Ok



Open Banking Implementation Entity

OBIE, OBE and ETSI Signature Format

Report

Prepared for



©RAiDiAM Services Limited 2020

Information Classification - CONFIDENTIAL

Document Control

|  |  |
| --- | --- |
| Title | Open Banking Implementation Entity OBIE, OBE and ETSI Signature Format |
| Author | Ralph Bragg (Partner), Dr Tony Silveston (Consultant) |

|  |  |
| --- | --- |
| Owner (Responsible for Approval of Issued Versions) | |
| Name | Role |
| Ralph Bragg | Raidiam Director |

|  |
| --- |
| Review Panel |

|  |  |  |
| --- | --- | --- |
| Name | Role | Review Requirement |
| Ralph Bragg | Partner – Peer Review | Mandatory |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Change History | | | |
| Issue | Date | Author/ Editor | Details of Change |
| Version 0.1 | 15th May 2020 | Tony Silveston | Draft for Internal Raidiam Review |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
| References | | | |
| Number | Title | Link | Description |
| 1 | Read/Write Data API Specification v3.12 |  | Open Banking Implementation Entity Read/Write Data APIs.  Includes IETF RFC 7515 JWS signature format specification for OBIE. |
| 2 | PRETA Open Banking Europe: JSON Web Signature Profile for Open Banking |  | Open Banking Europe proposal for common signature format based on both IETF RFC 7515 (JWS) and ETSI TS 119 182-1 (JAdES) |
| 3 | ETSI TS 119 182-1 "Electronic Signatures and Infrastructures (ESI); JAdES digital signatures; Part 1: Building blocks and JAdES baseline signatures" | <https://portal.etsi.org/webapp/WorkProgram/Report_WorkItem.asp?WKI_ID=52897> | ETSI link to draft JAdES specification.  [Requires ETSI account to access] |
| 4 | IETF Signing HTTP Messages | <https://datatracker.ietf.org/doc/draft-ietf-httpbis-message-signatures/> | Signing HTTP messages based on draft-cavage-http-signatures-12 proposed by Oracle and Digital Bazaar. |
| 5 | IETF JSON Web Signature | <https://tools.ietf.org/html/rfc7515> | IETF Standard JSON Web Signature standard text. |

TABLE OF CONTENTS

List of Acronyms 5

1 Executive Summary 6

2 Assessment and Findings 7

2.1 Assessment 7

2.2 Problem / Opportunity statement 7

2.3 Engagement Scope 7

3 Summary of Current API Usage in UK and Europe 8

3.1 Open Banking Europe survey summary 8

3.2 Summary of current API signature formats in UK and Europe 8

3.3 Summary of current OBIE JOSE Message Signing Header 9

4 Summary of OBE PSD2 JWS Profile 11

4.1 OBE Proposal to align API signature formats 11

4.2 OBE PSD2 JWS Profile Conformance Requirement 12

4.3 OBE Design Principals for new PSD2 JWS Profile 14

4.4 JAdES digital signatures 15

5 Impact Assessment on OBIE 16

5.1 Initial Engagement with OBE and ETSI 16

5.2 OBIE Options Summary 16

5.3 OBIE Impact Summary 16

5.4 Future 17

Appendix A 18

JWS Creation Example 18

JWS Validation Example 22

# List of Acronyms

For the purposes of the present document, the abbreviations given following apply:

* API Application Programming Interface
* EBA European Banking Authority
* EIDAS Electronic IDentification, Authentication and trust Services
* ETSI European Telecommunications Standards Institute
* EU European Union
* HTTP HyperText Transfer Protocol
* IETF Internet Engineering Task Force
* JAdES JSON Advanced Electronic Signature
* JSON JavaScript Object Notation
* JWS JSON Web Signature
* OB Open Banking
* OBIE Open Banking Implementation Entity
* OBE Open Banking Europe
* PSD2 Payment Services Directive 2 NOTE: See Directive (EU) 2015/2366 [i.2].
* PSP Payment Service Provider
* RFC Request For Comment
* RFP Request For Proposal
* RTS Regulatory Technical Standard for PSD2 strong customer authentication and common and secure open standards of communication

# Executive Summary

Open Banking Europe recently (Q1 2020) brought together a group of experts from the PSD2 API communities with experts on signature formats from ETSI. The group carried out a survey of the current approaches to secure communications for PSD2 based on EU Qualified Certificates as required under the EU "regulatory technical standards for strong customer authentication and common and secure open standards of communication". As a result of the survey it was found that there were two basic approaches taken by the banking community.

1. About half the API communities used proposed standard IETF JSON Web Signatures to protect API payload **[REF 5]**
2. The other half used HTTP Signatures based on a draft specification originally authored by M Cavage at Oracle and M Sporny at Digital Bazaar **[REF 4]**

As a result, Open Banking Europe have produced a common specification **[REF 2]** of how to protect PSD2 payloads which brings together the JSON Web Signatures with the ability of HTTP Signatures to protect HTTP header information. Open Banking Europe have proposed a new profile named “**OBE PSD2 JWS Profile”** that promises to unite the two approaches used today.

Open Banking Europe have also decided to align their specification with the ETSI "JAdES" specification, currently under development for advanced electronic signatures and seals in line with the EU eIDAS regulation **[REF 3].**

Open Banking Implementation Entity (OBIE) currently supports JSON Web Signatures **[REF 1, 5]** and would like to understand the impact of supporting the new OBE format **[REF 2]** in draft format.

This report summarizes:

* Current approaches to API signatures
* New OBE PSD2 JWS Profile
* Summary and impact on current OBIE

# Assessment and Findings

This section sets out the assessment undertaken and the summary findings thereof.

## Assessment

Raidiam consultants have:

* Attended workshop with Open Banking Europe, PRETA, Berlin Group and various other organisations to discuss API signature formats across Europe and UK.
* Summarized the current API signature formats

## Problem / Opportunity statement

**Problem**

* Open Banking Implementation Entity would like to understand the impact of supporting the common API signature format proposed by Open Banking Europe

**Opportunity**

* A. The OBIE remains compliant with European ETSI requirements for signatures such that they will be accepted as technically be accepted as advanced electronic signatures.
* B. The OBIE remains compatible with European ETSI requirements for signatures and adopts the consistent European standard signing patterns, the “OBE PSD2 JWS Profile”.

