in the system. The ledger is broken into a series of pages. The client posts transactions into the current page of the ledger, called the active page. Each page starts with a page key, which is a hash of the prior page, forming a chain. Following the hash are a series of transactions. The pages prior to the active one

will all - through the LDP protocol - be signed by the server. These pages are called closed pages, and the server's signature over the page forms the final element in a closed page. The client is responsible for storing the prior pages in the ledger persistently.

Clients do not need to maintain prior pages indefinitely. Recall that each page is composed of a series of create and burn transactions. For a particular page, a client can delete a page from storage when all of the following conditions are met:

1. All the prior pages have been deleted

2. All of the create transactions in the page have been burnt in a subsequent page which has been closed

3. All of the Create transactions in the page have a subsequent Create transaction in a page which has been closed

In essence, the client maintains a sliding window of pages, with the tail being the current active page, and the head being the newest page that still contains an unburnt coin or Create transaction that formed the seed of the hash for the current, in-progress one.

The client is required to maintain these pages because they will need to be presented to the server to sign the current page, transitioning it from active to closed.

If a client should lose its pages, it forfeits any coin which it may have created. This is a significant difference compared to traditional Bitcoin, which uses a distributed storage system to provide a global ledger based on consensus, shared by all participants. In SIPCoin, there are many parallel ledgers, and each is stored locally only to that participant. This also means that all partiicpants in SIPCoin can mine coins; it is not a competition. Competitive mining favors the largest and most invested players, preventing others from being able to mine at all, in some cases. Since it is not possible to transfer SIPCoin, such a situation would mean that a SIP entity might not be able to place a call since it never won a lottery.

When a new client is created by an administrative entity, it needs to begin a new ledger. Each ledger and ledger page must be unique, ensureing that the proof of work transactions on one ledger cannot be copied into any other ledger. To create a new ledger, the client transacts with the server to obtain a first page. The first page is signed by the server - like all other pages. However, unlike subsequent pages, it contains no transcations - just a page key. The

server will choose a crypto-random value for the page key, ensuring that no two ledger pages start with the same value.

7.3. Transaction Types

SIPCoin supports only two types of transactions that can be placed into the ledger. These are the create transaction and the burn transaction.

7.3.1. Create Transaction

The create transaction is composed of the following elements:

1. The challenge. This is a number that forms the seed of the hashing. For the first transaction in a page, the challenge is equal to the page key. For all subsequent create transactions, the challenge is a hash of the prior Create transaction in the ledger.

2. The solution. This is a number which demonstrates that the proof of work has been done. Each proof of work is a hash function Ht() which takes as input two numbers, and returns a hashed result. The proof is demonstrated by providing a value S for the solution which, when hashed with the challenge C, forms a result H(S,C) which has N_Zero consecutive zeroes in the result. N_Zero is a global configuration parameter, and is discussed in more detail later on. Its adjustment is a principle part of the governance of the operation of SIPCoin.

3. The Coin ID: This is computed by the client as a hash over its public key, the challenge, and the solution. It serves as a unique identifier for the Coin produced by this create transaction.

7.3.2. Burn Transaction

The Burn transaction is created by the client when it wishes to place a SIP call. Consequently, each burn transaction is bound with a SIP INVITE. To perform this linkage, the burn transaction is composed of the Coin ID (obtained from a prior create transaction for an unspent coin) along with a hash over several fields of the SIP INVITE. The fields incude the From, To, Call-ID and fields from the SDP, such as media encryption keys. The hash also includes the timestamp for the burn transaction.

Beacuse the burn transaction is a hash over these various parameters, when it is sent to the server for signature, the server has no way to invert the hash. Consequently, the server learns nothing about the

originator of the call, the recipient of the call, the type of media in the call, or anything else. All that the server learns is that a call was placed, and that it was placed by the administrative entity that has a relationship with the server. This does mean that servers, through the observation of burn transaction rates, will know the call volume being emitted by the entity, but thats it.

The SIP agent running the client will not be able to send the SIP INVITE until it has received a burn receipt from the server. In essence, it needs to hold the INVITE until the ledger page is complete. For this reason, in SIPCoin, ledger pages close very fast. A client can post a ledger page for closure at a frequency on the order of one every 250ms to 500ms.

