ETSI GR CIM 018 V1.1.1 (2022-09)

cross-cutting Context Information Management (CIM); NGSI-LD; Enabling chain of trust from Content Sources to Content Consumers

Disclaimer

The present document has been produced and approved by the cross-cutting Context Information Management (CIM) ETSI Industry Specification Group (ISG) and represents the views of those members who participated in this ISG.
It does not necessarily represent the views of the entire ETSI membership.

Reference
DGR/CIM-0018

Keywords
blockchain, digital signature, distributed ledger, provenance, trust, verifiable registry

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - APE 7112B
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° w061004871

Contents

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The declarations pertaining to these essential IPRs, if any, are publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (https://ipr.etsi.org/).

Pursuant to the ETSI Directives including the ETSI IPR Policy, no investigation regarding the essentiality of IPRs, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

DECT™, **PLUGTESTS™**, **UMTS™** and the ETSI logo are trademarks of ETSI registered for the benefit of its Members. **3GPP™** and **LTE™** are trademarks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners. **oneM2M™** logo is a trademark of ETSI registered for the benefit of its Members and of the oneM2M Partners. **GSM**® and the GSM logo are trademarks registered and owned by the GSM Association.

Foreword

This Group Report (GR) has been produced by ETSI Industry Specification Group (ISG) cross-cutting Context Information Management (CIM).

Modal verbs terminology

In the present document "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

Executive summary

The present document focuses on the issue of trust and identity, scrutinizing the current landscape and pointing out how the current initiatives can be integrated into the NGSI-LD [i.18] ecosystem.

The goal is to design technical means for enabling a chain of trust from content sources to content consumers that helps endorse documents, by connecting or embedding verifiable credentials into NGSI-LD documents (defined in [i.18]).

These solutions will then enhance the current NGSI-LD standard [i.18].

Introduction

When focusing the issues of trust and identity, it is important to notice that trust is orthogonal to identity: while Self-Sovereign Identity (SSI) is about a digital identity that benefits the individual, trust is not about giving identifiers to assets, but about providing provenance to an asset, thus complementing each other.

Frameworks for provenance seek to delegate and transfer trust, in order to build authenticity solutions that are decentralized. In September 2021, the Coalition for Content Provenance and Authenticity (C2PA) released its content provenance specifications. C2PA leverages DID (a form of SSI) and Verifiable Credentials from W3C®.

The W3C Verifiable Credentials and the C2PA initiative, although still in their infancy, aim at filling a gap, creating coordinated efforts towards standardization of technical specifications for provenance that robustly link content to producers.

The present document includes material copied from or derived from "Decentralized Identifiers (DIDs) v1.0" [i.10], available at https://www.w3.org/TR/did-core/ and "Verifiable Credentials Data Model v1.1" [i.11], available at https://www.w3.org/TR/vc-data-model/. Copyright© 2022 W3C® (MIT, ERCIM, Keio, Beihang).

It also uses material form the "C2PA Technical Specifications" [i.12]) (licensed under a Creative Commons Attribution 4.0 International License) and from the W3C® "Data Integrity 1.0" [i.1] Specification, published by the Credentials Community Group under the W3C® Community Final Specification Agreement (FSA).

The present document discusses possible approaches about how such specifications should be completed and integrated with distributed registry/ledger technologies having precise requirements, for the purpose of integration with the NGSI-LD ecosystem [i.18].

1 Scope

The present document focuses on the issue of trust and identity, scrutinizing the current landscape, analysing the existing requirements, and pointing out how such current initiatives and requirements can be integrated into the NGSI-LD ecosystem.

The approach revolves around two goals:

- A first goal is decentralization.
- A second goal is not only the verification of the credential, but to have means to distinguish which of its claims can be considered authoritative and which cannot, or, equivalently, to discover information that can be used to evaluate the risk of accepting the claims in the VC.

Thus, the present document examines solutions to verify integrity, and to precisely evaluate attribution and authenticity of NGSI-LD Context Information throughout its lifecycle. The goal is to design technical means for enabling a chain of trust from content sources to content consumers that helps endorse documents, by connecting or embedding verifiable credentials into NGSI-LD documents.

These solutions will then enhance the current NGSI-LD standard (ETSI GS CIM 009) [i.18].

2 References

2.1 Normative references

Normative references are not applicable in the present document.

