STIR
Internet-Draft
Intended status: Standards Track
Expires: January 9, 2017

C. Wendt
Comcast
J. Peterson
Neustar Inc.

July 8, 2016

Persona Assertion Token
draft-ietf-stir-passport-04

Abstract

This document defines a token format for verifying with non-repudiation the sender of and authorization to send information related to the originator of personal communications. A cryptographic signature is defined to protect the integrity of the information used to identify the originator of a personal communications session (e.g. the telephone number or URI) and verify the accuracy of this information at the destination. The cryptographic signature is defined with the intention that it can confidently verify the originating persona even when the signature is sent to the destination party over an unsecure channel. The Persona Assertion Token (PASSporT) is particularly useful for many personal communications applications over IP networks and other multi-hop interconnection scenarios where the originating and destination parties may not have a direct trusted relationship.

Status of This Memo

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

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

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

This Internet-Draft will expire on January 9, 2017.

Table of Contents

1. Introduction

In today's IP-enabled telecommunications world, there is a growing concern about the ability to trust incoming invitations for communications sessions, including video, voice and messaging. [RFC7340] As an example, modern telephone networks provide the ability to spoof the calling party telephone number for many legitimate purposes including providing network features and services on the behalf of a legitimate telephone number. However, as we have seen, bad actors have taken advantage of this ability for illegitimate and fraudulent purposes meant to trick telephone users to believe they are someone they are not. This problem can be extended to many emerging forms of personal communications.

This document defines a common method for creating and validating a token that cryptographically verifies an originating identity, or more generally a URI or application specific identity string representing the originator of personal communications. Through extended profiles other information relevant to the personal communications can also be attached to the token. The primary goal of PASSporT is to provide a common framework for signing persona related information in an extensible way. A secondary goal is to provide this functionality independent of any specific personal communications signaling call logic, so that creation and verification of persona information can be implemented in a flexible way and can be used in many personal communications applications including end-to-end applications that require different signaling protocols. It is anticipated that signaling protocol specific guidance will be provided in other related documents and specifications to specify how to use and transport PASSporT tokens, however this is intentionally out of scope for this document.

Note: As of the authoring of this document, [I-D.ietf-stir-rfc4474bis] provides details of how to use PASSporT within SIP signaling for the signing and verification of telephone numbers.

2. Token Overview

Tokens are a convenient way of encapsulating information with associated digital signatures. They are used in many applications that require authentication, authorization, encryption, non-repudiation and other use cases. JSON Web Token (JWT) [RFC7519] and JSON Web Signature (JWS) [RFC7515] are designed to provide a compact form for many of these purposes and define a specific method and syntax for signing a specific set of information or "claims" within the token and therefore providing an extensible set of claims. Additionally, JWS provides extensible mechanisms for specifying the method and cryptographic algorithms used for the associated digital signatures.

3. PASSporT Definition

The PASSporT is constructed based on JWT [RFC7519] and JWS [RFC7515] specifications. JWS defines the use of JSON data structures in a specified canonical format for signing data corresponding to JOSE header, JWS Payload, and JWS Signature. JWT defines specific set of claims that are represented by specified key value pairs which can be extended with custom keys for specific applications.

3.1. PASSporT Header

The JWS token header is a JOSE header [RFC7515] that defines the type and encryption algorithm used in the token.

An example of the header for the case of an ECDSA P-256 digital signature would be the following,

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

3.1.1. "typ" (Type) Header Parameter

JWS defines the "typ" (Type) Header Parameter to declare the media type of the JWS.

For PASSporT Token the "typ" header MUST minimally include and begin with "passport". This represents that the encoded token is a JWT of type passport.

3.1.2. "alg" (Algorithm) Header Parameter

For PASSporT, the "alg" should be defined as follows, for the creation and verification of PASSporT tokens and their digital signatures ES256 MUST be implemented.

Note that JWA [RFC7518] defines other algorithms that may be utilized or updated in the future depending on cryptographic strength requirements guided by current security best practice.

3.1.3. "x5u" (X.509 URL) Header Parameter

As defined in JWS, the "x5u" header parameter is used to provide a URI [RFC3986] referring to the resource for the X.509 public key certificate or certificate chain [RFC5280] corresponding to the key used to digitally sign the JWS. Note: The definition of what the URI represents in terms of the actor serving the X.509 public key is out of scope of this document. However, generally this would correspond to an HTTPS or DNSSEC resource with the guidance that it MUST be a TLS protected, per JWS spec.