7.4. Closing Ledger Pages

A client closes the active ledger page when one of two conditions is met:

1. The ledger page contains N_trans transactions in it

2. The client requires a burn receipt for a burn transaction on the page, and it has not posted a ledger to the server within the last T_min seconds

A client is not required to close a ledger every T_min seconds; if it has no pending burn transactions in the ledger (only creates), it can wait. T_min specifies the minimum interval, and it is nominally enforced on the server to ensure the server is not overloaded.

To actually close the page, the client signs the active page with its public key, and then transmits the active page to the server, along with the public key. The first time it closes a page, it will also need to post all closed pages to the server. The server will validate the transactions in the current page, including insuring that the client has not double burnt the same coin. That particular check requires the server to have all active pages for the client, which is why they must be sent.

Once the server performs its checks, it will send back a signed version of the page, closing it. This enables the client to start a new active page in the ledger. The server also returns a signature over the now-closed page, using its trusted certificate.

The server also returns a signed hash, described below, that allows the client to compute burn receipts for each SIPCoin that was burned.

7.5. Server Validation

The server follows a standardized process for validating the page submitted by the client. At a high level, it composes the following steps:

1. The server authenticates the client; typically this is done using an administrative credential for the administrative entity responsible for the client. [[NOTE: Use ACME techniques for this??]]. LVP technically speaking does not require the server to actually authenticate the client if it chooses not to.

2. The server checks the signature on all pages sent by the client to ensure that they have been signed by itself.

3. The server validates that the pages form a sequential chain. It starts at the first page, computes it hash, and ensures that the result matches the page key of the subsequent page.

4. The server keeps stored, for each unique client (as indexed by public key), the hash of the most recently signed active page from that client, thus closing it. It checks that the active page that is to be signed is the successor, by comparing the page key in the active page to the stored value. This prevents malicious clients from forking the ledger and placing the same burn transaction, but for different INVITEs, into each fork.

5. The server examines every burn transaction in all pages sent by the server, and makes sure it matches exactly one create transaction. This ensures that the server has received all pages from the client (omission of a page from the client would enable it to double burn).

6. The server processes the transactions in order in the active page which is to be signed. If a transaction is ia create transaction, it verifies that the challenge is either the page key (for the first ever Create transaction) or the hash of the prior Create transaction in the ledger otherwise. The server stores, indexed by the public key of the client, the hash of the most recent Create transaction. It verifies this Create transaction has used that value as the challenge. It then takes H(), and uses it with the challenge and solution values. It verifies that the result has N_zero consecutive zeros. It then hashes the client public key with the challenge and solution, and makes sure it matches the Coin ID. If the transaction is a burn transaction, the server takes the CoinID and searches through all burn transactions in all pages sent by the client, and makes sure it doesnt match the Coin ID in any other burn transaction.

Once these validation steps pass, the server generates a signature over the active page using its certificate. It then stores the hash of this closed page to enable it to validate the next one, and stores the hash of the last Create transaction in the page to validate the next Create transaction.

To enable the client to create and send burn receipts, the server computes a balanced binary merkle tree, where the leaf nodes in the tree represent the Burn transactions from the page which was just closed. The head of the merkle tree is the signed by the CA with its private key. The signed head is returned to the client, along with the signed page that was just closed.

For purposes of performance optimization, the server can elect the cache the inactive pages, avoiding the need for the client to resend them each time. To do that, the server stores the pages and generates a cache key, which is an opaque parameter chosen by the server. The client, in subsequent validation requests, can include this key. It can then be used by the server to route those requests to the server instance which is holding the cache, and then used to extract the cached pages indexed by that key. If the server has a cache miss, it can reject the request and force the client to resubmit all its inactive pages.

7.6. Constructing Burn Receipts

To construct burn receipts, the client computes the merkle tree identically to the algorithm used by the server. It then verifes the signature over the head. This will normally be valid, since the CA is trusted in this architecture. The burn receipt for a SIPCoin is a digital object composed of:

1. All of the nodes in the merkle tree, starting at the leaf for the burn transaction for the coin in question, to the head of the tree.