2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] W3C® Final Community Group Report 22 July 2022: "Data Integrity 1.0".

NOTE: Available at https://www.w3.org/community/reports/credentials/CG-FINAL-data-integrity-20220722.

[i.2] W3C® Recommendation 3 May 2022: "BBS+ Signatures 2020".

NOTE: Available at https://w3c-ccg.github.io/ldp-bbs2020/.

[i.3] W3C® Recommendation 16 July 2020: "JSON-LD 1.1 Framing".

NOTE: Available at https://www.w3.org/TR/json-ld11-framing/.

[i.4] IETF draft-sporny-hashlink-07 expired on November 2021: "Cryptographic Hyperlinks ".

NOTE: Available at https://www.ietf.org/archive/id/draft-sporny-hashlink-07.txt.

[i.5] InterPlanetary File System (IPFS).

NOTE: Available at https://ipfs.tech.

[i.6] IETF RFC 7515: "JSON Web Signature (JWS)".

NOTE: Available at https://tools.ietf.org/html/rfc7515.

[i.7] IETF RFC 7516: "JSON Web Encryption (JWE)".

NOTE: Available at https://tools.ietf.org/html/rfc7516.

[i.8] IETF RFC 7519: "JSON Web Token (JWT)".

NOTE: Available at https://tools.ietf.org/html/rfc7519.

[i.9] IETF RFC 7518: "JSON Web Algorithms (JWA)".

NOTE: Available at https://tools.ietf.org/html/rfc7518.

[i.10] W3C® Recommendation 19 July 2022: "Decentralized Identifiers (DIDs) v1.0".

NOTE: Available at https://www.w3.org/TR/did-core/.

[i.11] W3C® Recommendation 03 March 2022: "Verifiable Credentials Data Model v1.1".

NOTE: Available at https://www.w3.org/TR/vc-data-model/.

[i.12] C2PA Specifications: "C2PA Technical Specification".

NOTE: Available at https://c2pa.org/specifications/specifications/1.0/specs/C2PA_Specification.html.

[i.13] IETF RFC 8785: "JSON Canonicalization Scheme (JCS)".

NOTE: Available at https://tools.ietf.org/html/rfc8785.

[i.14] IETF draft-jordan-jws-ct-01: "JWS Clear Text JSON Signature Option (JWS/CT)".

NOTE: Available at https://datatracker.ietf.org/doc/html/draft-jordan-jws-ct-01.

[i.15] W3C® Recommendation 25 February 2014: "RDF 1.1 Concepts and Abstract Syntax".

NOTE: Available at https://www.w3.org/TR/rdf11-concepts/.

[i.16] W3C® Working Group Note 07 November 2013: "RDF 1.1 JSON Alternate Serialization (RDF/JSON)".

NOTE: Available at https://www.w3.org/TR/rdf-json/.

[i.17] W3C® Draft Community Group Report 13 April 2021: "RDF Dataset Canonicalization".

NOTE: Available at https://json-ld.github.io/rdf-dataset-canonicalization/spec.

[i.18] ETSI GS CIM 009: "Context Information Management (CIM); NGSI-LD API".

[i.19] ETSI EN 319 412-1: "Electronic Signatures and Infrastructures (ESI); Certificate Profiles; Part 1: Overview and common data structures".

[i.20] ISO 3166-1: "Codes for the representation of names of countries and their subdivisions - Part 1: Country codes".

[i.21] Payments Services Directive (EU) 2015/2366 of the European Parliament and of the Council of 25 November 2015 on payment services in the internal market, amending Directives 2002/65/EC, 2009/110/EC and 2013/36/EU and Regulation (EU) No 1093/2010, and repealing Directive 2007/64/EC.

[i.22] ETSI TS 119 495: "Electronic Signatures and Infrastructures (ESI); Sector Specific Requirements; Certificate Profiles and TSP Policy Requirements for Open Banking".

[i.23] ISO 17442-1:2020: "Financial services - Legal entity identifier (LEI) - Part 1: Assignment".

NOTE: Available at https://www.iso.org/standard/78829.html.

[i.24] JSON-LD Signatures implementation.

NOTE: Available at https://github.com/digitalbazaar/jsonld-signatures.

[i.25] jsonld-signatures-bbs implementation.

NOTE: Available at https://github.com/mattrglobal/jsonld-signatures-bbs.

