STIR
Internet-Draft
Intended status: Standards Track
Expires: November 25, 2016

C. Wendt
Comcast
J. Peterson
Neustar Inc.

May 24, 2016

Persona Assertion Token
draft-ietf-stir-passport-02

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

Copyright Notice

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. 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.crt"
}
```

3.1.1. "typ" (Type) Header Parameter

JWS defines the "typ" (Type) Header Parameter to declare the media type [IANA.MediaTypes] 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. Note with extensions explained later in this

document, the typ may be another value if defined as a passport extension.

3.1.2. "alg" (Algorithm) Header Parameter

For PASSporT, the "alg" should be defined as follows, for the creation of PASSporT tokens and their corresponding digital signatures,

- ES256 MUST be implemented.

- RS256 SHOULD be implemented

For the verification of PASSporT tokens, both ES256 and RS256 MUST be supported.

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 Identities

Baseline PASSporT defines claims that convey the identity of the origination and destination of personal communications represented as either telephone numbers or Uniform Resource Indicators (URIs). Some using protocols may require other identifiers for personae; these may be specified as claims through the PASSporT extensibility mechanisms. But for telephone numbers and URIs, the following claims should be used:

3.2.2.1.1. "otn" and "dtn" - Originating and Destination Telephone Number claim

If the originating identity is a telephone number, the claim "otn" SHOULD be included. If the destination identity is a telephone number, the claim "dtn" SHOULD be included.

Telephone Number strings for "otn" and "dtn" claims MUST be canonicalized according to the procedures specified in [I-D.ietf-stir-rfc4474bis] Section 6.1.1.

3.2.2.1.2. "ouri" and "duri" - Originating and Destination URI claims

If the originating identity is not a telephone number, the claim "ouri" SHOULD be included with the string cooresponding to the URI form of the identity as defined in [RFC3986], alternatively it could also contain an application specific identity string, if URI format is not appropriate.

If the destination identity is not a telephone number, the claim "duri" SHOULD be included. The same string format rules apply as stated for "ouri".

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 'algorithm' and 'digest' 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.

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",
    "otn":"12155551212",
    "duri":"sip:alice@example.com",
    "mky":"[
    {
        "algorithm":"sha-256",
        "digest":"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"
    },
    {
        "algorithm":"sha-256",
        "digest":"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"
    }]"
}
```

3.2.3. Multi-party Communications

Personal communications in the context of PASSporT can certainly extend to multi-party scenerios where there is more than one destination identity. In the future, it is anticipated that PASSporT will be extended to support these cases.

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, however there will certainly be new and extended applications and usage of PASSPorT that will need to extend the claims to represent other information specific to the origination identities beyond the identity itself.

There are two mechanisms defined to extend PASSporT. The first includes an extension of the base passport claims to include additional claims. An alternative method of extending PASSporT is for applications of PASSporT unrelated to the base set of claims, that will define it's own set of claims. Both are described below.

4.1. "ppt" (PASSporT) header parameter

For extended profiles of PASSporT, a new JWS header parameter "ppt" MUST be used with a string that uniquely identifies the 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.crt"
}
```

4.2. Extended PASSporT Claims

Future specifications that define such extensions to the PASSporT mechanism MUST explicitly designate what claims they include, 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.

4.3. Alternate PASSporT Extension

Some applications may want to use the mechanism of the PASSporT digital signature that is not a superset of the base set of claims of the PASSporT token as defined in Section 3. Rather, a specification may use PASSporT with its own defined set of claims.

In this case, the specification SHOULD define its own MIME media type [RFC2046] in the "Media Types" registry [IANA.MediaTypes]. The MIME subtype SHOULD start with the string "passport-" to signify that it is related to the PASSporT token. For example, for the "foo" application the MIME type/sub-type could be defined as "application/passport-foo".

4.4. Registering PASSporT Extensions

Toward interoperability and to maintain uniqueness of the extended PASSporT profile header parameter string, there SHOULD be an industry registry that tracks the definition of the profile strings.

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,"otn":"12155551212","duri":
"sip:alice@example.com","mky":{"algorithm":"sha-256","digest":
"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;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"}}
```

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.