## Engagement Scope

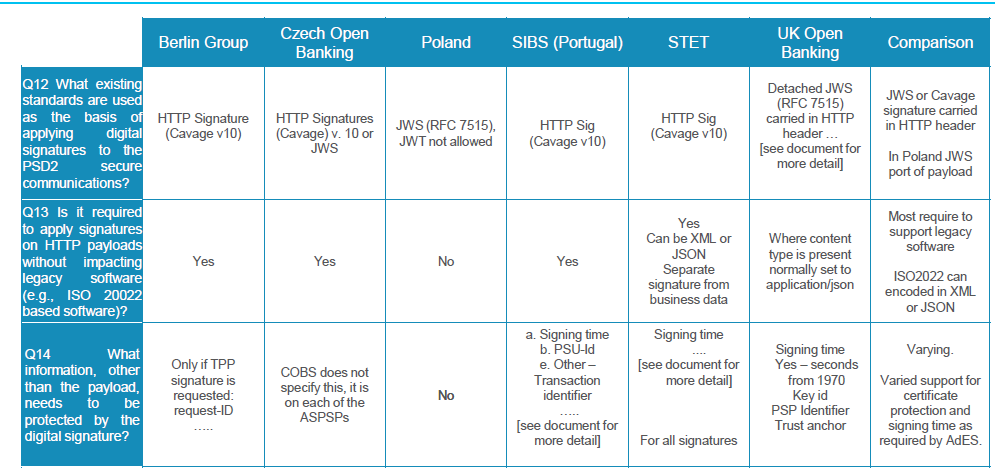
### In scope

* + To summarize current OBIE JSON web signature format
  + To summarize the new proposed OBE PSD2 Profile and JAdES signature format
  + To summarize options for OBIE

# Summary of Current API Usage in UK and Europe

This section summarizes the findings of Open Banking Europe in summarizing current formats used by various organisations in UK and Europe for API signatures for PSD2 exchanges.

## Open Banking Europe survey summary

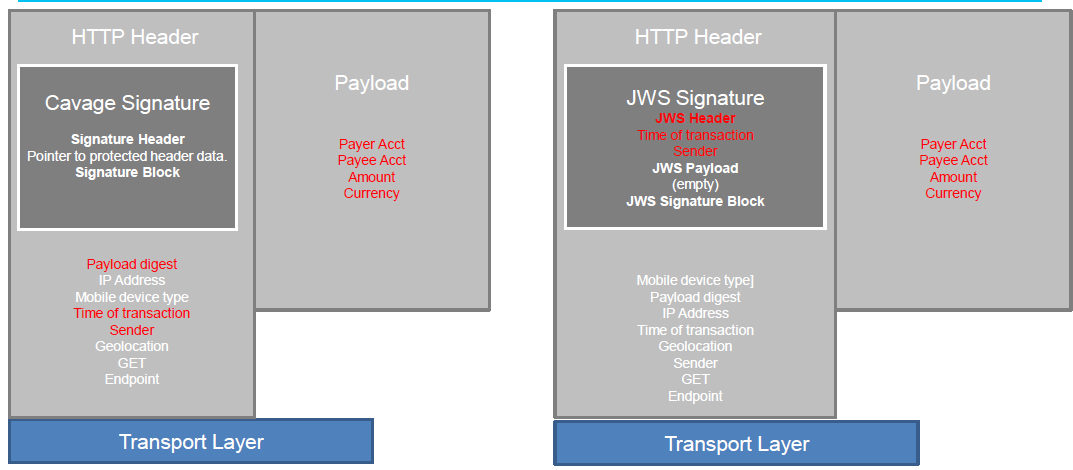


The table above summarizes a number of organisations usage of either JWS or HTTP Signatures across their API interfaces.

Inspection of the above table demonstrates the survey findings of two technologies being widely used for API signatures across UK and Europe.

The table above also highlights the fact different organisations expect different data items in the API call to be protected by a digital signature. For instance, OBIE expects signing time, keyed, PSP Identifier and the Trust Anchor to be protected as defined in the Read/Write Specification **[REF 3].**

## Summary of current API signature formats in UK and Europe



The above diagram summarizes what data is protected by a digital signature using the HTTP Signature or JWS Signature format.

The above diagram indicates there is variation on what is digitally signed (and therefore can be digitally validated) with each format across organisations.

## Summary of current OBIE JOSE Message Signing Header

The APIs require TLS 1.2 Mutual Authentication, and this may be used as a means of non-repudiation. However, it would be difficult to maintain digital records and evidence of non-repudiation if the API only relied on TLS 1.2.

A solution for non-repudiation that does not rely on TLS, would be achieved by providing a JWS with detached content (as defined in [RFC 7515](https://tools.ietf.org/html/rfc7515)) in the HTTP header of each API request.

The JOSE header for the signature must contain the following claims **[REF 1]:**

| **Claim** | **Description** |
| --- | --- |
| alg | The algorithm that will be used for signing the JWS.  The list of valid algorithms is here <https://tools.ietf.org/html/rfc7518#section-3.1>.  This value must be PS256 |
| typ | This is an optional claim.  If it is specified, it must be set to the value "JOSE" |
| cty | This is an optional claim for JSON payloads  If it is specified, it must be set to the value "json" or "application/json" for json payloads.  For non-json payloads (e.g. for PDF files), the mime type of the payload must be specified in this claim. |
| kid | This is a mandatory claim.  It must match a value that can be used to look up the key in a key store hosted by the Trust Anchor. |
| b64 | This **must** have the boolean value false.  This indicates that the message payload is not base64 url encoded.  (See [RFC 7797 - The "b64" header Parameter](https://tools.ietf.org/html/rfc7797#section-3)) |
| http://openbanking.org.uk/iat | This **must** be a JSON number representing the number of seconds from 1970-01-01T0:0:0Z as measured in GMT until the date/time.  This is a private header parameter name. (See [RFC 7515 - Private Header Parameter Names](https://tools.ietf.org/html/rfc7515#section-4.3)) |
| http://openbanking.org.uk/iss | This **must** be a string that identifies the PSP.  If the issuer is using a certificate this value **must** match the**subject**of the signing certificate**.**  If the issuer is using a signing key lodged with a Trust Anchor, the value is defined by the Trust Anchor and should uniquely identify the PSP.  For example, when using the Open Banking Directory, the value must be:   * When issued by a TPP, of the form {{orgi-id}}/{{software-statement-id}}, * When issued by an ASPSP of the form {{org-id}}   Where:   * org-id is the open-banking issued organization id   + software-statement-id is the open-banking issued software-statement-id |
| http://openbanking.org.uk/tan | This **must** be a string that consists of a domain name that is registered to and identifies the Trust Anchor that hosts the public counter-part of the key used for signing.  For example, when using the Open Banking Directory, the value must be openbanking.org.uk |
| crit | This **must** be a string array consisting of the values "b64", "http://openbanking.org.uk/iat", "http://openbanking.org.uk/iss", "http://openbanking.org.uk/tan"  This indicates that the JWS signature validator must understand and process the three additional claims. |