2. For each node in the list above, the sibling of that node.

3. The signature over the head, as provided by the server.

This object is readily verified by having the receiving call agent hash upwards through the merkle tree and compare the result against the signature on the head. This burn receipt is included in the SIP INVITE. The usage of a merkle tree reduces the number of signing operations at the CA and also reduces the amount of data that must be transferred back to the client.

8. Usage of SIPCoin with SIP

The usage of SIPCOin with SIP is relatively straightforward. We say that a "SIPCoin is included in the INVITE" when the INVITE includes a Burn receipt for that coin; in this architecture coins are not actually transfer, only proof of their destruction. SIPCoins can be included in a SIP INVITE proactively with a Burn receipt, or they can be inserted reactively at request of the receiving agent. Its easiest to understand through the reactive flow.

The generating agent sends an INVITE normally, without any SIPCoin in it. This arrives at the receiving agent. Ideally, the receiving agent will verify the caller ID (see [draft-rosenberg-stir-callback] for a solution to enable this to occur). Once verified, the receiving agent checks whether the caller is known to be acceptable to the called party. The definition of acceptable is a matter of local policy and depends on the physical entities performing the receiving agent role, as discussed below.

If the caller is acceptable, the call is passed to the called party. If the nature of the caller is unknown (which is again a matter of local policy), the receiving agent rejects the INVITE with a response code 4xx which challenges for SIPCoin in order to accept the call.

When this is received at the generating agent, it constructs a new INVITE, burns a coin, constructs the burn receipt, and places those into the INVITE. This passes to the same receiving agent. If the caller ID is verified (whcih would have been done from the prior step) and it continues to be unknown, the receiving agent validates the burn receipt.

To validate it, the receiving agent performs the hashing through the merkle tree and verifies the signature on the hash at the top. The certificate verification requires the generating and calling agents to share a common trust anchor. This specification mandates that all agents trust the same set of CAs present in the Mozilla Firefox browser. This allows SIPCoin to be rooted in a well vetted, continuously maintained set of trust anchors which is proven to work globally.

If the signature is valid, the receiving agent considers the burnt coin as a sufficient proof of work to allow the call to proceed to the called party.

In the proactive model, which can be used by the caller to speed up call setup if they desire, they burn a SIPCoin prior to the challenge and include it in the INVITE straight away.

9. Deployment Considerations

There are many ways in which SIPCoin can be used. And in fact, the hardest part of rolling out a solution like SIPCoin is handling the intermediate states where it is only partially deployed on the Internet. This document proposes a phased rollout where each step is motivated by economic benefit to the parties at hand.

9.1. Enterprise SIP Trunks

The easiest deployment topology, and the best way to start, is on SIP trunks between a customer and their provider. In this model, the generating agent is that of the administrative entity which is using the SIP trunk, and the receiving agent is that of the provider. These are adjacent agents connected by a single SIP hop. As an example, the generating agent could be an enterprise, and the receiving agent would be a traditional telco offering enterprise SIP trunks. This would also be combined with the reverse role, where the service provider also runs a generating agent and the enterprise runs a receiving agent.

This arrangement provides a value proposition for the enterprise to protect itself from inbound spam calls which are received through their SIP trunk provider. If the spammer is another enterprise customer of the same provider, that enterprise becomes disincented from spamming due to costs. If the spammer is farther away - and in this phase they are most likely to be - the SP eats the cost and genreates the SIPCoin.

In such a service model, the service provider would - through its bilateral relationships with its customers, insist its customers implement the Outbound SIP Trunk role. As a result, the service provider itself would not need to generate SIPCoin for intra-provider calls. However, it would genreate them for inter-provider calls. This provides a benefit to the enterprise, who are now protected from spammers connected to the same SP, and the fact that the SP creates and burns calls for transit calls means that the enterprise gets the benefit of only ever accepting inbound calls which have SIPCoins burned.

In this model, the SP can save itself money in one of two ways. Firstly is through whitelisting. As part of the SIP trunk specification, enterprises on the receiving side should maintain a database of callers they ’trust’. A caller ID is trusted if the caller ID has been verified [draft-rosenberg-stir-callback], and the enterprise had previously, in the last few weeks, placed multiple calls to that number, those calls having connected and had a duration of at least a few minutes. This provides a simple model of: I’ll

trust your inbound call if I've called you previously. The enterprise PBX can also use contact lists from employees contianing phone numbers to populate this list.