3 Definition of terms, symbols and abbreviations

3.1 Terms

For the purposes of the present document, the following terms apply:

assertion: data structure which represents a statement asserted by an actor concerning the asset

atomic entity: signed NGSI-LD Entity with only one attribute

Base64url: modification of the main Base64 standard, the purpose of which is the ability to use the encoding result as filename or URL address

BBS+ signature: digital signature algorithm by Boneh, Boyen, and Shachum

BLS signature: Boneh–Lynn–Shacham (BLS) cryptographic signature scheme

CL signature: signature scheme developed by Jan Camenisch and Anna Lysyanskay

claim: assertion made about a subject

controller document: set of data that specifies one or more relationships between a controller and a set of data, such as a set of public cryptographic keys

derivation process: process that transforms NGSI-LD attributes in Sealed Attributes

DID controller: entity that has the capability to make changes to a DID document

DID document: set of data describing the DID subject

DID method: definition of how a specific DID method scheme is implemented

DID resolution: process that takes as its input a DID and a set of resolution options and returns a DID document

hard binding: one or more cryptographic hashes that uniquely identifies either the entire asset or a portion thereof

holder: role an entity might perform by possessing one or more verifiable credentials and generating presentations from them

issuer: role an entity can perform by asserting claims about one or more subjects, creating a verifiable credential

manifest: set of information about the provenance of an asset based on the combination of one or more assertions (including content bindings), a single claim, and a claim signature

NGSI-LD Attribute: reference to both an NGSI-LD Property and to an NGSI-LD Relationship

NGSI-LD Context Broker: architectural component that implements all the NGSI-LD interfaces

NGSI-LD Context Consumer: agent that uses the query and subscription functionality of NGSI-LD to retrieve context information

NGSI-LD Context Producer: agent that uses the NGSI-LD context provision and/or registration functionality to provide or announce the availability of its context information to an NGSI-LD Context Broker

NGSI-LD Context Source: source of context information which implements the NGSI-LD consumption and subscription (and possibly provision) interfaces defined by the present document

NGSI-LD Entity: informational representative of something that is supposed to exist in the real world, physically or conceptually

NGSI-LD Property: description instance which associates a main characteristic, i.e. an **NGSI-LD Value**, to either an NGSI-LD Entity, an NGSI-LD Relationship or another NGSI-LD Property and that uses the special *hasValue* property to define its target value

NGSI-LD Relationship: description of a directed link between a subject which is either an NGSI-LD Entity, an NGSI-LD Property or another NGSI-LD Relationship on one hand, and an object, which is an NGSI-LD Entity, on the other hand, and which uses the special *hasObject* property to define its target object

recreation process: opposite process of the Derivation Process

Sealed Attribute: NGSI-LD attribute structure with "ngsildproof" property

soft binding: content identifier that is either:

a) not statistically unique, such as a fingerprint; or

b) embedded as a watermark in the identified digital content

verification method: set of parameters that can be used together with a process to independently verify a proof

3.2 Symbols

Void.

3.3 Abbreviations

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

API	Application Programming Interface
ASCII	American Standard Code for Information Interchange
BBS	Boneh, Boyen, and Shachum
BLS	Boneh-Lynn-Shacham
BMFF	Base Media File Format
CBOR	Concise Binary Object Representation
CID	Content IDentifier
CL	Jan Camenisch and Anna Lysyanskay
DID	Decentralized IDentifier
DoS	Denial of Service
GDPR	General Data Protection Regulation
HMAC	Hash-based Message Authentication Code
HTTP	Hypertext Transfer Protocol
ID	Identifier
I-JSON	Internet JSON
IPFS	InterPlanetary File System
IRI	Internationalized Resource Identifier
JCS	JSON Canonicalization Scheme
JOSE	JSON Object Signing and Encryption
JSON	JavaScript Object Notation
JSON-LD	JSON-Linked Data
JUMBF	JPEG Universal Metadata Box Format
JWA	JSON Web Algorithms
JWE	JSON Web Encryption
JWS	JSON Web Signature
JWS/CT	JWS "Clear Text"
JWT	JSON Web Token
MAC	Message Authentication Code
NGSI	Next Generation Service Interfaces
NGSI-LD	NGSI-Linked Data
PIA	Privacy Impact Assessment

PKI	Public Key Infrastructure
RDF	Resource Description Format
SSI	Self-Sovereign Identity
URI	Uniform Resource Identifier
URL	Universal Resource Locator
UTF	Unicode (or Universal Coded Character Set) Transformation Format
UTF-8	Unicode Transformation Format, 8 bit
VC	Verifiable Credential
ZKP	Zero Knowledge Proof

4 Data integrity and provenance

4.1 Goals

This clause is focused on data integrity problem, specifically to JSON-LD documents and NGSI-LD Entities.