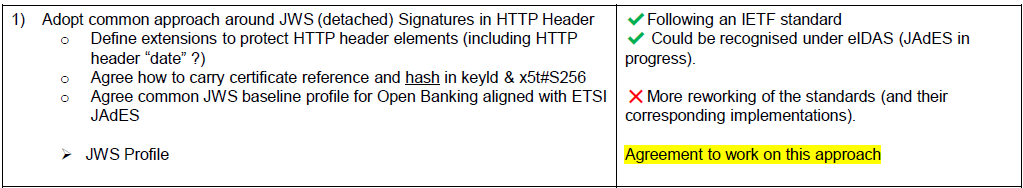
The signer must compute the signature as a detached JWS as defined in [RFC 7515](https://tools.ietf.org/html/rfc7515)[, Appendix F](https://tools.ietf.org/html/rfc7515#appendix-F).

The signer **must** include an HTTP header called **x-jws-signature** with its value set to the signature computed.

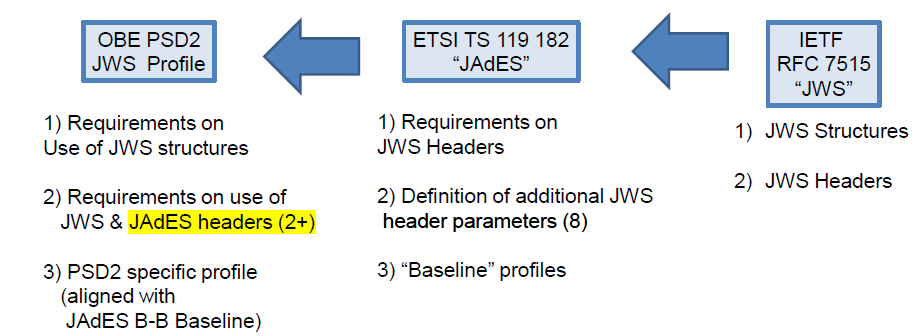
# Summary of OBE PSD2 JWS Profile

OBE have proposed a new specification **[REF 2]** to align the two widely used API signature formats as discussed below.

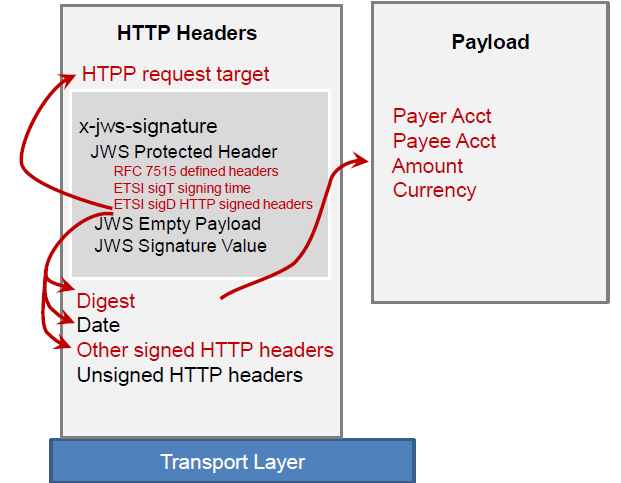
## OBE Proposal to align API signature formats



The table above shows Open Banking Europe proposal for a common specification **[REF 2]** based on aligning with ETSI JAdES **[REF 3]** to incorporate both the JWS and HTTP Signature approaches.



The above diagram depicts the pathway to creating a new specific profile for PSD2 “**OBE PSD2 JWS Profile**” that incorporates both existing JWS and HTTP Signature formats.



The above diagram depicts the format for the new “**OBE PSD2 JWS Profile”**

The new profile is intended to be a common specification of how to protect PSD2 API payloads. The profile brings together JSON Web Signatures with the ability of HTTP Signatures to protect HTTP header information.

The new profile also aligns with the specification with ETSI "JAdES" currently under development for advanced electronic signatures and seals in line with the EU eIDAS regulation. The current OBE profile is aligned with the basic (B-B) level of JAdES and makes use of JWS extensions defined in JAdES.

## OBE PSD2 JWS Profile Conformance Requirement

Key:

**M = Mandatory**

**R = Recommended**

**C = Conditional**

**O = Optional**

**NP = Mandatory Not present**

OBE welcomes suggestions from API providers on further constraints on the below field requirements for their new proposed profile.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Description and notes** | **Signer to include** | **Relying Party to process if present** | **Section ref "-" requirement / recommendation / option number** |
| alg | Indicates the cryptographic algorithm used for a signature  It is recommended to use algorithms that are listed in both RFC 7518 and ETSI TS 119 312 | M | M | 5.2 - 6 |
| x5c | X.509 public key certificate or certificate chain corresponding to the key used to create the JSON Web Signature. | C | M | 5.3 - 7, 8, 9 |
| x5t | Digest of the X.509 signing certificate using SHA1. Not required. | NP | - | 5.3 – 12 |
| x5t#S256 | Digest of the X.509 signing certificate using SHA 256. | C | C | 5.3 - 7, 10,11 |
| x5t#o | Alternative hashing algorithms for the digest of the X.509 signing certificate  [JAdES] | C | C | 5.3 – 7, 10,11  6.4 – 25 |
| kid | Identifier of the public key for validating the signature | O | O | 5.3 – 13 |
| x5u | URI pointing to the resource where the X.509 signing certificate (with or without the certification path) may be retrieved from | O | O | 5.3 – 14 |
| typ | Used to identify that the general structure of header.  Set to fixed string: JOSE | R | R | 5.4 – 15 |
| cty | Content Type. Not Required.  This requirement is better met through use of the *HTTP Header* Parameter Content-Type | NP | - | 5.5 – 16 |
| crit | Header parameters can be either critical or non-critical.  Critical Headers are:  b64  sigT  sigD | M | M | 5.6 – 17 |
| jwk | The public key for validating the signature. Not required.  The use of this attribute is not considered appropriate to the use of PSD2 which requires the use of qualified certificates | NP | - | 5.7 – 18 |
| jku | A URI pointing to the resource where the public key may be retrieved. Not Required.  The use of this attribute is not considered appropriate to the use of PSD2 which requires the use of qualified certificates | NP | - | 5.7 – 18 |
| sigT | The JAdES "sigT" header parameter contains the claimed signing time encoded using RFC 3339 Internet time format for UTC without fractional seconds (e.g. "2019-11-19T17:28:15Z"). | M | R | 6.2 - 19, 20 |
| sigD | The JAdES "sigD" header parameter contains:  mID: A URI which identifies the mechanism used to identify the Data to be Signed.  pars: Parameters of this mechanism  The JAdES "sigD" header parameter shall be present with "mID" set to "http://uri.etsi.org/19182/HttpHeaders".  pars" shall include the following HTTP Header field name:  "Digest" as defined in RFC 3230[26] applied to the HTTP body. If the HTTP Body is not present, the "Digest" header shall contain the hash of an empty bytelist  The JAdES "sigD" "pars" should include the following HTTP Header field names:   * "(request-target)" for HTTP Requests * "Content-Type" if present * "Content-Encoding" if present | M | M | 6.3 - 21, 22, 23, 24 |