- either "dtn" claim or "duri" 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 [IANA.MediaTypes] 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: "otn"

- Claim Description: Originating Telephone Number String

- Change Controller: IESG

- Specification Document(s): Section 3.2 of draft-ietf-stir-passport-00

- Claim Name: "dtn"

- Claim Description: Destination Telephone Number String

- Change Controller: IESG

- Specification Document(s): Section 3.2 of draft-ietf-stir-passport-00

- Claim Name: "ouri"

- Claim Description: Originating URI String

- Change Controller: IESG

- Specification Document(s): Section 3.2 of draft-ietf-stir-passport-00

- Claim Name: "duri"

- Claim Description: Destination URI 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

[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-07 (work in progress), February 2016.

[IANA.MediaTypes]
"IANA, "Media Types"", <Media Types>.

[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>.

[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>.

[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>.

[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>.

[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/>.

Appendix A. Example ES256 base PASSporT JWS Serialization and Signature

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

- o "protected", with the value BASE64URL(UTF8(JWS Protected Header))

- o "payload", with the value BASE64URL (JWS Payload)

- o "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.crt"
}
```

This would be serialized to the form:

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

Encoding this with UTF8 and BASE64 encoding produces this value:

```
eyJ0eXAiOiJwYXNzcG9ydCIsImFsZyI6IkVTMjU2IiwieDV1IjoiaHR0cHM6Ly9j
ZXJ0LmV4YW1wbGUub3JnL3Bhc3Nwb3J0LmNydCJ9
```

Second, an example PASSporT Payload is as follows:

```
{
    "iat":"1443208345",
    "otn":"12155551212",
    "duri":"sip:alice@example.com"
}
```

This would be serialized to the form:

```
{"iat":"1443208345","otn":"12155551212","duri":
"sip:alice@example.com"}
```

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

```
eyJpYXQiOiIxNDQzMjA4MzQ1Iiwib3RuIjoiMTIxNTU1NTEyMTIiLCJkdXJp
Ijoic2lwOmFsaWNlQGV4YW1wbGUuY29tIn0
```

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

```
KK89q2RFY-BkKQQhiB0z6-fIaFUy6NDyUboKXOix9XnYLxTCjdw1UHjCbw4Ce
feKwH_t7W-bnGlZz4pI-rMjfQ
```

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
ZXJ0LmV4YW1wbGUub3JnL3Bhc3Nwb3J0LmNydCJ9
.
eyJpYXQiOiIxNDQzMjA4MzQ1Iiwib3RuIjoiMTIxNTU1NTEyMTIiLCJkdXJpIjoi
c2lwOmFsaWNlQGV4YW1wbGUuY29tIn0
.
KK89q2RFY-BkKQQhiB0z6-fIaFUy6NDyUboKXOix9XnYLxTCjdw1UHjCbw4CefeK
wH_t7W-bnGlZz4pI-rMjfQ
```

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

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

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

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

Appendix B. Example RS256 base 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":"RS256",
    "x5u":"https://cert.example.org/passport.crt"
}
```

This would be serialized to the form:

```
{"typ":"passport","alg":"RS256","x5u":"https://cert.example.org/
passport.crt"}
```

Encoding this with UTF8 and BASE64 encoding produces this value:

```
eyJ0eXAiOiJwYXNzcG9ydCIsImFsZyI6IlJTMjU2IiwieDV1IjoiaHR0cHM
6Ly9jZXJ0LmV4YW1wbGUub3JnL3Bhc3Nwb3J0LmNydCJ9
```

Second, an example PASSporT Payload is as follows:

```
{
    "iat":"1443208345",
    "otn":"12155551212",
    "duri":"sip:alice@example.com"
}
```

This would be serialized to the form:

```
{"iat":"1443208345","otn":"12155551212","duri":
"sip:alice@example.com"}
```

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

```
eyJpYXQiOiIxNDQzMjA4MzQ1Iiwib3RuIjoiMTIxNTU1NTEyMTIiLCJkdXJp
Ijoic2lwOmFsaWNlQGV4YW1wbGUuY29tIn0
```

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

