Workgroup:	anima Working Group
Internet-Draft:	draft-ietf-anima-jws-voucher-13
Published:	25 October 2024
Intended Status:	Standards Track
Expires:	28 April 2025
Authors:	T. Werner	M. Richardson
---	---	---
	Siemens AG	*Sandelman Software Works*

JWS signed Voucher Artifacts for Bootstrapping Protocols

Abstract

I-D.ietf-anima-rfc8366bis defines a digital artifact (known as a voucher) as a YANG-defined JSON document that is signed using a Cryptographic Message Syntax (CMS) structure. This document introduces a variant of the voucher artifact in which CMS is replaced by the JSON Object Signing and Encryption (JOSE) mechanism described in RFC7515 to support deployments in which JOSE is preferred over CMS. In addition to specifying the format, the "application/voucher-jws+json" media type is registered and examples are provided.

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 28 April 2025.

Copyright Notice

Table of Contents

1. Introduction

"A Voucher Artifact for Bootstrapping Protocols" [I-D.ietf-anima-rfc8366bis] defines a YANG-based data structure used in "Bootstrapping Remote Secure Key Infrastructure" (BRSKI) [RFC8995] and "Secure Zero Touch Provisioning" (SZTP) [RFC8572] to transfer ownership of a device from a manufacturer to a new owner (customer or operational domain). That document provides a serialization of the voucher data to JSON [RFC8259] with cryptographic signing according to the Cryptographic Message Syntax (CMS) [RFC5652]. That resulting voucher artifact has the media type `application/voucher-cms+json`.

This document provides cryptographic signing of voucher data in form of JSON Web Signature (JWS) [RFC7515] and the media type `application/voucher-jws+json` to identify the voucher format. The encoding specified in this document is used by [I-D.ietf-anima-brski-prm] and may be more handy for use cases already using Javascript Object Signing and Encryption (JOSE).

This document should be considered as enhancement of [I-D.ietf-anima-rfc8366bis], as it provides a new voucher format. It is similar to [I-D.ietf-anima-constrained-voucher], which provides cryptographic signing according COSE [RFC8812] and the media type `application/voucher-cose+cbor`. These documents do not change nor extend the YANG definitions of [I-D.ietf-anima-rfc8366bis].

With the availability of different voucher formats, it is up to an industry-specific application statement to decide which format is to be used. The associated media types are used to distinguish different voucher formats.

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.

This document uses the following terms:

JSON Voucher Data: An unsigned JSON representation of the voucher data.

JWS Voucher: A JWS structure signing the JSON Voucher Data.

Voucher: A short form for voucher artifact and refers to the signed statement from Manufacturer Authorized Signing Authority (MASA) service that indicates to a Pledge the cryptographic identity of the domain it should trust, per [I-D.ietf-anima-rfc8366bis].

Voucher Data: The raw (serialized) representation of the `ietf-voucher` YANG module without any enclosing signature, per [I-D.ietf-anima-rfc8366bis].

MASA (Manufacturer Authorized Signing Authority): The entity that, for the purpose of this document, issues and signs the vouchers for manufacturer's pledges. In some onboarding protocols, the MASA may have an Internet presence and be integral to the onboarding process, whereas in other protocols the MASA may be an offline service that has no active role in the onboarding process, per [I-D.ietf-anima-rfc8366bis].

Pledge: The prospective component attempting to find and securely join a domain. When shipped or in factory reset mode, it only trusts authorized representatives of the manufacturer, per [I-D.ietf-anima-rfc8366bis].

Registrar: A representative of the domain that is configured, perhaps autonomically, to decide whether a new device is allowed to join the domain, per [I-D.ietf-anima-rfc8366bis].

This document uses the following encoding notations:

BASE64URL(OCTETS): Denotes the base64url encoding of OCTETS, per Section 2 of [RFC7515].

UTF8(STRING): Denotes the octets of the UTF-8 [RFC3629] representation of STRING, per Section 1 of [RFC7515].

3. Voucher Artifact with JSON Web Signature

JWS voucher artifacts MUST use the "General JWS JSON Serialization Syntax" defined in Section 7.2.1 of [RFC7515]. This syntax supports multiple signatures as already supported by [RFC8366] for CMS-signed vouchers. The following figure summarizes the serialization of JWS voucher artifacts:

```
{
  "payload": BASE64URL(UTF8(JSON Voucher Data)),
  "signatures": [
    {
      "protected": BASE64URL(UTF8(JWS Protected Header)),
      "signature": BASE64URL(JWS Signature)
    }
  ]
}
```

Figure 1: Voucher Representation in General JWS JSON Serialization Syntax (JWS Voucher)

The JSON Voucher Data MUST be UTF-8 encoded to become the octet-based JWS Payload defined in [RFC7515]. The JWS Payload is further base64url-encoded to become the string value of the `payload` member as described in Section 3.2 of [RFC7515]. The octets of the UTF-8 representation of the JWS Protected Header are base64url-encoded to become the string value of the `protected` member. The generated JWS Signature is base64url-encoded to become the string value of the `signature` member.