The above table summarizes the OBE conformance requirements to align with their new PSD2 JWS profile **[REF 2]**

## OBE Design Principals for new PSD2 JWS Profile

The profile defined **[REF 2]** is designed to meet the following requirements:

DESIGN-PRINCIPLE#1: The signatures compliant with this profile support the use of qualified certificates for electronic seals in line with Commission Delegated Regulation (EU) 2018/389 [15].

DESIGN-PRINCIPLE#2: The profile is aligned with JAdES baseline digital signatures as specified by ETSI.

DESIGN-PRINCIPLE#3: The signature protects a *HTTP body* and optionally selected *HTTP Header* fields.

DESIGN-PRINCIPLE#4: A single signature is to be carried in an *HTTP Header* detached from the payload.

DESIGN-PRINCIPLE#5: The signature is as transparent as possible to any intermediate device that they may traverse when they are exchanged between parties (firewalls, front-ends, relays, etc).

DESIGN-PRINCIPLE#6: The profile aims to maximise interoperability.

DESIGN-PRINCIPLE#7: No restrictions are imposed to the contents of the signed payloads. It can be used to protect JSON, XML ISO 20022 or any other form of data.

DESIGN-PRINCIPLE#8: *JSON Web Signature* headers and *HTTP Header* fields which are critical to the security of the exchange, as well as *HTTP body*, are protected such that they cannot be modified.

DESIGN-PRINCIPLE#9: This profile follows generally accepted security best practices.

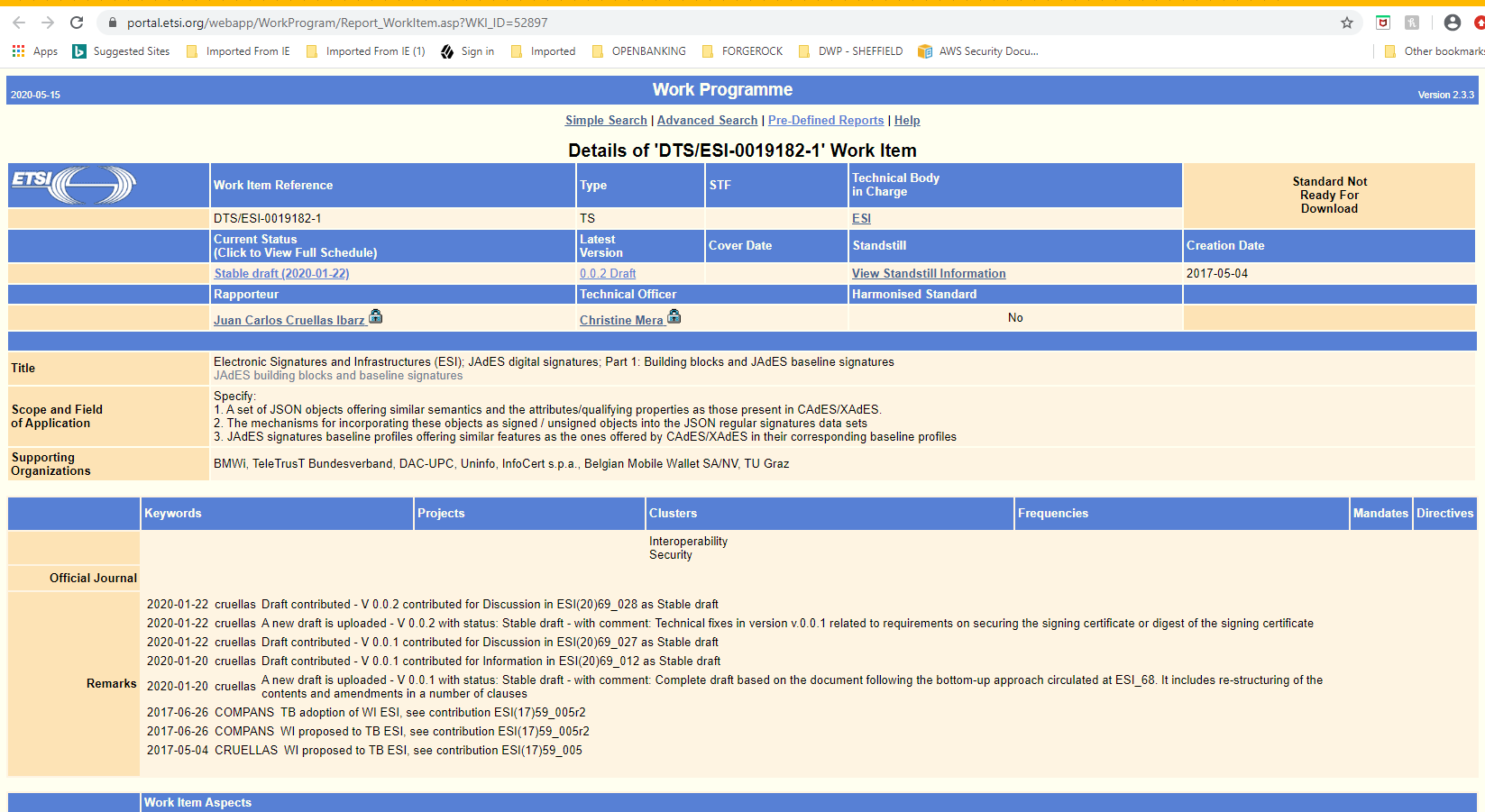
DESIGN-PRINCIPLE#10: The profile defined is extensible.