3.2. PASSporT Payload

The token payload claims should consist of the information which needs to be verified at the destination party. This claim should correspond to a JWT claim [RFC7519] and be encoded as defined by the JWS Payload [RFC7515]

The PASSporT defines the use of a number of standard JWT defined headers as well as two new custom headers corresponding to the two parties associated with personal communications, the originator and terminator. These headers or key value pairs are detailed below.

3.2.1. JWT defined claims

3.2.1.1. "iat" - Issued at claim

The JSON claim MUST include the "iat" [RFC7519] defined claim issued at. As defined this should be set to a date cooresponding to the origination of the personal communications. The time value should be of the format defined in [RFC7519] Section 2 NumericDate. This is included for securing the token against replay and cut and paste attacks, as explained further in the security considerations in section 7.

3.2.2. PASSporT specific claims

3.2.2.1. Originating and Destination Identity Claims

Baseline PASSporT defines claims that convey the identity of the origination and destination of personal communications. There are two claims that are required for PASSporT, the "orig" and "dest" claims. Both "orig" and "dest" should have values that are JSON objects that include identities represented by key value pairs, where the key represents an identity type and the value is the identity string. Currently, these identities can be represented as either telephone numbers or Uniform Resource Indicators (URIs). The definition of how telephone numbers or URIs and examples are provided below.

The "orig" JSON object MUST only have one key value pair representing the asserted identity of any type (currently either "tn" or "uri") of the originator of the personal communications signaling.

The "dest" JSON object MUST at least have one key value pair, but could have an arbitrary number of destination identities of any type.

3.2.2.1.1. "tn" - Telephone Number identity

If the originating or destination identity is a telephone number, the key representing the identity should be "tn".

Telephone Number strings for "tn" MUST be canonicalized according to the procedures specified in [I-D.ietf-stir-rfc4474bis] Section 7.2.

3.2.2.1.2. "uri" - URI identity

If any of the originating or destination identities is of the form URI, as defined in [RFC3986], the key representing the identity should be "uri" URI form of the identity.

3.2.2.1.3. Future identity forms

We recognize that in the future there may be other standard mechanisms for representing identities. The "orig" and "dest" JSON objects with "tn" and "uri" allow for other identity types with unique keys to represent these forms.

3.2.2.1.4. Examples

Single Originator to Single Destination example:

```
          {
          "iat":"1443208345",
          "orig":{"tn":"12155551212"},
          "dest":{"uri":"sip:alice@example.com"}
      }
```

Single Originator to Multiple Destination Identities example:

```
          {
          "iat":"1443208345",
          "orig":{"tn":"12155551212"},
          "dest":{
                         "uri":"sip:alice@example.com",
                         "tn":"12125551212",
                         "uri":"sip:bob@example.net"
          }
      }
```

3.2.2.2. "mky" - Media Key claim

Some protocols that use PASSporT convey hashes for media security keys within their signaling in order to bind those keys to the identities established in the signaling layers. One example would be the DTLS-SRTP key fingerprints carried in SDP via the "a=fingerprint" attribute; multiple instances of that fingerprint may appear in a single SDP body corresponding to difference media streams offered. The "mky" value of PASSporT contains a hexadecimal key presentation of any hash(es) necessary to establish media security via DTLS-SRTP. This mky value should be formated in a JSON form including the 'alg' and 'dig' keys with the corresponding algorithm and hexadecimal values. Note that per guidance of Section 5 of this document any whitespace and line feeds must be removed. If there is multiple fingerprint values, more than one, the fingerprint values should be constructed as a JSON array denoted by bracket characters. For the 'dig' key value, the hash value should be the hexadecimal value without any colons, in order to provide a more efficient, compact form to be encoded in PASSporT token claim.

An example claim with "mky" claim is as follows:

For an SDP offer that includes the following fingerprint values,

```
    a=fingerprint:sha-256 02:1A:CC:54:27:AB:EB:9C:53:3F:3E:4B:65:
    2E:7D:46:3F:54:42:CD:54:F1:7A:03:A2:7D:F9:B0:7F:46:19:B2
    a=fingerprint:sha-256 4A:AD:B9:B1:3F:82:18:3B:54:02:12:DF:3E:
    5D:49:6B:19:E5:7C:AB:3E:4B:65:2E:7D:46:3F:54:42:CD:54:F1
```

the PASSporT Payload object would be:

```
          {
          "iat":"1443208345",
          "orig":{"tn":"12155551212"},
                  "dest":{"uri":"sip:alice@example.com"},
                  "mky":[
          {
             "alg":"sha-256",
             "dig":"021ACC5427ABEB9C533F3E4B652E7D463F5442CD54
                  F17A03A27DF9B07F4619B2"
          },
          {
             "alg":"sha-256",
             "dig":"4AADB9B13F82183B540212DF3E5D496B19E57C
                  AB3E4B652E7D463F5442CD54F1"
          }
        ]
      }
```

3.3. PASSporT Signature

The signature of the PASSporT is created as specified by JWS using the private key corresponding to the X.509 public key certificate referenced by the "x5u" header parameter.