3.1. JSON Voucher Data

The JSON Voucher Data is an unsigned JSON document [RFC8259] that conforms with the data model described by the ietf-voucher YANG module [RFC7950] defined in Section 7.3 of [I-D.ietf-anima-rfc8366bis] and is encoded using the rules defined in [RFC7951]. The following figure provides an example of JSON Voucher Data:

```
{
  "ietf-voucher:voucher": {
    "assertion": "logged",
    "serial-number": "0123456789",
    "nonce": "5742698422680472",
    "created-on": "2022-07-08T03:01:24.618Z",
    "pinned-domain-cert": "base64encodedvalue=="
  }
}
```

Figure 2: JSON Voucher Data Example

3.2. JWS Protected Header

The JWS Protected Header defined in [RFC7515] uses the standard header parameters `alg`, `typ`, and `x5c`:

- The `alg` parameter MUST contain the algorithm type (e.g., `ES256`) used to create the signature as defined in Section 4.1.1 of [RFC7515].
- The `typ` parameter is optional and used when more than one kind of object could be present in an application data structure as described in Section 4.1.9 of [RFC7515]. If present, the `typ` parameter MUST contain the value `voucher-jws+json`.
- If X.509 (PKIX) certificates [RFC5280] are used, the `x5c` parameter MUST contain the base64-encoded (not base64url-encoded) X.509 v3 (DER) certificate as defined in Section 4.1.6 of [RFC7515] and SHOULD also contain the certificate chain.

Implementation Note: base64-encoded values, in contrast to base64url-encoded values, may contain slashes (`/`). JSON [RFC8259] optionally allows escaping these with backslashes (`\\`). Hence, depending on the JSON parser/serializer implementation used, they may or may not be included. JWS Voucher parsers need to be prepared accordingly to extract certificates correctly.

To validate voucher signatures, all certificates of the certificate chain are required up to the trust anchor. Note, to establish trust the trust anchor SHOULD be provided out-of-band up front.

The following figure gives an example of a JWS Protected Header:

```
{
  "alg": "ES256",
  "typ": "voucher-jws+json",
  "x5c": [
    "base64encodedvalue1==",
    "base64encodedvalue2=="
  ]
}
```

Figure 3: JWS Protected Header Example

3.3. JWS Signature

The JWS Signature is generated over the JWS Protected Header and the JWS Payload (= UTF-8 encoded JSON Voucher Data) as described in Section 5.1 of [RFC7515].

4. Privacy Considerations

The Pledge-Voucher-Request (PVR) reveals the IDevID of the component (Pledge) that is in the process of bootstrapping.

A PVR is transported via HTTP-over-TLS. However, for the Pledge-to-Registrar TLS connection a Pledge provisionally accepts the Registrar server certificate during the TLS server authentication. Hence, it is subject to disclosure by a Dolev-Yao attacker (a "malicious messenger") [ON-PATH], as explained in Section 10.2 of [RFC8995].

The use of a JWS header brings no new privacy considerations.

5. Security Considerations

The issues of how [I-D.ietf-anima-rfc8366bis] vouchers are used in a [BRSKI] system is addressed in Section 11 of [RFC8995]. This document does not change any of those issues, it just changes the signature technology used for voucher request and response artifacts.

Section 9 of [RFC8572] deals with voucher use in Secure Zero Touch Provisioning (SZTP), for which this document also makes no changes to security.

6. IANA Considerations

6.1. Media-Type Registry

This section registers `application/voucher-jws+json` in the "Media Types" registry.

6.1.1. application/voucher-jws+json

```
Type name:  application
Subtype name:  voucher-jws+json
Required parameters:  none
Optional parameters:  none
Encoding considerations:  JWS+JSON vouchers are JOSE objects
                          signed with one or multiple signers.
Security considerations:  See section [Security Considerations]
Interoperability considerations:  The format is designed to be
  broadly interoperable.
Published specification:  [THIS RFC].
Applications that use this media type:  ANIMA, 6tisch, and other
  zero-touch bootstrapping/provisioning solutions
Additional information:
  Magic number(s):  None
  File extension(s):  .vjj
  Macintosh file type code(s):  none
Person & email address to contact for further information:  IETF
  ANIMA WG
Intended usage:  LIMITED
Restrictions on usage:  NONE
Author:  ANIMA WG
Change controller:  IETF
Provisional registration? (standards tree only):  NO
```

7. Acknowledgments

We would like to thank the various reviewers for their input, in particular Steffen Fries, Ingo Wenda, Esko Dijk and Toerless Eckert. Thanks for the supporting PoC implementations to Hong Rui Li and He Peng Jia.

8. Examples

These examples are folded according to the [RFC8792] Single Backslash rule.