This means the SP cost is reduced for trusted callers, and not for others. To further reduce costs, the SPs are incented therefore to establish bilateral peering with each other over Inter-carrier trunks.

9.2. Inter-Carrier Trunks

These work identically to the enterprise SIP trunks; the carriers on each side of an inter-carrier peering link implement both the generating and terminating roles of the call agents. When a terminating enterprise challenges its SP for a coin, if the call arrived via an inbound trunk from another carrier, the SP can propagate the request for a coin upstream to save itself costs. If the upstream provider doesnt support SIPCoin, the SP must burn the coin itself, creating costs, and thus incentive for each side to insist on implementation to reduce costs.

In this way, SIPCoin implementations propagate outwrads, ultimately reaching the originating carriers for consumer services and enterprises. This brings us to the final phases.

9.3. Consumer provider to Mobile Phone

This specification recommends that the terminating role be implemented in smartphones implementing the IMS specifications. Consider now an enterprise which placed a call towards a consumer mobile phone. This call is received at the terminating mobile provider. Since it knows that the mobile callee SIP UA supports SIPCoin (from the SUpported header field in the REGISTER), it propagates the INVITE towards the called phone after verifying the caller ID. The callee, seeing that the caller ID is verified, checks its local contact list. If the caller is on the contact list, it doesnt challenge for coin. If it isnt, it challenges for the coin. This propagates all the way back to the originating enterprise, which burns a coin to place the call, which is then accepted by the callee.

The generting role is not appropriate for implementation on mobile phones, and as such the consumer mobile operator cannot pass its costs upstream. However, as part of bilateral peering arrangements and standards coordination, the SP can insist that each other require their mobile phones to comply with the specs that mandate implementation of the terminating role. That will save each other money in proportion to the balance of their inbound to outbound calls.

This then provides the final economic incentive to achieve the target architectural model.

9.4. Target Model

In the idealized model, the terminating role is implemented by the receiving phones, and the generating role implemented by the call agents operating on their behalf. The entire SIP core network supports these roles, but as this target deployment architecture is reached, they never need to generate or verify SIPCoin since it is fully handled e2e. This minimize cost for all parties and concentrates it on the entites generating calls to numbers which are never called back, and not on the contact lists of mobile phones.

10. Governance

In order for SIPCoin to be an effective tool against spammers, it requires ongoing governance. This governance takes three forms:

1. Updating of this specification

2. Periodic adjustment of the value of N_Zero

The first of these is fairly routine for the IETF, but new for cryptocurrencies, which rely on distribued consensus amongst majority implementations. SIPCoin is more managed than those networks, and as such we propose the IETF, in essence, manage the behavior of the system through the published RFC.

The second of these is more interesting. In order to deal with changes in the cost of computation over time, it is necessary to adjust the value of N_Zero periodically. This specification suggests that the IETF consensus process be used for this purpose. To speed up implementation, the value of N_Zero must be loaded dynamically by all clients and servers from an IETF maintained and verified website. This allows IETF governance to decide on a new value, and for that new value to be used instantly across the entirety of the SIP based telephone network.

11. Economic Analysis and Parameter Tuning

11.1. Cost Targets

The goal of SIPCoin is to incur cost to callers, in such a way that it erodes the profitability of the spammers to the point of making it no longer viable, and, at the same time, representing only a small increase in the cost to legitimate callers. This represents an operating window in which the system needs to operate.

Let us first consider the tolerable costs to legitimate callers. In most cases we anticipate the costs to be borne by the service providers, and then passed on to consumers or perhaps absorbed if the costs do not merit it. Its important to point out that the cost of SIPCoin is metered per call regardless of destination or duration of call. This tends to penalize entities that make many short calls (as telemarketers do) while benefit those who make fewer, long, international calls (which is more typical of users paying high costs today to call friends and family abroad).