```
AaeXRqm7kHnkZu2j6cQmDCiomZRiaE55bYWhFgnX8xMqpBFq96M0xgMM5OLa9
_LMrkuKv2ivK5GZz8OlFrmAirucRlAh8YdUkj5Cr5xPRr-gg9acD9jqJUnQ-Z
xpL1yq-FFVLhvpbsE5NMPHXUp5lpt62rD-S0NlhwHNCeMqZHxt6T5BmZBXITE
d1PRRij_6FhE3wxWEhZMthWJuEbcPpRMZDu-R7lTNddn62nUKjn3s00R3gm25
Dto5Z0dzfQpAysJvnbc1QRimfsYqJPUFc57lnglVLf4WrpeZCc8-LcoXeSr_d
seDgsrmg2EuHmn5h1nTOmLgF16ZHm121ZVjiXz2sMFvs9RaIxw0AFkM7rnV56
OxAFCRuzMNldiEVf8plRZVvqZ4BfVQlCNXNyyVgPOUtNr3ta6yD2H0oANQvvH
twjuSwB9Kruj4Wsu5N7Iki4MBs6SWJDmcUV-NW_AHYLaao-IvFVe4oCkJNjsq
wwXuLv1TO2sDHdc5sQO5zm21019PPxw1udHVtywsRVNKLo0RzE0TqYUF7XclC
Dur7MMOx9SnStV2PFIM7Jejyn9x54RtJEjOnchaSalfIFr_UXqXgVmRZVTzLD
QIlcmHjlhhLnCnNx3sYsAANen8Y8jtfgJ2ewjGotB4Lq8VYe1FacBKKk0VyCf
ImXba0u1hB8Q
```

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:

```
eyJ0eXAiOiJwYXNzcG9ydCIsImFsZyI6IlJTMjU2IiwieDV1IjoiaHR0cHM6L
y9jZXJ0LmV4YW1wbGUub3JnL3Bhc3Nwb3J0LmNydCJ9
.
eyJpYXQiOiIxNDQzMjA4MzQ1Iiwib3RuIjoiMTIxNTU1NTEyMTIiLCJkdXJpI
joic2lwOmFsaWNlQGV4YW1wbGUuY29tIn0
.
AaeXRqm7kHnkZu2j6cQmDCiomZRiaE55bYWhFgnX8xMqpBFq96M0xgMM5OLa9
_LMrkuKv2ivK5GZz8OlFrmAirucRlAh8YdUkj5Cr5xPRr-gg9acD9jqJUnQ-Z
xpL1yq-FFVLhvpbsE5NMPHXUp5lpt62rD-S0NlhwHNCeMqZHxt6T5BmZBXITE
d1PRRij_6FhE3wxWEhZMthWJuEbcPpRMZDu-R7lTNddn62nUKjn3s00R3gm25
Dto5Z0dzfQpAysJvnbc1QRimfsYqJPUFc57lnglVLf4WrpeZCc8-LcoXeSr_d
seDgsrmg2EuHmn5h1nTOmLgF16ZHm121ZVjiXz2sMFvs9RaIxw0AFkM7rnV56
OxAFCRuzMNldiEVf8plRZVvqZ4BfVQlCNXNyyVgPOUtNr3ta6yD2H0oANQvvH
twjuSwB9Kruj4Wsu5N7Iki4MBs6SWJDmcUV-NW_AHYLaao-IvFVe4oCkJNjsq
wwXuLv1TO2sDHdc5sQO5zm21019PPxw1udHVtywsRVNKLo0RzE0TqYUF7XclC
Dur7MMOx9SnStV2PFIM7Jejyn9x54RtJEjOnchaSalfIFr_UXqXgVmRZVTzLD
QIlcmHjlhhLnCnNx3sYsAANen8Y8jtfgJ2ewjGotB4Lq8VYe1FacBKKk0VyCf
ImXba0u1hB8Q
```

B.1. X.509 Private Key Certificate for Example in Appendix A