8.1. Example Pledge-Voucher-Request (PVR)

The following is an example of a Pledge-Voucher-Request (PVR) as JWS Voucher artifact, which would be sent from a Pledge to the Registrar:

```
{
  "payload": "eyJpZXRmLXZvdWNoZXItcmVxdWVzdDp2b3VjaGVyIjp7InNlcmlhbC\
1udW1iZXIiOiIwMTIzNDU2Nzg5Iiwibm9uY2UiOiI2R3RuK1pRS04ySHFERlZrQkV4Wk\
xRPT0iLCJjcmVhdGVkLW9uIjoiMjAyMi0wNy0wOFQwODo0MDo0Mi44MjBaIiwicHJveG\
ltaXR5LXJlZ2lzdHJhci1jZXJ0IjoiTUlJQjRqQ0NBWWlnQXdJQkFnSUdBWFk3MmJiWk\
1Bb0dDQ3FHU000OUJBTUNNRFV4RXpBUkJnTlZCQW9NQ2sxNVFuVnphVzVsYzNNeERUQU\
xCZ05WQkFjTUJGTnBkR1V4RHpBTkJnTlZCQU1NQmxSbGMzUkRRVEFlRncweU1ERXlNRG\
N3TmpFNE1USmFGdzB6TURFeU1EY3dOakU0TVRKYU1ENHhFekFSQmdOVkJBb01DazE1UW\
5WemFXNWxjM014RFRBTEJnTlZCQWNNQkZOcGRHVXhHREFXQmdOVkJBTU1EMFJ2YldGcG\
JsSmxaMmx6ZEhKaGNqQlpNQk1HQnlxR1NNNDlBZ0VHQ0NxR1NNNDlBd0VIQTBJQUJCaz\
E2Sy9pNzlvUmtLNVliZVBnOFVTUjgvdXMxZFBVaVpITXRva1NkcUtXNWZuV3NCZCtxUk\
w3V1JmZmVXa3lnZWJvSmZJbGx1cmNpMjV3bmhpT1ZDR2plekI1TUIwR0ExVWRKUVFXTU\
JRR0NDc0dBUVVGQndNQkJnZ3JCZ0VGQlFjREhEQU9CZ05WSFE4QkFmOEVCQU1DQjRBd1\
NBWURWUjBSQkVFd1A0SWRjbVZuYVhOMGNtRnlMWFJsYzNRdWMybGxiV1Z1Y3kxaWRDNX\
VaWFNDSG5KbFoybHpkSEpoY2kxMFpYTjBOaTV6YVdWdFpXNXpMV0owTG01bGREQUtCZ2\
dxaGtqT1BRUURBZ05JQURCRkFpQnhsZEJoWnEwRXY1SkwyUHJXQ3R5UzZoRFlXMXlDTy\
9SYXVicEM3TWFJRGdJaEFMU0piZ0xuZ2hiYkFnMGRjV0ZVVm8vZ0dOMC9qd3pKWjBTbD\
JoNHhJWGsxIn19",
  "signatures": [{
    "protected": "eyJ4NWMiOlsiTUlJQitUQ0NBYUNnQXdJQkFnSUdBWG5WanNVNU\
1Bb0dDQ3FHU000OUJBTUNNRDB4Q3pBSkJnTlZCQVlUQWtGUk1SVXdFd1lEVlFRS0RBeE\
thVzVuU21sdVowTnZjbkF4RnpBVkJnTlZCQU1NRGtwcGJtZEthVzVuVkdWemRFTkJNQ0\
FYRFRJeE1EWXdOREExTkRZeE5Gb1lEems1T1RreE1qTXhNak0xT1RVNVdqQlNNUXN3Q1\
FZRFZRUUdFd0pCVVRFVk1CTUdBMVVFQ2d3TVNtbHVaMHBwYm1kRGIzSndNUk13RVFZRF\
ZRUUZFd293TVRJek5EVTJOemc1TVJjd0ZRWURWUVFEREE1S2FXNW5TbWx1WjBSbGRtbG\
paVEJaTUJNR0J5cUdTTTQ5QWdFR0NDcUdTTTQ5QXdFSEEwSUFCQzc5bGlhUmNCalpjRU\
VYdzdyVWVhdnRHSkF1SDRwazRJNDJ2YUJNc1UxMWlMRENDTGtWaHRVVjIxbXZhS0N2TX\
gyWStTTWdROGZmd0wyM3ozVElWQldqZFRCek1Dc0dDQ3NHQVFVRkJ3RWdCQjhXSFcxaG\
MyRXRkR1Z6ZEM1emFXVnRaVzV6TFdKMExtNWxkRG81TkRRek1COEdBMVVkSXdRWU1CYU\
FGRlFMak56UFwvU1wva291alF3amc1RTVmdndjWWJNQk1HQTFVZEpRUU1NQW9HQ0NzR0\
FRVUZCd01DTUE0R0ExVWREd0VCXC93UUVBd0lIZ0RBS0JnZ3Foa2pPUFFRREFnTkhBRE\
JFQWlCdTN3UkJMc0pNUDVzTTA3MEgrVUZyeU5VNmdLekxPUmNGeVJST2xxcUhpZ0lnWE\
NtSkxUekVsdkQycG9LNmR4NmwxXC91eW1UbmJRRERmSmxhdHVYMlJvT0U9Il0sInR5cC\
I6InZvdWNoZXItandzK2pzb24iLCJhbGciOiJFUzI1NiJ9",
    "signature": "abVg4TDGzSTjVHkQlNeIW3ABu5ZXdMl1cEqwcIAlHFW4BrlGbO\
-DRTKfyCOGxSW49-ktJcrVlYgKqC4xmZoy0Q"
  }]
}
```


Figure 4: Example Pledge-Voucher-Request (PVR)

8.2. Example Parboiled Registrar-Voucher-Request (RVR)

The term parboiled refers to food which is partially cooked. In [BRSKI], the term refers to a Pledge-Voucher-Request (PVR) that was received by the Registrar, then has been processed by the Registrar ("cooked"), and is now being forwarded to the MASA.