As a back of the envelope analysis - the average phone bill in the U.S. is approximately $100 for a mobile phone each month. According to [PR Newswire][<https://www.prnewswire.com/news-releases/no-time-to-talk-americans-sendingreceiving-five-times-as-many-texts-compared-to-phone-calls-each-day-according-to-new-report-300056023.html>], the average American makes or answers six phone calls per day. Assuming this is symmetric, thats 3 placed calls per day, 90 per month. With a three percent increase in their bill as an upper bound, this means $3 per month, or 3 cents per call.

On the other side of the house - the spammers. Its hard to get precise data - but here is a back of the envelope. A recent [Boston Globe article][<https://www.bostonglobe.com/ideas/2017/05/11/the-onslaught-spam-calls-will-keep-getting-worse/2w1tyrSnzEj8NPO81hUUBK/story.html>] cites that in the US, 2.5B robocalls were placed in the US in April of 2017. Later in the article, it quotes a cost to Americans of $350 million between 2011 and 2013. If we assume this translates directly to the profits of the spammers, over that 36 month period thats $9.7M profit per month. If it took 2.5B robocalls per month to achieve that profit, that is a profit of 0.38 cents per call.

This means there is - on the surface - a viable operating point here. Assuming a 50% erosion in profit is enough to make a dent in telemarketing, our lower bound on the cost of SIPCoin is 0.19 cents per call, and our upper bound is 3 cents per call. This represents an order of magnitude spread. That is without consideration to the addition of whitelists.

When combined with the whitelist and verified caller ID, we can signicantly shift the cost to the spammers. As a back of the envelope, costs are incurred to non-spammers when a user makes a call to a number that the user has never received a call from nor is on the contact list of the callee. There are real use cases for this - a call to a contact center is one such case. Another is a call to a new contact number learned via business card or personal introduction. These are, relativey few. If we assume that, of the 100 or so calls made each month perhaps one is like that, this adds

another two order of magnitude to the spread, resulting in a three order of magnitude improvement. This means that, as long as we can keep the economics of calling such that it is not three times cheaper for a spammer than an SP to mine SIPCoin, the system can be effective.

11.2. Impact of Compute Variability

The hardest challenge in building a system that operates in the cost targets is dealing with the highly variable costs of computation. To give some perspective on this, a somewhat dated article on Bitcoin compute costs [<https://en.bitcoin.it/wiki/Non-specialized_hardware_comparison>] shows a spread of three orders of magnitude in hashing performance across a range of Intel CPUs (from 0.245 Mhash/s (million hashes per second), up to 140 million). It cites the performance of GPUs as sitting in a range from 1 MHash/s up to 2568 MHash/s, and quotes ASICs as being able to reach 1000 GHash/s (Billion hashes per second). The performance spread is therefore seven orders of magnitude. Though there is surely a spread in cost as well, it is assuredly not as large. This means that in SIPCoin, the spammers will be incentivized to buy high performance compute which is viable economically only at high scale.

However, considering the deployment architecture described above, the generating role is implemented by enterprises that have SIP trunks to their carriers, and the carriers. The low end computational devices - mobile phones - actually delegate their generating role to the call agent acting on their behalf. If we imagine that small home networks and small businesses would similarly delegate their generating role to their service provider, we end up in a model where trust relationships primarily put the burden of computation on larger entities, which can in general afford to just all use ASICs, which can eliminate the disparity between the spammers and the good guys.

In other words, if the spammer can afford some ASIC-based machines, Verizon can too.

11.3. Load Analysis on the CAs

This proposal introduces a new role to be played by a CA, in the verification of SIPCoin ledgers. This process is, fortunately, almost stateless, requiring a query for just two hash value indexed by a public key. There are no user records, payment systems, cryptographic storage (beyond what they already implement). However, it is extremely high volume.

Assume a large carrier is about 100 million subscribers. Assume that they do an average of about 10 calls attempts per day per user.

Assume volume at peak is 3x average (ignoring things like earthquakes in California). For calculation purposes, lets say we we are closing ledger ever 0.5 seconds. That gives us (100,000,000 * 10 * 3 / 246060 / 0.5) = 70,000 entries per close in busy case. Lets say our EC signature are 100 bytes and that a burn or create transaction fit in 256 bytes total and that a given page has about equal number of create and burn. This gives me that the CA, even it it only goes back a 2 pages, needs to look at 3 pages * 70,000 entries * 2 (for create and burn) * 256 bytes = 100 Mbyte each half second or about 1.6 Gbps.