DESIGN-PRINCIPLE#11: Signatures can be later used as evidence in court (i.e. are "non-repudiable").

DESIGN-PRINCIPLE#12: Selected HTTP header parameters can be signed without there being an *HTTP body* (e.g. for GET or DELETE requests)

DESIGN-PRINCIPLE#13: Signatures can be applied to HTTP requests as well as responses.

## JAdES digital signatures



The ETSI JAdES stable draft is at version 0.0.0.2 from Q1 2020. It is marked as not ready for download on the ETSI site and this report does not contain a version embedded in the reference section. However, the above ETSI webpage URL is located at **[REF 3].**

It is somewhat difficult to assess the JAdES draft specification any further without a more mature specification available for download and review.

# Impact Assessment on OBIE

## Initial Engagement with OBE and ETSI

The OBE PSD2 JWS Profile specification **[REF 2]** does not particularly call out well what is OBE versus what is ETSI JAdES. The very latest draft of the ETSI JAdES specification was not included in Open Banking Europe initial engagement with Raidiam, so this report is relying on OBEs own review of the underlying ETSI JAdES specification.

The ETSI JAdES draft specification is located at **[REF 3]** but this requires an ETSI account and is marked as not available for download on the ETSI site.

## OBIE Options Summary

**OBIE + ETSI JAdES Headers**

If the OBIE’s objective is to adopt enough of the requirements of JAdES message signing to have the signatures be accepted across Europe then the introduction of only the mandatory sigT and sigD fields should be sufficient.

These header fields could be layered on the existing OBIE JWS Read/Write API format with a minimum of change either to the trust framework or for participants.

**OBIE Migration to OBE**

If the OBIE wanted to align across Europe with the OBE signature format that will be adopted by the other European standards bodies then the changes will be more painful for the ecosystem to swallow. Headers that are mandatory for OBIE are repurposed in a different way by OBE and the mandatory requirement for CTY header by OBIE is banned by OBE in favour of the header being protected by the sigD header protection process.

**Duplicate use of x-jws-signature:**

Whilst it is frustrating that standards bodies are going to use x-jws-signature header to convey the signature, the use and presence of OBIE critical properties in the crit header is enough to indicate that OBIE processing of the signature should be applied.

## OBIE Impact Summary

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Property** | **OBIE** | **OBE** | **ETSI** | **OBIE Value** | **OBE/ETSI** | **Notes** |
| Kid | MUST | MAY |  | Equivalent. To x5t256 | Must be the IssuerSerial | This change of value for the kid is a breaking change for the ecosystem. Currently the obie aligns the kid with x5t256 for eIDAS certs |
| x5t256 | MAY | MAY |  |  |  | Migration: Introduce x5t256 as a mandatory claim and then retire the use of kid |
| Cty | MUST | MUST NOT |  |  |  | OBE requires that the MIME content type be provided as part of the request header |
| openbanking.org.uk/iat | MUST |  |  | RFC3339 |  | Two different formats time are used note that the sigT is the ETSI requirement |
| sigT |  | MUST | MUST |  | RFC 7519 |  |
| sigD |  |  | MUST |  |  | The OBIE specification doesn’t define the headers that need to be included; ETSI has a set of mandatory headers and this must be included. Given that the OBIE specification doesn't include the x-idempotency header in the signature at the moment then technically it could be argued that it doesn't meet the requirement for non-repudiation as application level message reply attacks are still possible. |
| sigD - mID |  |  | MUST |  |  | Digest property is required to indicate that it's a message digest. Should this field be empty in the case of a DELETE then it should be the hash of an empty byte list. |
| sigD –  miD - Sub Headers | |  | SHOULD |  |  |  |

## Future

During the review, it has been noted that the OBIE does not include any header protection in the signature signing process, the absence of the inclusion of the idempotency header in the message signature therefore means that the signature process doesn’t meet the requirement for non-repudiation as the same payload body could be replayed with a different idempotency key from an attacker and thus processed by a recipient. In practical terms that this has very little affect and ability to exploit would be virtually zero however it still would allow a participant to argue that didn’t send a duplicate or a repeated message as the unique property of the message isn’t signed. The ETSI signature process has the ability to protect this and other headers.

Raidiam are still waiting for comments back from Nick Pope at Open Banking Europe regarding inconsistencies in the proposed OBE specification including recommendation 21 and 24 which are mutually exclusive. Both the ETSI draft and OBE signing document are likely to be revised when another meeting is scheduled in Q2 2020.

# Appendix A

OBE have provided a worked example of how their new PSD2 profile may work in practice and this is reproduced below.

This appendix contains examples of creation and validation of a JSON Web Signature following the OBE JSON Web Signature Profile for Open Banking.

Disclaimer – These examples should not be taken as reference point for implementations of the OBE JWS profile. They are based on a single implementation and have not been tested for correctness against other implementations.

In addition, the keys and certificates used in this example is only issued by a test rig. The test certificate is not intended to be conformant to eIDAS or PSD2 and may have expired. The order of the steps described are only those take by the demonstration implementation and other ordering may be used if they produce the same result.

JWS Creation Example

Step 0: Input HTTP message

|  |
| --- |
| **Description**: Prepare unsigned HTTP message input to JWS function |
| POST https://api.testbank.com/v1/payments/sepa-credit-transfers HTTP/1.1  Content-Type: application/json  X-Request-ID: 99391c7e-ad88-49ec-a2ad-99ddcb1f7721  PSU-IP-Address: 192.168.8.78  PSU-GEO-Location: GEO:52.506931,13.144558  PSU-User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:54.0) Gecko/20100101 Firefox/54.0  Date: Fri, 3 Apr 2020 16:38:37 GMT  {  "instructedAmount": {"currency": "EUR", "amount": "123.50"},  "debtorAccount": {"iban": "DE40100100103307118608"},  "creditorName": "Merchant123",  "creditorAccount": {"iban": "DE02100100109307118603"},  "remittanceInformationUnstructured": "Ref Number Merchant"  } |