The following is an example Registrar-Voucher-Request (RVR) as JWS Voucher artifact, which would be sent from the Registrar to the MASA. Note that the previous PVR can be seen in the payload in the field `prior-signed-voucher-request`.

```
{
  "payload": "eyJpZXRmLXZvdWNoZXItcmVxdWVzdDp2b3VjaGVyIjp7InNlcmlhbC\
1udW1iZXIiOiIwMTIzNDU2Nzg5IiwiaWRldmlkLWlzc3VlciI6IkJCZ3dGb0FVVkF1TT\
NNLzlMK1NpNk5EQ09Ea1RsKy9CeGhzPSIsIm5vbmNlIjoiNkd0bitaUUtOMkhxREZWa0\
JFeFpMUT09IiwicHJpb3Itc2lnbmVkLXZvdWNoZXItcmVxdWVzdCI6ImV5SndZWGxzYj\
JGa0lqb2laWGxLY0ZwWVVtMU1XRnAyWkZkT2IxcFlTWFJqYlZaNFpGZFdlbVJFY0RKaU\
0xWnFZVWRXZVVscWNEZEpiazVzWTIxc2FHSkRNWFZrVnpGcFdsaEphVTlwU1hkTlZFbD\
ZUa1JWTWs1Nlp6VkphWGRwWW0wNWRWa3lWV2xQYVVreVVqTlNkVXN4Y0ZKVE1EUjVVMG\
hHUlZKc1duSlJhMVkwVjJ0NFVsQlVNR2xNUTBwcVkyMVdhR1JIVm10TVZ6bDFTV3B2YV\
UxcVFYbE5hVEIzVG5rd2QwOUdVWGRQUkc4d1RVUnZNRTFwTkRSTmFrSmhTV2wzYVdOSV\
NuWmxSMngwWVZoU05VeFlTbXhhTW14NlpFaEthR05wTVdwYVdFb3dTV3B2YVZSVmJFcF\
JhbEp4VVRCT1FsZFhiRzVSV0dSS1VXdEdibE5WWkVKWFJtc3pUVzFLYVZkck1VSmlNR1\
JFVVROR1NGVXdNREJQVlVwQ1ZGVk9UbEpHVmpSU1dIQkNWV3RLYmxSc1drTlJWemxPVV\
RKemVFNVdSblZXYm5Cb1ZucFdjMWw2VGs1bFJWSlZVVlY0UTFvd05WZFJhMFpxVkZWS1\
IxUnVRbXRTTVZZMFVraHdRbFJyU201VWJGcERVVlV4VGxGdGVGTmlSMDE2Vld0U1VsWk\
ZSbXhTYm1OM1pWVXhSVkpZYkU1U1IwNHpWRzF3Ums1Rk1WVlRiVVpIWkhwQ05sUlZVa1\
psVlRGRldUTmtUMkZyVlRCVVZsSkxXVlV4UlU1SWFFWmxhMFpUVVcxa1QxWnJTa0ppTU\
RGRVlYcEZNVlZYTlZkbGJVWllUbGQ0YWswd01UUlNSbEpDVkVWS2JsUnNXa05SVjA1T1\
VXdGFUMk5IVWtoV1dHaElVa1ZHV0ZGdFpFOVdhMHBDVkZVeFJVMUdTakpaYkdSSFkwZE\
tjMU50ZUdGTmJYZzJXa1ZvUzJGSFRuRlJiSEJPVVdzeFNGRnViSGhTTVU1T1RrUnNRbG\
93VmtoUk1FNTRVakZPVGs1RWJFSmtNRlpKVVZSQ1NsRlZTa05oZWtVeVUzazVjRTU2Yk\
haVmJYUk1UbFpzYVZwV1FtNVBSbFpVVldwbmRtUllUWGhhUmtKV1lWWndTVlJZVW5aaE\
1VNXJZMVYwV0U1WFduVldNMDVEV2tOMGVGVnJkek5XTVVwdFdtMVdXR0V6Ykc1YVYwcD\
JVMjFhU21KSGVERmpiVTV3VFdwV00ySnRhSEJVTVZwRVVqSndiR1ZyU1RGVVZVbDNVak\
JGZUZaWFVrdFZWa1pZVkZWS1VsSXdUa1JqTUdSQ1ZWWldSMUZ1WkU1UmEwcHVXak5LUT\
Fvd1ZrZFJiRVpxVWtWb1JWRlZPVU5hTURWWFUwWkZORkZyUm0xUFJWWkRVVlV4UkZGcV\
VrSmtNVTVDVjFWU1YxVnFRbE5SYTFaR1pERkJNRk5YVW1waVZscDFXVlpvVDAxSFRuUl\
NibXhOVjBaS2MxbDZUbEprVjAxNVlrZDRhVll4V2pGWk0ydDRZVmRTUkU1WVZtRlhSaz\
JGZUZaWFVrdFZWa1pZVkZWS1VsSXdUa1JqTUdSQ1ZWWldSMUZ1WkU1UmEwcHVXak5LUT\
Fvd1ZrZFJiRVpxVWtWb1JWRlZPVU5hTURWWFUwWkZORkZyUm0xUFJWWkRVVlV4UkZGcV\