```
-----BEGIN RSA PRIVATE KEY-----
MIIJKQIBAAKCAgEAsrKb3NsMgrXTzEcNlg3vaBbI12mG3D9QBn61H8PpsVFIh3MA
XNXjkV64he/eEQou3STTEgSqUXj5kj+jnnVFF0Cd0T6j7SuRvpq5YaiKfXgdUlsv
F3LjTRGyoKRNOf16f/zEFiyJBX10vj/LKvnWos1vVTSqBeui2dNLynr0+f1n8b0+
0FZwACceo3qaVwuSNIJWSQgM1qAINBpPEKnrIpdt5fa7mUorJ5gjITys3gjNJ4ee
sjqUEu5ZGXDgMshVtH2iMceC1393sK6rJ7z+g3jVziSo6vy9lA2wveKMuoqQTwp0
V0IrkzExU7vpTzyx0E3mJNmlgmDp7Whp2HCvKjeG+iPfsuPMDRggUrdy9qG6QTFq
QORzLywTpu78ExYMSfqt94NVhf2Dv+QEPoytT1avN6bwGu/R/84g2z0YMfum5roS
TG5PGP4H56vjML8wNTd6v8Ny8SLAgzG/XBaV7c8Ll2awLEj4FSeBpNzTyDgnLrth
7Tk0LmM8EtO1aozDDEaMFrNy4/L+Uuwxp/wcFADawE9N7VdHa9endEo9V/bu1tkq
ecv1Ma+G4NvJZzD8JTBRVsHNc3zvI0qD0KWCjqPvaIMiiVATVAIW9ZEtUZNm5UVu
DzhcY7QrXNRpGE6ULIXgim66mbfUQ0LFq4G+zUjoRZTA92rFBn4vDKvsPs8CAwEA
AQKCAgEAk3Sc9sbucOGXbuZmyJ6hIhRDELXsacv4vhNKZHbmXMJFBjgYYYLBsRAn
VaZUaV0sxKEBZsngvTAFSPAolLYSGBji4Wo+HJQqRM0qEfLgrJ40G+RQXJoaBFuJ
OdO6QhLvRbOPHvkK7DPU5LSBcuoMefTpXLcDYbVKgVJBJUkG405+ulS+A26AJzAg
sSeXOiK7N4chqkvxRB10B4J6IbcE51trfSp3LQutxpNc0a1evC0pFOhtRIbf7yss
7Lhe2KjFSBWvhEIsrqRpYKsRYs4qetR4IQ3RMW7zBLuzT00lcTyrzf1DeUmJ6YDb
Qkw6Pn9H/yp4sYnDcG2GOrhBNy/s6ZecANLDbKg6MqwszDqLZnIOh7zPV1MPGEjc
LkfLue1CA7FaipDUAlSYDfkaNHEcGFxHEgTuv3zmcuMijgNzCtA71M/6kG41DZa1
8PZmqcw8CmMo+1UD3QwL5hHvMbeCyq1UZQvrTmwSLaGjC/goTjChzrsq5NBQcNNb
eiGUFXciqJlh10NfxT8arefoQ/EDuARiZNvwGDqrGkvZk3/xokGeQi6nf1DL4NtU
wQJyzVDJERFs9SohwkJWlPACFxISbxBztyw3nUvGN2iUQdnglGXmwmo7Ork0uook
R2TV1OS7INVOTiEx8AApdiFXWZ752pB96ww6s5pDP3Isp0yxddECggEBAOuSVK2/
7v9aCzlf+IZiklTkpG9CRkBqEsIj6p9ADBMWahqxlzjKwzGJH55v0U/etqOOZYVV
HFzHIzlfZN1Yy1LXYKKcrwU23rLLiG5IsYFCcX2t1Cw6ZxHEsuMsOi8X8IbhSjJe
xTGmwYiJqdKlSyznFopPtZ0leVqMjHTAMCk36AzRwTMnjOIhA1p1Ru0HPFK1RF/5
/EuAUPws2ur1CDjsJwOQa2gRpInbxzZMCE/J+OqgWz4DMivLMCYT40zSjvY7+sqW
```