Is this too much? Its a lot. But not out of the realm of reasonableness.

12. Alternative Consensus Techniques

The proposal here uses the CAs as trusted third parties to verify the ledger. This is owing to the challenges in achieving rapid consensus in large scale distributed blockchains. However, a variant on the proposal here is to elect randomly a small subset of the entities participating in bitcoin and require consensus only amongst a subset. The size of the subset needs to only be larger than twice the number of malicious entities we wish to tolerate. One can argue that the incentives for being malicious in SIPCoin are smaller (just spammers), perhaps they only represent 5% of call agents in the network (whcih would be a lot!). So we only need 10% of the nodes for consensus.

If the set of elected nodes can be small, and they are very well connected to each other, we can run full-mesh consensus protocols which are potenitally fast enough to achieve consensus and sign results and then distribute them at a speed which meets the requirements here. These elected agents would exactly implement the server side role of LVP, and validation is by looking at consensus view rather than verifying signatures.

13. Security Considerations

There are many attacks possible in this system. THe primary ones to prevent are the clients acting maliciously in order to either create additional SIPCoin without doing the hashing work, or use the same SIPCoin for multiple SIP INVITEs. We consider both forms of attack.

13.1. Creating Additional SIPCoin

A client might maliciously obtain a SIPCoin from another client in some way (perhaps eavesdropping or theft of databaase), and then use it for itself. However, it cannot do that. Since the challenge in

the SIPCoin is bound to the ledger in which it lies, by using the page key, and then the page key is linked to the entire ledger chain for the same client, it is not possible to insert SIPCoins into different ledgers.

A client might try and perform the hashing and then insert the same SIPCoin twice into the same ledger page. However, this is not possible because the server will confirm each Create transaction derives from a unique predecssor. In a similar way, a client might try to insert the same create transaction into two different ledgers. Since the server maintains an index of the most recent Create transaction, it would detect this.

13.2. Burning a SIPCoin Multiple Times

One way in which a client might try and burn the same coin twice is to literally have the same burn transaction reference the same coin in its sequential ledger chain. This is prevented through the core validation steps performed by server, which looks for such duplicates.

Another way in which a client might try and burn the same coin twice is to fork the ledger, and put the same Burn event in different pages. This is prevented because the server will verify and then sign the first such forked page presented to it. When it does, the server basically advances the pointer it maintains to the most recently closed page in the ledger. When the client tries to fool the server into verifying the second fork, the server will reject it because the currently active page is not the direct descendant of the previously closed page. Thus, the client can only maintain a single, sequential ledger.

THe client might try and use the same Burn Receipt in two different SIP transactions. This is not possible, because the Burn receipt includes a hash over the fields in the INVITE which cannot be duplicated by the call agent without for differnt calls - the called party and timestamp. Narrow timestamp windows (say, 2 seconds), prevent even calls to the same number with the same Call-ID within that window.

A client might try and take burn receipts from INVITEs it reuses, and replay them in different INVITEs. The binding of the burn receipt to the called user prevents this.

[[TODO: lot more rigor needed here]]

14. IANA Considerations

TODO

15. Acknowledgments

Many thanks to Ram Jagadeesan and Richard Barnes for their input.

16. References

16.1. 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>.

16.2. Informative References

[draft-rosenberg-stir-callback] Rosenberg, J. and C. Jennings, "Bootstrapping STIR Deployments with Self-Signed Certs and Callbacks", March 2018, <https://tools.ietf.org/html/draft-rosenberg-stir-callback-00>.

[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>.

[RFC5039] Rosenberg, J. and C. Jennings, "The Session Initiation Protocol (SIP) and Spam", RFC 5039, DOI 10.17487/RFC5039, January 2008, <https://www.rfc-editor.org/info/rfc5039>.

Authors' Addresses

Jonathan Rosenberg
Cisco Systems

Email: jdrosen@jdrosen.net

Cullen Jennings
Cisco Systems

Email: fluffy@iii.ca