VrSmtNVTVDVjFWU1YxVnFRbE5SYTFaR1pERkJNRk5YVW1waVZscDFXVlpvVDAxSFRuUl\
NibXhOVjBaS2MxbDZUbEprVjAxNVlrZDRhVll4V2pGWk0ydDRZVmRTUkU1WVZtRlhSaz\
VFVTBjMVMySkdiM2xpU0hCclUwVndiMWt5YTNoTlJuQlpWR3BDVDJGVVZqWlpWbVJYWk\
Vad1dFNVljRTFXTUc5M1ZFY3dNV0pIVWtWUlZYUkRXakprZUdGSGRIRlVNVUpTVlZWU1\
Fsb3dOVXBSVlZKRFVtdEdjRkZ1YUhOYVJVcHZWMjVGZDFKWVdURlRhM2Q1VlVoS1dGRX\
pValZWZWxwdlVrWnNXRTFZYkVSVWVUbFRXVmhXYVdORlRUTlVWMFpLVWtka1NtRkZSaz\
FWTUhCcFdqQjRkVm95YUdsWmEwWnVUVWRTYWxZd1dsWldiVGgyV2pCa1QwMURPWEZrTT\
NCTFYycENWR0pFU205T1NHaEtWMGR6ZUVsdU1Ua2lMQ0p6YVdkdVlYUjFjbVZ6SWpwYm\
V5SndjbTkwWldOMFpXUWlPaUpsZVVvMFRsZE5hVTlzYzJsVVZXeEtVV2wwVlZFd1RrSl\
pWVTV1VVZoa1NsRnJSbTVUVldSQ1YwYzFWMkZ1VGxaT1ZURkNZakJrUkZFelJraFZNRE\
F3VDFWS1FsUlZUazVTUkVJMFVUTndRbE5yU201VWJGcERVVlpzVlZGWGRFZFZhekZUVm\
xoa1JtUXhiRVZXYkVaU1V6QlNRbVZGZEdoV2VsWjFWVEl4YzJSV2IzZFVibHBxWW10R0\
5GSnVjRUpXYTBwdVZHeGFRMUZWTVU1U1IzUjNZMGRLZEZwRmRHaFdlbFoxVm10a1YyVn\
RVa1pVYTBwT1VUQkdXVkpHVWtwbFJURkZWMWhrVDFKRlJYaFVhMUphWlVMVIySXhiRV\
ZsYlhNeFZERlNjbVZGTVhGVVdHaE9ZV3N3ZUZReFVsWk9WbVJ4VVd4T1RsVllUak5STV\
VaYVVrWmFVbFZWWkVaa01IQkRWbFpTUmxack1VTlVWV1JDVFZaV1JsRXlaRE5VVms1MF\
lraFdZVTFJUW5kWmJURnJVa2RKZWxOdVpFNVZhekV6VWxaR1dsSkdXbEpWVlZwR1pEST\
VNMVJXVWtwbGF6VkZWbFJLVDJWdFl6RlVWa3BxWkRCYVVsZFZVbGRWVmtaRlVrVkZNVk\
15UmxoT1Z6VlVZbGQ0TVZkcVFsTmlSMUowWWtkd1lWWkZTbUZVVlVwT1VqQktOV05WWk\
ZSVVZGRTFVVmRrUmxJd1RrUmpWV1JVVkZSUk5WRllaRVpUUlVWM1UxVkdRMUY2WXpWaV\
IyeG9WVzFPUTJGc2NHcFNWVlpaWkhwa2VWWlhWbWhrYmxKSVUydEdNVk5FVW5kaGVsSk\
tUa1JLTWxsVlNrNWpNVlY0VFZkc1RWSkZUa1JVUjNSWFlVaFNWbFpxU1hoaVdGcG9Vek\
JPTWxSWVozbFhVM1JVVkZka1VrOUhXbTFrTUhkNVRUTnZlbFpGYkZkUmJHUnhXa1pTUT\
JWck1VUmpNR1JFVVROT1NGRldSbFpTYTBvelVsZGtRMUZxYUZoVFJtTjRZVWROZVZKWV\
VtdFNNVm8yV2tWTk1XVnRSbGhXYmxKaFZucFdObFJHWkV0TlJYaDBUbGQ0YTFKSE9ERl\
VhMUpTWldzeFEwOUZaRUpOVmxaclUxaGtVbGRWTVVOWlZVWkhVbXhHVFdGck5UWlZSbm\
QyVlRGM2RtRXlPVEZoYkVZellXMWpNVkpVVm0xa2JtUnFWMWRLVGxGck1VaFJWRVpXV2\
tWd1VsVlZNVTVSVnpsSVVUQk9lbEl3UmxKV1ZWcERaREF4UkZSVlJUQlNNRVY0VmxkU1\
JXUXdWa05ZUXprelZWVldRbVF3YkVsYU1GSkNVekJLYmxvelJtOWhNbkJRVlVaR1VsSk\
ZSbTVVYTJoQ1VrVktSbEZYYkVOa1ZFNHpWV3RLVFdNd2NFNVZSRlo2VkZSQk0wMUZaM0\
```

```
pXVlZwNVpWVTFWazV0WkV4bGEzaFFWVzFPUjJWV1NsTlVNbmg0WTFWb2NGb3diRzVYUl\