4. Extending PASSporT

PASSporT represents the bare minimum set of claims needed to assert the originating identity and support the secure propoerties discussed in various parts of this document, however there will certainly be both new uses and ways of extending the application and usage of PASSPorT that requires the ability to extend the defined base set of claims to represent other information requiring assertion or validation beyond the identity itself.

4.1. "ppt" (PASSporT) header parameter

For the extension of the base set of claims defined in this document, a new JWS header parameter "ppt" MUST be used with a string that uniquely identifies and points to a profile specification that defines any new claims that would extend the base set of claims of PASSporT.

An example header with an extended PASSporT profile of "foo" is as follows:

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

4.2. Extended PASSporT Claims

Future specifications that define such extensions to the PASSporT mechanism MUST explicitly designate what claims they include beyond the base set of claims from this document, the order in which they will appear, and any further information necessary to implement the extension. All extensions MUST incorporate the baseline JWT elements specified in Section 3; claims may only be appended to the claims object specified; they can never be subtracted or re-ordered. Specifying new claims follows the baseline JWT procedures ([RFC7519] Section 10.1). Note that understanding an extension as a verifier is always optional for compliance with this specification (though future specifications or profiles for deployment environments may make other "ppt" values mandatory). The creator of a PASSporT object cannot assume that verifiers will understand any given extension. Verifiers that do support an extension may then trigger appropriate application-level behavior in the presence of an extension; authors of extensions should provide appropriate extension-specific guidance to application developers on this point.

5. Deterministic JSON Serialization

In order to provide a deterministic representation of the PASSporT Header and Claims, particularly if PASSporT is used across multiple signaling environments, the JSON header object and JSON Claim object MUST be computed as follows.

The JSON object MUST follow the rules for the construction of the thumbprint of a JSON Web Key (JWK) as defined in [RFC7638] Section 3. Each JSON object MUST contain no whitespace or line breaks before or after any syntactic elements and with the required members ordered lexicographically by the Unicode [UNICODE] code points of the member names.

In addition, the JSON header and claim members MUST follow the lexicographical ordering and character and string rules defined in [RFC7638] Section 3.3.

5.1. Example PASSport deterministic JSON form

For the example PASSporT Payload shown in Section 3.2.2.2, the following is the deterministic JSON object form.

{"iat": 1443208345,"orig":{"tn":"12155551212"},"dest":
{"uri":"sip:alice@example.com"},"mky":[{"alg":"sha-256","dig":
"021ACC5427ABEB9C533F3E4B652E7D463F5442CD54F17A03A27DF9B07F4619
B2"},{"alg":"sha-256","dig":"4AADB9B13F82183B540212DF3E5D496B19
E57CAB3E4B652E7D463F5442CD54F1"}]}

6. Human Readability

JWT [RFC7519] and JWS [RFC7515] are defined to use Base64 and/or UTF8 encoding to the Header, Payload, and Signature sections. However, many personal communications protocols, such as SIP and XMPP, use a "human readable" format to allow for ease of use and ease of operational debugging and monitoring. As such, specifications using PASSporT may provide guidance on whether Base64 encoding or plain text will be used for the construction of the PASSporT Header and Claim sections.

7. Security Considerations

7.1. Avoidance of replay and cut and paste attacks

There are a number of security considerations for use of the token for avoidance of replay and cut and paste attacks. PASSporT tokens must be sent along with other application level protocol information (e.g. for SIP an INVITE as defined in [RFC3261]). There should be a link between various information provided in the token and information provided by the application level protocol information.

These would include:

- "iat" claim should closely correspond to a date/time the message was originated. It should also be within a relative delta time that is reasonable for clock drift and transmission time characteristics associated with the application using the PASSporT token.

- "dest" claim is included to prevent the ability to use a previously originated message to send to another destination party

7.2. Solution Considerations

It should be recognized that the use of this token should not, in it's own right, be considered a full solution for absolute non-repudiation of the persona being asserted. This only provides non-repudiation of the signer of PASSporT. If the signer and the persona are not one in the same, which can and often will be the case in telecommunications networks today, protecting the destination party from being spoofed may take some interpretation or additional verification of the link between the PASSporT signature and the persona being asserted.