Step 1: JWS Protected Header

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Description:** Produce JWS header parameters which define how the signature is created:   |  |  | | --- | --- | | "b64":" false | Means don't base64url encode header data to be signed | | "x5t#S256": "…." | Hash of signing certificate | | "crit":["sigT","sigD","b64"] | non-standard (ie. not defined in RFC 7515) JWS header parameters which are critical | | "sigT":"…." | Claimed siging time | | "sigD":{….} | HTTP Header fields to be signed | | "alg": "RS256" | Signature algorithm |   **Note:** Thisis shown below in a pretty print layout. It will be sent as a single string without line brakes or extra spaces. Escape characters are not used. |
| {  "b64": "false",  "x5t#S256": "dytPpSkJYzhTdPXSWP7jhXgG4kCOWIWGiesdzkvNLzY=",  "crit": [  "sigT",  "sigD",  "b64"  ],  "sigT": "2020-04-16T07:35:56Z",  "sigD": {  "pars": [  "(request-target)",  "Content-Type",  "PSU-IP-Address",  "PSU-GEO-Location",  "Digest"  ],  "mId": "http: //uri.etsi.org/19182/HttpHeaders"  },  "alg": "RS256"  } |

Step 2: Base64url JWS Protected Header

|  |
| --- |
| **Description:** Covert JWS Protected Header (without line breaks or extra spaces) into a Base64url encoded string. |
| eyJiNjQiOiJmYWxzZSIsIng1dCNTMjU2IjoiZHl0UHBTa0pZemhUZFBYU1dQ  N2poWGdHNGtDT1dJV0dpZXNkemt2Tkx6WT0iLCJjcml0IjpbInNpZ1QiLCJz  aWdEIiwiYjY0Il0sInNpZ1QiOiIyMDIwLTA0LTE2VDA3OjM1OjU2WiIsInNp  Z0QiOnsicGFycyI6WyIocmVxdWVzdC10YXJnZXQpIiwiQ29udGVudC1UeXBl  IiwiUFNVLUlQLUFkZHJlc3MiLCJQU1UtR0VPLUxvY2F0aW9uIiwiRGlnZXN0  Il0sIm1JZCI6Imh0dHA6Ly91cmkuZXRzaS5vcmcvMTkxODIvSHR0cEhlYWRl  cnMifSwiYWxnIjoiUlMyNTYifQ |

Step 3a: HTTP Header to be signed

|  |
| --- |
| **Description:** Create HTTP header string, as selected using the JWS header parameter sigD, ready to be signed (excluding digest). |
| (request-target): post https://api.testbank.com/v1/payments/sepa-credit-transfers  content-type: application/json  psu-ip-address: 192.168.8.78  psu-geo-location: GEO:52.506931,13.144558 |

Step 3b: Digest of HTTP Body

|  |
| --- |
| **Description**: Add a hash digest of the HTTP Body (payload without HTTP header and following empty line). |
| (request-target): post https://api.testbank.com/v1/payments/sepa-credit-transfers  content-type: application/json  psu-ip-address: 192.168.8.78  psu-geo-location: GEO:52.506931,13.144558  digest: SHA-256=+xeh7JAayYPh8K13UnQCBBcniZzsyat+KDiuy8aZYdI= |

Step 4: Prepare input for Signature Value Computation

|  |
| --- |
| **Description:** Combine Base64url encoded JWS Protected Header with HTTP Header to be signed, separated by ".", ready for computation of signature value. |
| eyJiNjQiOiJmYWxzZSIsIng1dCNTMjU2IjoiZHl0UHBTa0pZemhUZFBYU1dQ  N2poWGdHNGtDT1dJV0dpZXNkemt2Tkx6WT0iLCJjcml0IjpbInNpZ1QiLCJz  aWdEIiwiYjY0Il0sInNpZ1QiOiIyMDIwLTA0LTE2VDA3OjM1OjU2WiIsInNp  Z0QiOnsicGFycyI6WyIocmVxdWVzdC10YXJnZXQpIiwiQ29udGVudC1UeXBl  IiwiUFNVLUlQLUFkZHJlc3MiLCJQU1UtR0VPLUxvY2F0aW9uIiwiRGlnZXN0  Il0sIm1JZCI6Imh0dHA6Ly91cmkuZXRzaS5vcmcvMTkxODIvSHR0cEhlYWRl  cnMifSwiYWxnIjoiUlMyNTYifQ.(request-target): post https://api.testbank.com/v1/payments/sepa-credit-transfers  content-type: application/json  psu-ip-address: 192.168.8.78  psu-geo-location: GEO:52.506931,13.144558  digest: SHA-256=+xeh7JAayYPh8K13UnQCBBcniZzsyat+KDiuy8aZYdI= |

Step 5: Compute JWS Signature Value

|  |
| --- |
| **Description:** Compute the digital signature cryptographic value calculated over a sequence of octets derived from the JWS Protected Header and HTTP Header Data to be Signed. This is created using the signing key associated with the certificate identified in the JWS Protected Header "x5t#S256" and using the signature algorithm identified by "alg". |
| MNnQk2xmc3XqWqeAQ4UOFJifCSZV2pKMPGZJzP\_DIm-T0X8zAY1J-x4G2805  \_4ZO8QvNh\_MjOi2l3HIJHT8-wIbuTRIZpQVORt-rjHSpRnEbt9WwtW8WUcIo  DbIBMf1ykAa04uxH3T4aFOyzHcxHU8SyNxwjjQAcDe-k-0IR2ETMrRb-KXbz  UhgDOseoFDD3qULZXSuwEnPvR09ZZMFAdFHtNJL\_ZdsJf8VjMmzZ1dO9DLWy  fdHnlaj7CMDAU9TSl6bqMIe6864EJKWLHQ69EeefgOI2a7obOHGdG22TCrVx  iR2I3C-D2oDCM5p4PNk38LHISFd3bLMaKJMArgmurA |