U1MFUydDRWV1ZyVm5Oa2ExRjVZMGM1VEU1dFVqUk9iWGQ0V0VNNU1XVlhNVlZpYlVwU1\
VrVlNiVk50ZUdoa1NGWlpUV3hLZGxRd1ZUbEpiREJ6U1c1U05XTkRTVFpKYmxwMlpGZE\
9iMXBZU1hSaGJtUjZTekp3ZW1JeU5HbE1RMHBvWWtkamFVOXBTa1pWZWtreFRtbEtPU0\
lzSW5OcFoyNWhkSFZ5WlNJNkltRmlWbWMwVkVSSGVsTlVhbFpJYTFGc1RtVkpWek5CUW\
5VMVdsaGtUV3d4WTBWeGQyTkpRV3hJUmxjMFFuSnNSMkpQTFVSU1ZFdG1lVU5QUjNoVF\
Z6UTVMV3QwU21OeVZteFpaMHR4UXpSNGJWcHZlVEJSSW4xZGZRPT0iLCJjcmVhdGVkLW\
9uIjoiMjAyMi0wNy0wOFQwODo0MDo0Mi44NDhaIn19",
  "signatures": [{
    "protected": "eyJ4NWMiOlsiTUlJQm96Q0NBVXFnQXdJQkFnSUdBVzBlTHVJRk\
1Bb0dDQ3FHU000OUJBTUNNRFV4RXpBUkJnTlZCQW9NQ2sxNVFuVnphVzVsYzNNeERUQU\
xCZ05WQkFjTUJGTnBkR1V4RHpBTkJnTlZCQU1NQmxSbGMzUkRRVEFlRncweE9UQTVNVE\
V3TWpNM016SmFGdzB5T1RBNU1URXdNak0zTXpKYU1GUXhFekFSQmdOVkJBb01DazE1UW\
5WemFXNWxjM014RFRBTEJnTlZCQWNNQkZOcGRHVXhMakFzQmdOVkJBTU1KVkpsWjJsem\
RISmhjaUJXYjNWamFHVnlJRkpsY1hWbGMzUWdVMmxuYm1sdVp5QkxaWGt3V1RBVEJnY3\
Foa2pPUFFJQkJnZ3Foa2pPUFFNQkJ3TkNBQVQ2eFZ2QXZxVHoxWlVpdU5XaFhwUXNrYV\
B5N0FISFFMd1hpSjBpRUx0NnVOUGFuQU4wUW5XTVlPXC8wQ0RFaklrQlFvYnc4WUtxan\
R4SkhWU0dUajlLT295Y3dKVEFUQmdOVkhTVUVEREFLQmdnckJnRUZCUWNESERBT0JnTl\
ZIUThCQWY4RUJBTUNCNEF3Q2dZSUtvWkl6ajBFQXdJRFJ3QXdSQUlnWXIyTGZxb2FDS0\
RGNFJBY01tSmkrTkNacWRTaXVWdWdJU0E3T2hLUnEzWUNJRHhuUE1NbnBYQU1UclBKdV\
BXeWNlRVIxMVB4SE9uKzBDcFNIaTJxZ3BXWCIsIk1JSUJwRENDQVVtZ0F3SUJBZ0lHQV\
cwZUx1SCtNQW9HQ0NxR1NNNDlCQU1DTURVeEV6QVJCZ05WQkFvTUNrMTVRblZ6YVc1bG\
MzTXhEVEFMQmdOVkJBY01CRk5wZEdVeER6QU5CZ05WQkFNTUJsUmxjM1JEUVRBZUZ3MH\
hPVEE1TVRFd01qTTNNekphRncweU9UQTVNVEV3TWpNM016SmFNRFV4RXpBUkJnTlZCQW\
9NQ2sxNVFuVnphVzVsYzNNeERUQUxCZ05WQkFjTUJGTnBkR1V4RHpBTkJnTlZCQU1NQm\
xSbGMzUkRRVEJaTUJNR0J5cUdTTTQ5QWdFR0NDcUdTTTQ5QXdFSEEwSUFCT2t2a1RIdT\
hRbFQzRkhKMVVhSTcrV3NIT2IwVVMzU0FMdEc1d3VLUURqaWV4MDZcL1NjWTVQSmlidm\
dIVEIrRlwvUVRqZ2VsSEd5MVlLcHdjTk1jc1N5YWpSVEJETUJJR0ExVWRFd0VCXC93UU\
lNQVlCQWY4Q0FRRXdEZ1lEVlIwUEFRSFwvQkFREFnSUVNQjBHQTFVZERnUVdCQlRvWk\
lNelFkc0RcL2pcLytnWFwvN2NCSnVjSFwvWG1qQUtCZ2dxaGtqT1BRUURBZ05KQURCR0\
FpRUF0eFEzK0lMR0JQSXRTaDRiOVdYaFhOdWhxU1A2SCtiXC9MQ1wvZlZZRGpRNm9DSV\
FERzJ1UkNIbFZxM3loQjU4VFhNVWJ6SDgrT2xoV1V2T2xSRDNWRXFEZGNRdz09Il0sIn\
R5cCI6InZvdWNoZXItandzK2pzb24iLCJhbGciOiJFUzI1NiJ9",
    "signature": "0fzuqVdyhemWsu_HQeF-CmQwJeLp9IStNf-bWZwz6SojrEOR4a\
Dq6VStyG8eWXjGHNZiRyyLJo7RP1rKatuS2w"
  }]
}
```