In addition, the telecommunications systems and specifications that use PASSporT should in practice provide mechanisms for:

- Managing X.509 certificates and X.509 certificate chains to an authorized trust anchor that can be a trusted entity to all participants in the telecommunications network

- Accounting for entities that may route calls from other peer or interconnected telecommunications networks that are not part of the "trusted" communications network or may not be following the usage of PASSporT or the profile of PASSporT appropriate to that network

- Following best practices around management and security of X.509 certificates

7.3. Privacy Considerations

Because PASSporT explicity includes claims of identitifiers of parties involved in communications, times, and potentially other call detail, care should be taken outside of traditional protected or private telephony communications paths where there may be concerns about exposing information to either unintended or illegitimately intented actors. These identifiers are often exposed through many communications signaling protocols as of today, but appropriate precautions should be taken.

8. IANA Considerations

8.1. Media Type Registration

8.1.1. Media Type Registry Contents Additions Requested

This section registers the "application/passport" media type [RFC2046] in the "Media Types" registry in the manner described in

[RFC6838], which can be used to indicate that the content is a PASSporT defined JWT and JWS.

- Type name: application
- Subtype name: passport
- Required parameters: n/a
- Optional parameters: n/a
- Encoding considerations: 8bit; application/passport values are encoded as a series of base64url-encoded values (some of which may be the empty string), each separated from the next by a single period ('.') character.
- Security considerations: See the Security Considerations section of RFC 7515.
- Interoperability considerations: n/a
- Published specification: draft-ietf-stir-passport-00
- Applications that use this media type: STIR and other applications that require identity related assertion
- Fragment identifier considerations: n/a
- Additional information:
  - Magic number(s): n/a
  - File extension(s): n/a
  - Macintosh file type code(s): n/a
- Person and email address to contact for further information: Chris Wendt, chris-ietf@chriswendt.net
- Intended usage: COMMON
- Restrictions on usage: none
- Author: Chris Wendt, chris-ietf@chriswendt.net
- Change Controller: IESG
- Provisional registration? No

8.2. JSON Web Token Claims Registration

8.2.1. Registry Contents Additions Requested

- Claim Name: "orig"
- Claim Description: Originating Identity String
- Change Controller: IESG
- Specification Document(s): Section 3.2 of draft-ietf-stir-passport-00
- Claim Name: "dest"
- Claim Description: Destination Identity String
- Change Controller: IESG
- Specification Document(s): Section 3.2 of draft-ietf-stir-passport-00
- Claim Name: "mky"
- Claim Description: Media Key Fingerprint String
- Change Controller: IESG
- Specification Document(s): Section 3.2 of draft-ietf-stir-passport-00

9. Acknowledgements

Particular thanks to members of the ATIS and SIP Forum NNI Task Group including Jim McEchern, Martin Dolly, Richard Shockey, John Barnhill, Christer Holmberg, Victor Pascual Avila, Mary Barnes, and Eric Burger for their review, ideas, and contributions. Also thanks to Henning Schulzrinne, Russ Housley, Alan Johnston, and Richard Barnes for valuable feedback on the technical and security aspects of the document. Additional thanks to Harsha Bellur for assistance in coding the example tokens.

10. References

10.1. Normative References

[I-D.ietf-stir-rfc4474bis]
Peterson, J., Jennings, C., Rescorla, E., and C. Wendt, "Authenticated Identity Management in the Session Initiation Protocol (SIP)", draft-ietf-stir-rfc4474bis-10 (work in progress), July 2016.

[RFC2046] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types", RFC 2046, DOI 10.17487/RFC2046, November 1996, <http://www.rfc-editor.org/info/rfc2046>.

[RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, DOI 10.17487/RFC3986, January 2005, <http://www.rfc-editor.org/info/rfc3986>.

[RFC6838] Freed, N., Klensin, J., and T. Hansen, "Media Type Specifications and Registration Procedures", BCP 13, RFC 6838, DOI 10.17487/RFC6838, January 2013, <http://www.rfc-editor.org/info/rfc6838>.

[RFC7515] Jones, M., Bradley, J., and N. Sakimura, "JSON Web Signature (JWS)", RFC 7515, DOI 10.17487/RFC7515, May 2015, <http://www.rfc-editor.org/info/rfc7515>.

[RFC7518] Jones, M., "JSON Web Algorithms (JWA)", RFC 7518, DOI 10.17487/RFC7518, May 2015, <http://www.rfc-editor.org/info/rfc7518>.