Step 6: Form JSON Web Signature

|  |
| --- |
| **Description:** Create JSON Web Signature containing the Base64url encoded and ".." and the JWS Signature Value. This is encoded using JWS compact serialisation with the HTTP Header Data to be Signed detached from the signature. |
| eyJiNjQiOiJmYWxzZSIsIng1dCNTMjU2IjoiZHl0UHBTa0pZemhUZFBYU1dQ  N2poWGdHNGtDT1dJV0dpZXNkemt2Tkx6WT0iLCJjcml0IjpbInNpZ1QiLCJz  aWdEIiwiYjY0Il0sInNpZ1QiOiIyMDIwLTA0LTE2VDA3OjM1OjU2WiIsInNp  Z0QiOnsicGFycyI6WyIocmVxdWVzdC10YXJnZXQpIiwiQ29udGVudC1UeXBl  IiwiUFNVLUlQLUFkZHJlc3MiLCJQU1UtR0VPLUxvY2F0aW9uIiwiRGlnZXN0  Il0sIm1JZCI6Imh0dHA6Ly91cmkuZXRzaS5vcmcvMTkxODIvSHR0cEhlYWRl  cnMifSwiYWxnIjoiUlMyNTYifQ..MNnQk2xmc3XqWqeAQ4UOFJifCSZV2pKM  PGZJzP\_DIm-T0X8zAY1J-x4G2805\_4ZO8QvNh\_MjOi2l3HIJHT8-wIbuTRIZ  pQVORt-rjHSpRnEbt9WwtW8WUcIoDbIBMf1ykAa04uxH3T4aFOyzHcxHU8Sy  NxwjjQAcDe-k-0IR2ETMrRb-KXbzUhgDOseoFDD3qULZXSuwEnPvR09ZZMFA  dFHtNJL\_ZdsJf8VjMmzZ1dO9DLWyfdHnlaj7CMDAU9TSl6bqMIe6864EJKWL  HQ69EeefgOI2a7obOHGdG22TCrVxiR2I3C-D2oDCM5p4PNk38LHISFd3bLMa  KJMArgmurA |

Step 7: Resulting HTTP Signed Message

|  |
| --- |
| **Description:** The HTTP message as sent over the network with the JSON Web Signature inserted. |
| POST https://api.testbank.com/v1/payments/sepa-credit-transfers HTTP/1.1  Content-Type: application/json  X-Request-ID: 99391c7e-ad88-49ec-a2ad-99ddcb1f7721  PSU-IP-Address: 192.168.8.78  PSU-GEO-Location: GEO:52.506931,13.144558  PSU-User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:54.0) Gecko/20100101 Firefox/54.0  Date: Fri, 3 Apr 2020 16:38:37 GMT  Digest: SHA-256=+xeh7JAayYPh8K13UnQCBBcniZzsyat+KDiuy8aZYdI=  x-jws-signature: eyJiNjQiOiJmYWxzZSIsIng1dCNTMjU2IjoiZHl0UHBTa0pZemhUZFBYU1dQN2poWGdHNGtDT1dJV0dpZXNkemt2Tkx6WT0iLCJjcml0IjpbInNpZ1QiLCJzaWdEIiwiYjY0Il0sInNpZ1QiOiIyMDIwLTA0LTE2VDA3OjM1OjU2WiIsInNpZ0QiOnsicGFycyI6WyIocmVxdWVzdC10YXJnZXQpIiwiQ29udGVudC1UeXBlIiwiUFNVLUlQLUFkZHJlc3MiLCJQU1UtR0VPLUxvY2F0aW9uIiwiRGlnZXN0Il0sIm1JZCI6Imh0dHA6Ly91cmkuZXRzaS5vcmcvMTkxODIvSHR0cEhlYWRlcnMifSwiYWxnIjoiUlMyNTYifQ..MNnQk2xmc3XqWqeAQ4UOFJifCSZV2pKMPGZJzP\_DIm-T0X8zAY1J-x4G2805\_4ZO8QvNh\_MjOi2l3HIJHT8-wIbuTRIZpQVORt-rjHSpRnEbt9WwtW8WUcIoDbIBMf1ykAa04uxH3T4aFOyzHcxHU8SyNxwjjQAcDe-k-0IR2ETMrRb-KXbzUhgDOseoFDD3qULZXSuwEnPvR09ZZMFAdFHtNJL\_ZdsJf8VjMmzZ1dO9DLWyfdHnlaj7CMDAU9TSl6bqMIe6864EJKWLHQ69EeefgOI2a7obOHGdG22TCrVxiR2I3C-D2oDCM5p4PNk38LHISFd3bLMaKJMArgmurA  {  "instructedAmount": {"currency": "EUR", "amount": "123.50"},  "debtorAccount": {"iban": "DE40100100103307118608"},  "creditorName": "Merchant123",  "creditorAccount": {"iban": "DE02100100109307118603"},  "remittanceInformationUnstructured": "Ref Number Merchant"  } |

JWS Validation Example

Step 1: Input HTTP Signed Message

|  |
| --- |
| **Description:** The signed HTTP message as received over the network. |
| POST https://api.testbank.com/v1/payments/sepa-credit-transfers HTTP/1.1  Content-Type: application/json  X-Request-ID: 99391c7e-ad88-49ec-a2ad-99ddcb1f7721  PSU-IP-Address: 192.168.8.78  PSU-GEO-Location: GEO:52.506931,13.144558  PSU-User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:54.0) Gecko/20100101 Firefox/54.0  Date: Fri, 3 Apr 2020 16:38:37 GMT  Digest: SHA-256=+xeh7JAayYPh8K13UnQCBBcniZzsyat+KDiuy8aZYdI=  x-jws-signature: eyJiNjQiOiJmYWxzZSIsIng1dCNTMjU2IjoiZHl0UHBTa0pZemhUZFBYU1dQN2poWGdHNGtDT1dJV0dpZXNkemt2Tkx6WT0iLCJjcml0IjpbInNpZ1QiLCJzaWdEIiwiYjY0Il0sInNpZ1QiOiIyMDIwLTA0LTE2VDA3OjM1OjU2WiIsInNpZ0QiOnsicGFycyI6WyIocmVxdWVzdC10YXJnZXQpIiwiQ29udGVudC1UeXBlIiwiUFNVLUlQLUFkZHJlc3MiLCJQU1UtR0VPLUxvY2F0aW9uIiwiRGlnZXN0Il0sIm1JZCI6Imh0dHA6Ly91cmkuZXRzaS5vcmcvMTkxODIvSHR0cEhlYWRlcnMifSwiYWxnIjoiUlMyNTYifQ..MNnQk2xmc3XqWqeAQ4UOFJifCSZV2pKMPGZJzP\_DIm-T0X8zAY1J-x4G2805\_4ZO8QvNh\_MjOi2l3HIJHT8-wIbuTRIZpQVORt-rjHSpRnEbt9WwtW8WUcIoDbIBMf1ykAa04uxH3T4aFOyzHcxHU8SyNxwjjQAcDe-k-0IR2ETMrRb-KXbzUhgDOseoFDD3qULZXSuwEnPvR09ZZMFAdFHtNJL\_ZdsJf8VjMmzZ1dO9DLWyfdHnlaj7CMDAU9TSl6bqMIe6864EJKWLHQ69EeefgOI2a7obOHGdG22TCrVxiR2I3C-D2oDCM5p4PNk38LHISFd3bLMaKJMArgmurA  {  "instructedAmount": {"currency": "EUR", "amount": "123.50"},  "debtorAccount": {"iban": "DE40100100103307118608"},  "creditorName": "Merchant123",  "creditorAccount": {"iban": "DE02100100109307118603"},  "remittanceInformationUnstructured": "Ref Number Merchant"  } |