Figure 5: Example Parboiled Registrar-Voucher-Request (RVR)

8.3. Example Voucher Response

The following is an example voucher response as JWS Voucher artifact, which would be sent from the MASA to the Pledge via Registrar.

```
{
  "payload": "eyJpZXRmLXZvdWNoZXI6dm91Y2hlciI6eyJhc3NlcnRpb24iOiJsb2\
dnZWQiLCJzZXJpYWwtbnVtYmVyIjoiMDEyMzQ1Njc4OSIsIm5vbmNlIjoiZGRoSGQ4Ml\
FpUGtzMDBTck1USTlEUT09IiwiY3JlYXRlZC1vbiI6IjIwMjItMDctMDdUMTc6NDc6MD\
EuODkwWiIsInBpbm5lZC1kb21haW4tY2VydCI6Ik1JSUJwRENDQVVtZ0F3SUJBZ0lHQV\
cwZUx1SCtNQW9HQ0NxR1NNNDlCQU1DTURVeEV6QVJCZ05WQkFvTUNrMTVRblZ6YVc1bG\
MzTXhEVEFMQmdOVkJBY01CRk5wZEdVeER6QU5CZ05WQkFNTUJsUmxjM1JEUVRBZUZ3MH\
hPVEE1TVRFd01qTTNNekphRncweU9UQTVNVEV3TWpNM016SmFNRFV4RXpBUkJnTlZCQW\
9NQ2sxNVFuVnphVzVsYzNNeERUQUxCZ05WQkFjTUJGTnBkR1V4RHpBTkJnTlZCQU1NQm\
xSbGMzUkRRVEJaTUJNR0J5cUdTTTQ5QWdFR0NDcUdTTTQ5QXdFSEEwSUFCT2t2a1RIdT\
hRbFQzRkhKMVVhSTcrV3NIT2IwVVMzU0FMdEc1d3VLUURqaWV4MDYvU2NZNVBKaWJ2Z0\
hUQitGL1FUamdlbEhHeTFZS3B3Y05NY3NTeWFqUlRCRE1CSUdBMVVkRXdFQi93UUlNQV\
lCQWY4Q0FRRXdEZ1lEVlIwUEFRSC9CQVFEQWdJRU1CMEdBMVVkRGdRV0JCVG9aSU16UW\
RzRC9qLytnWC83Y0JKdWNIL1htakFLQmdncWhrak9QUVFEQWdOSkFEQkdBaUVBdHhRMy\
tJTEdCUEl0U2g0YjlXWGhYTnVocVNQNkgrYi9MQy9mVllEalE2b0NJUURHMnVSQ0hsVn\
EzeWhCNThUWE1VYnpIOCtPbGhXVXZPbFJEM1ZFcURkY1F3PT0ifX0",
  "signatures": [{
    "protected": "eyJ4NWMiOlsiTUlJQmt6Q0NBVGlnQXdJQkFnSUdBV0ZCakNrWU\
1Bb0dDQ3FHU000OUJBTUNNRDB4Q3pBSkJnTlZCQVlUQWtGUk1SVXdFd1lEVlFRS0RBeE\
thVzVuU21sdVowTnZjbkF4RnpBVkJnTlZCQU1NRGtwcGJtZEthVzVuVkdWemRFTkJNQj\
RYRFRFNE1ERXlPVEV3TlRJME1Gb1hEVEk0TURFeU9URXdOVEkwTUZvd1R6RUxNQWtHQT\
FVRUJoTUNRVkV4RlRBVEJnTlZCQW9NREVwcGJtZEthVzVuUTI5eWNERXBNQ2NHQTFVRU\
F3d2dTbWx1WjBwcGJtZERiM0p3SUZadmRXTm9aWElnVTJsbmJtbHVaeUJMWlhrd1dUQV\
RCZ2NxaGtqT1BRSUJCZ2dxaGtqT1BRTUJCd05DQUFTQzZiZUxBbWVxMVZ3NmlRclJzOF\
IwWlcrNGIxR1d5ZG1XczJHQU1GV3diaXRmMm5JWEgzT3FIS1Z1OHMyUnZpQkdOaXZPS0\
dCSEh0QmRpRkVaWnZiN294SXdFREFPQmdOVkhROEJBZjhFQkFNQ0I0QXdDZ1lJS29aSX\
pqMEVBd0lEU1FBd1JnSWhBSTRQWWJ4dHNzSFAyVkh4XC90elVvUVwvU3N5ZEwzMERRSU\
5FdGNOOW1DVFhQQWlFQXZJYjNvK0ZPM0JUbmNMRnNhSlpSQWtkN3pPdXNuXC9cL1pLT2\
FFS2JzVkRpVT0iXSwidHlwIjoidm91Y2hlci1qd3MranNvbiIsImFsZyI6IkVTMjU2In\
0",
    "signature": "y1HLYBFlwouf42XWSKUWjeYQHnG2Q6A4bjA7hvTkB3z1dPwTUl\
jPHtuN2Qex6gDxTfaSiKeoXGsOD4JWOgQJPg"
  }]
}
```


Figure 6: Example Voucher Response

9. References

9.1. Normative References

[BRSKI] Pritikin, M., Richardson, M., Eckert, T., Behringer, M., and K. Watsen, "Bootstrapping Remote Secure Key Infrastructure (BRSKI)", RFC 8995, DOI 10.17487/RFC8995, May 2021, <https://www.rfc-editor.org/rfc/rfc8995>.