```
K77khwSm9/wMhuvVDedXHaNcrrQQbrk89oYt0Lx74RjUGc/nF79loQDOTZ/Hc7Fq
1Nj2cuw0rIdJoUkCggEBAMIxrp4jSjdNT62WpnTfejJUdLVvn+3zvuNWcwIpUrIl
ILBINlVofMMKLIi0VuqFc7tJiim+dUufp+taoj4E2rPumxZGMb7m9/XGFIyDY+2j
qJEin6kK8WMT5he94C5uQg3faSzMi+sbEa4HSXMhBOP6iLqSQyUZpq1ecRjOdnDk
bWCAHoRRYKSaPMJRdQjHD5++hItLCo2MiwVFBl2nRVh3vHIYARY3K84BMnvbUayd
nfZB/tGOuvTksMRHcqoDFgXNj5/ymqBzoSpQzCMfH79Sv1uQPkDqKO/YbBT3HVDv
6nKX44Vv4iy9Xwqsv3nTtuq2gpFJU30tfHBVltYB91cCggEAUaJhE+EaeoUCtLxM
TI2mNiMR1Lh7zeC0ZXC64rr4NDklReDbDcQ+RlFFkssfFvWQBzfWeJEZBhHAZCZp
tscJlsiqZU+02zK7k+wyeD1avfd/itUNXNJUW3T1pQHzm9RI9wTliHUNEvq9wIos
PqInXgUq631Z635MApQILIFZbz8/fAnIUOjYypg0KEnR7Vv/jI3ihvwDcUqjRfBp
YNjPI6K6lmKaxfKvOVLfQzKwAq50QyKU2/WRklmUcu2bbEjfX/dDHqdRu5JIM9WE
xGS28MzhR5UJ4U3CAQZcyHaW28LOvjKTu93sn/5uXVZjp/rWLZOZxRbHcfRduPs7
+poKeQKCAQBc8nqppip3ncFtTJYPiodqX5Ic5Xie4/ORzGbvueei7LJgra+T4ZcV
o2D9bZPMXGOwWNqQcGCj+Z7dv1u4Y4pqZOJGHwLgZJx6PnzHZHwH2jVsgi35Mwum
aHfRFUif8JYdHbmxf5XYyfQEX+h/+mXk2J1o72jD8Ssd//4R6YA3OJ5BehEhM/IV
1t0OBP8HXH/V7dJy+U/rwEEqHIeXe+BtH6JK2cJrZ6zHxTrsnWTSQf7BR4U3uCEz
5eHVkH0JcsCvtlvwKqZn9fBF2LZceSEw6eI9aSTi3TEK24Of5Uda3fpRLvHvhEW1
NE6xRU3Aed0rKoAEGhyj5YmSGuU/OWGxAoIBAQDbREen8GWGLFmj0iQFs0I2Jr1k
1iazomLyR9Vvhe8sUu57mE0lKbFo6vt8RPm69NSJ7nMCrSbCwG+qERMdMLK8OuiY
v+W3wvvKcpXCShJ1GpgqKmBdP4VnHKvgHQ/kzdtLDmJI4SkTim1Mi94szSMPIfQw
cMdZAGivDPjdXw95xENLClPOkhjX9t/qZjkZclQyjYCYGJHRxX6J7PdcKRY0/9VV
jgRwxooE2POv11/qSk1O3lhFvjjm5oxr7CKPcHvESk/r8mh+VWO4DaOD4gQ9ke00
2QGhocy3K578uL4ph7nfTR2QD96mxCNX9b2Pj9HG8Qb3wEvtaGBfUu8do2mT
-----END RSA PRIVATE KEY-----
```

B.2. X.509 Public Key Certificate for Example in Appendix A

```
-----BEGIN PUBLIC KEY-----
MIICIjANBgkqhkiG9w0BAQEFAAOCAg8AMIICCgKCAgEAsrKb3NsMgrXTzEcNlg3v
aBbI12mG3D9QBn61H8PpsVFIh3MAXNXjkV64he/eEQou3STTEgSqUXj5kj+jnnVF
F0Cd0T6j7SuRvpq5YaiKfXgdUlsvF3LjTRGyoKRNOf16f/zEFiyJBX10vj/LKvnW
os1vVTSqBeui2dNLynr0+f1n8b0+0FZwACceo3qaVwuSNIJWSQgM1qAINBpPEKnr
Ipdt5fa7mUorJ5gjITys3gjNJ4eesjqUEu5ZGXDgMshVtH2iMceC1393sK6rJ7z+
g3jVziSo6vy9lA2wveKMuoqQTwp0V0IrkzExU7vpTzyx0E3mJNmlgmDp7Whp2HCv
KjeG+iPfsuPMDRggUrdy9qG6QTFqQORzLywTpu78ExYMSfqt94NVhf2Dv+QEPoyt
T1avN6bwGu/R/84g2z0YMfum5roSTG5PGP4H56vjML8wNTd6v8Ny8SLAgzG/XBaV
7c8Ll2awLEj4FSeBpNzTyDgnLrth7Tk0LmM8EtO1aozDDEaMFrNy4/L+Uuwxp/wc
FADawE9N7VdHa9endEo9V/bu1tkqecv1Ma+G4NvJZzD8JTBRVsHNc3zvI0qD0KWC
jqPvaIMiiVATVAIW9ZEtUZNm5UVuDzhcY7QrXNRpGE6ULIXgim66mbfUQ0LFq4G+
zUjoRZTA92rFBn4vDKvsPs8CAwEAAQ==
-----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