Step 2: Extract JWS Protected header

|  |
| --- |
| **Description:** Extract the Base64url JWS Protected Header from the first part (up to"..") of the JSON Web Signature which includes identification of HTTP header fields to be signed. |
| eyJiNjQiOiJmYWxzZSIsIng1dCNTMjU2IjoiZHl0UHBTa0pZemhUZFBYU1dQ  N2poWGdHNGtDT1dJV0dpZXNkemt2Tkx6WT0iLCJjcml0IjpbInNpZ1QiLCJz  aWdEIiwiYjY0Il0sInNpZ1QiOiIyMDIwLTA0LTE2VDA3OjM1OjU2WiIsInNp  Z0QiOnsicGFycyI6WyIocmVxdWVzdC10YXJnZXQpIiwiQ29udGVudC1UeXBl  IiwiUFNVLUlQLUFkZHJlc3MiLCJQU1UtR0VPLUxvY2F0aW9uIiwiRGlnZXN0  Il0sIm1JZCI6Imh0dHA6Ly91cmkuZXRzaS5vcmcvMTkxODIvSHR0cEhlYWRl  cnMifSwiYWxnIjoiUlMyNTYifQ |

Step 3: Decode JWS Protected Header

|  |
| --- |
| **Description:** Decoded JWS Protected header.  Note: This is converted to pretty print |
| {  "b64": "false",  "x5t#S256": "dytPpSkJYzhTdPXSWP7jhXgG4kCOWIWGiesdzkvNLzY=",  "crit": [  "sigT",  "sigD",  "b64"  ],  "sigT": "2020-04-16T07:35:56Z",  "sigD": {  "pars": [  "(request-target)",  "Content-Type",  "PSU-IP-Address",  "PSU-GEO-Location",  "Digest"  ],  "mId": "http: //uri.etsi.org/19182/HttpHeaders"  },  "alg": "RS256"  } |

Step 4: Recreate HTTP Header to be signed

|  |
| --- |
| **Description:** Recreate HTTP Header for the fields identified in the JWS header sigD. |
| (request-target): post https://api.testbank.com/v1/payments/sepa-credit-transfers  content-type: application/json  psu-ip-address: 192.168.8.78  psu-geo-location: GEO:52.506931,13.144558  digest: SHA-256=+xeh7JAayYPh8K13UnQCBBcniZzsyat+KDiuy8aZYdI= |

Step 5: Recreate input for Signature Value Validation

|  |
| --- |
| **Description:** Combine Base64url encoded JWS Protected Header with HTTP Header to be signed ready for validation of signature value |
| eyJiNjQiOiJmYWxzZSIsIng1dCNTMjU2IjoiZHl0UHBTa0pZemhUZFBYU1dQ  N2poWGdHNGtDT1dJV0dpZXNkemt2Tkx6WT0iLCJjcml0IjpbInNpZ1QiLCJz  aWdEIiwiYjY0Il0sInNpZ1QiOiIyMDIwLTA0LTE2VDA3OjM1OjU2WiIsInNp  Z0QiOnsicGFycyI6WyIocmVxdWVzdC10YXJnZXQpIiwiQ29udGVudC1UeXBl  IiwiUFNVLUlQLUFkZHJlc3MiLCJQU1UtR0VPLUxvY2F0aW9uIiwiRGlnZXN0  Il0sIm1JZCI6Imh0dHA6Ly91cmkuZXRzaS5vcmcvMTkxODIvSHR0cEhlYWRl  cnMifSwiYWxnIjoiUlMyNTYifQ.(request-target): post https://api.testbank.com/v1/payments/sepa-credit-transfers  content-type: application/json  psu-ip-address: 192.168.8.78  psu-geo-location: GEO:52.506931,13.144558  digest: SHA-256=+xeh7JAayYPh8K13UnQCBBcniZzsyat+KDiuy8aZYdI= |

Step 6: Validate signature

|  |
| --- |
| **Description:** ExtractJWS signature value from x-jws-signature after "'.." and validated against recreated JWS protected Header and HTTP Header with the certificate identified in the JWS Protected Header "x5t#S256" and using the signature algorithm identified by "alg". |
| MNnQk2xmc3XqWqeAQ4UOFJifCSZV2pKMPGZJzP\_DIm-T0X8zAY1J-x4G2805  \_4ZO8QvNh\_MjOi2l3HIJHT8-wIbuTRIZpQVORt-rjHSpRnEbt9WwtW8WUcIo  DbIBMf1ykAa04uxH3T4aFOyzHcxHU8SyNxwjjQAcDe-k-0IR2ETMrRb-KXbz  UhgDOseoFDD3qULZXSuwEnPvR09ZZMFAdFHtNJL\_ZdsJf8VjMmzZ1dO9DLWy  fdHnlaj7CMDAU9TSl6bqMIe6864EJKWLHQ69EeefgOI2a7obOHGdG22TCrVx  iR2I3C-D2oDCM5p4PNk38LHISFd3bLMaKJMArgmurA |
| Success ! |

Step 7: Check digest against HTTP Body

|  |
| --- |
| **Description:** Recalculate digest of HTTP body and check against value in HTTP Header Digest using the identified hashing algorithm. |
| {  "instructedAmount": {"currency": "EUR", "amount": "123.50"},  "debtorAccount": {"iban": "DE40100100103307118608"},  "creditorName": "Merchant123",  "creditorAccount": {"iban": "DE02100100109307118603"},  "remittanceInformationUnstructured": "Ref Number Merchant"  } |
| Digest: SHA-256=+xeh7JAayYPh8K13UnQCBBcniZzsyat+KDiuy8aZYdI= |
| Success ! |