[I-D.ietf-anima-rfc8366bis] Watsen, K., Richardson, M., Pritikin, M., Eckert, T. T., and Q. Ma, "A Voucher Artifact for Bootstrapping Protocols", Work in Progress, Internet-Draft, draft-ietf-anima-rfc8366bis-12, 8 July 2024, <https://datatracker.ietf.org/doc/html/draft-ietf-anima-rfc8366bis-12>.

[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/rfc/rfc2119>.

[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/rfc/rfc5280>.

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

[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/rfc/rfc8174>.

[RFC8259] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", STD 90, RFC 8259, DOI 10.17487/RFC8259, December 2017, <https://www.rfc-editor.org/rfc/rfc8259>.

[RFC8995] Pritikin, M., Richardson, M., Eckert, T., Behringer, M., and K. Watsen, "Bootstrapping Remote Secure Key Infrastructure (BRSKI)", RFC 8995, DOI 10.17487/RFC8995, May 2021, <https://www.rfc-editor.org/rfc/rfc8995>.

9.2. Informative References

[I-D.ietf-anima-brski-prm] Fries, S., Werner, T., Lear, E., and M. Richardson, "BRSKI with Pledge in Responder Mode (BRSKI-PRM)", Work in Progress, Internet-Draft, draft-ietf-anima-brski-prm-15, 26 August 2024, <https://datatracker.ietf.org/doc/html/draft-ietf-anima-brski-prm-15>.

[I-D.ietf-anima-constrained-voucher] Richardson, M., Van der Stok, P., Kampanakis, P., and E. Dijk, "Constrained Bootstrapping Remote Secure Key Infrastructure (cBRSKI)", Work in Progress, Internet-Draft, draft-ietf-anima-constrained-voucher-25, 8 July 2024, <https://datatracker.ietf.org/doc/html/draft-ietf-anima-constrained-voucher-25>.

[ON-PATH] "can an on-path attacker drop traffic?", n.d., <https://mailarchive.ietf.org/arch/msg/saag/m1r9uo4xYznOcf85Eyk0Rhut598/>.

[RFC3629] Yergeau, F., "UTF-8, a transformation format of ISO 10646", STD 63, RFC 3629, DOI 10.17487/RFC3629, November 2003, <https://www.rfc-editor.org/rfc/rfc3629>.

[RFC5652] Housley, R., "Cryptographic Message Syntax (CMS)", STD 70, RFC 5652, DOI 10.17487/RFC5652, September 2009, <https://www.rfc-editor.org/rfc/rfc5652>.

[RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <https://www.rfc-editor.org/rfc/rfc7950>.

[RFC7951] Lhotka, L., "JSON Encoding of Data Modeled with YANG", RFC 7951, DOI 10.17487/RFC7951, August 2016, <https://www.rfc-editor.org/rfc/rfc7951>.

[RFC8366] Watsen, K., Richardson, M., Pritikin, M., and T. Eckert, "A Voucher Artifact for Bootstrapping Protocols", RFC 8366, DOI 10.17487/RFC8366, May 2018, <https://www.rfc-editor.org/rfc/rfc8366>.

[RFC8572] Watsen, K., Farrer, I., and M. Abrahamsson, "Secure Zero Touch Provisioning (SZTP)", RFC 8572, DOI 10.17487/RFC8572, April 2019, <https://www.rfc-editor.org/rfc/rfc8572>.

[RFC8792] Watsen, K., Auerswald, E., Farrel, A., and Q. Wu, "Handling Long Lines in Content of Internet-Drafts and RFCs", RFC 8792, DOI 10.17487/RFC8792, June 2020, <https://www.rfc-editor.org/rfc/rfc8792>.

[RFC8812] Jones, M., "CBOR Object Signing and Encryption (COSE) and JSON Object Signing and Encryption (JOSE) Registrations for Web Authentication (WebAuthn) Algorithms", RFC 8812, DOI 10.17487/RFC8812, August 2020, <https://www.rfc-editor.org/rfc/rfc8812>.

Contributors

Toerless Eckert
Futurewei Technologies Inc.
Email: tte+ietf@cs.fau.de

Esko Dijk
Email: esko.dijk@iotconsultancy.nl

Steffen Fries
Siemens AG
Email: steffen.fries@siemens.com

Authors' Addresses

Thomas Werner
Siemens AG
Email: thomas-werner@siemens.com

Michael Richardson
Sandelman Software Works
Email: mcr+ietf@sandelman.ca