[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>.

[RFC7638] Jones, M. and N. Sakimura, "JSON Web Key (JWK) Thumbprint", RFC 7638, DOI 10.17487/RFC7638, September 2015, <http://www.rfc-editor.org/info/rfc7638>.

[UNICODE] "The Unicode Consortium, "The Unicode Standard"", <http://www.unicode.org/versions/latest/>.

10.2. Informative References

[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>.

[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, <http://www.rfc-editor.org/info/rfc5280>.

[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>.

Appendix A. Example PASSporT JWS Serialization and Signature

For PASSporT, there will always be a JWS with the following members:

- "protected", with the value BASE64URL(UTF8(JWS Protected Header))
- "payload", with the value BASE64URL (JWS Payload)
- "signature", with the value BASE64URL(JWS Signature)

Note: there will never be a JWS Unprotected Header for PASSporT.

First, an example PASSporT Protected Header is as follows:

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

This would be serialized to the form:

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

Encoding this with UTF8 and BASE64 encoding produces this value:

```
eyJ0eXAiOiJwYXNzcG9ydCIsImFsZyI6IkVTMjU2IiwieDV1IjoiaHR0cHM6Ly9j
ZXJ0LmV4YW1wbGUub3JnL3Bhc3Nwb3J0LmNlciJ9
```

Second, an example PASSporT Payload is as follows:

```
        {
        "iat":"1443208345",
        "orig":{"tn":"12155551212"},
        "dest":{"uri":"sip:alice@example.com"}
    }
```

This would be serialized to the form:

```
        {"iat":"1443208345","orig":{"tn":"12155551212"},"dest":
        {"uri":"sip:alice@example.com"}}
```

Encoding this with the UTF8 and BASE64 encoding produces this value:

```
     eyJpYXQiOiIxNDQzMjA4MzQ1Iiwib3JpZyI6eyJ0biI6IjEyMTU1NTUxMjEyIn0s
     ImRlc3QiOnsidXJpIjoic2lwOmFsaWNlQGV4YW1wbGUuY29tIn19
```

Computing the digital signature of the PASSporT Signing Input ASCII(BASE64URL(UTF8(JWS Protected Header)) || '.' || BASE64URL(JWS Payload))

```
     2bbTbLeDIf52Vv0yESUqebUBYrKIuouOfKQME6MD9kfgZ59dMAvvrIC94XsKdzV0
     3evDS8wd6CubUqSalM7Dpg
```

The final PASSporT token is produced by concatenating the values in the order Header.Payload.Signature with period (',') characters. For the above example values this would produce the following:

```
     eyJ0eXAiOiJwYXNzcG9ydCIsImFsZyI6IkVTMjU2IiwieDV1IjoiaHR0cHM6Ly9j
     ZXJ0LmV4YW1wbGUub3JnL3Bhc3Nwb3J0LmNlciJ9
     .
     eyJpYXQiOiIxNDQzMjA4MzQ1Iiwib3JpZyI6eyJ0biI6IjEyMTU1NTUxMjEyIn0s
     ImRlc3QiOnsidXJpIjoic2lwOmFsaWNlQGV4YW1wbGUuY29tIn19
     .
     2bbTbLeDIf52Vv0yESUqebUBYrKIuouOfKQME6MD9kfgZ59dMAvvrIC94XsKdzV0
     3evDS8wd6CubUqSalM7Dpg
```

A.1. X.509 Private Key Certificate for Example

```
     -----BEGIN EC PRIVATE KEY-----
     MHcCAQEEIFeZ1R208QCvcu5GuYyMfG4W7sH4m99/7eHSDLpdYllFoAoGCCqGSM49
     AwEHoUQDQgAE8HNbQd/TmvCKwPKHkMF9fScavGeH78YTU8qLS8I5HLHSSmlATLcs
     lQMhNC/OhlWBYC626nIlo7XeebYS7Sb37g==
     -----END EC PRIVATE KEY-----
```

A.2. X.509 Public Key Certificate for Example

```
-----BEGIN PUBLIC KEY-----
MFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAE8HNbQd/TmvCKwPKHkMF9fScavGeH
78YTU8qLS8I5HLHSSmlATLcslQMhNC/OhlWBYC626nIlo7XeebYS7Sb37g==
-----END PUBLIC KEY-----
```

Authors' Addresses

Chris Wendt
Comcast
One Comcast Center
Philadelphia, PA 19103
USA

Email: chris-ietf@chriswendt.net

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

Email: jon.peterson@neustar.biz