The preferred solution, in both literature and industry, to achieve data integrity is the implementation of a digital signature system, where, in this case, a digest file of the NGSI-LD Entity, cryptographically encoded with the signer private key, bound with it, will guarantee the non-corruption of data and the association to a specific private key.

Thus, the first goal of the present document is to design a standard to guarantee the integrity of an NGSI-LD Entity.

In order to do this, it is important to create guidelines that ensure data accuracy and consistency over the Entity's entire life-cycle.

4.2 Scenarios and Use Cases

A generic scenario can be the signing process of a generic JSON-LD document.

In this case, a typical scenario is the generation of an NGSI-LD Entity from a content creator (Producer or Source) sent through multiple Context Brokers to a Client.

In this scenario, where a typical NGSI-LD Entity will contain Attributes (Property or Relationship), it is important to guarantee that these values will not be altered through all its cycles, so that a client without contact creators can be sure of its integrity.

4.3 Architectures and Existing Specifications

4.3.1 W3C data integrity model

4.3.1.0 Introduction

One already implemented solution can be the W3C Data integrity [i.1], which is not yet a W3C standard.

With this, it is possible to create a data integrity proof, which is a set of attributes that represent a digital proof and all parameters required to verify it.

A data integrity proof is containing the following attributes:

- type: which indicates the specific type of digital signature used
- proofPurpose: a parameter that ensures that the digital proof is used for the reason it was created for
- verificationMethod: a set of parameters required to independently verify the proof
- created: date and time of the proof generation
- proofValue: the value of the encoded hash file

Here is a proof example that uses the JcsSignature2020 proof type:

```
"proof": {
  "type": "JcsSignature2020",
  "created": "2020-11-05T19:23:24Z",
  "verificationMethod":"https://di.example/issuer#z6MkjLrk3gKS2nnkeWcmcxiZPGskmesDpuwRBorgHxUXfxnG",
  "proofPurpose": "assertionMethod",
  "proofValue":"zQeVbY4oey5q2M3XKaxup3tmzN4DRFTLVqpLMweBrSxMY2xHX5XT…"
}
```

More optional attributes can be implemented in the data integrity proof in order to add further information or limitation (`domain, challenge`).

At the moment the following signature suites are contemplated in W3C Data integrity specification [i.1]: eddsa-2022, nist-ecdsa-2022, koblitz-ecdsa-2022, rsa-2022, pgp-2022, bbs-2022, eascdsa-2022, ibsa-2022, and jws-2022.

4.3.1.1 Controller Document

The verification process is possible through the access of the Controller Document, a set of data that specifies the relationship between a controller, the entity who can change the controller document, and other data sets such as a public cryptographic key.

Whoever wants to verify the data integrity proof needs to ensure that a verification method is bound to a specific controller, by going from the verification method attribute in the proof to the controller document, ensuring that this also contains the same verification method and the same proof purpose.

The `verificationMethod` property in the Controller Document is optional, but if present it has to be a set of verification method's map.

Each verification method has to include the following properties:

- `id:` a string that conforms to the URL syntax.
- `type:` which indicates the specific type of verification method used.
- `controller:` a string that conforms to the URL syntax.

The `controller` value for a verification method is not necessarily a controller. Controllers are expressed using the controller property at the highest level of the controller document.

The verification method could be filled with other verification material for example the cryptographic public key, accordingly with the verification method type.

Examples of supported verification material are `publicKeyJwk` and `publicKeyMultibase`, representing respectively the JSON Web Key and the multibase encoded public key.

Here is a Controller Document example:

```
{
  "@context": [
     "https://w3id.org/security/suites/ed25519-2020/v1",
     "https://w3id.org/security/v2"
  ],
  "id": "https://example.org/controllerDocument",
  "verificationMethod":{
    "id":https://example.org/controllerDocument.json#z6MksnCpNjGV",
    "type": "Ed25519VerificationKey2020",
    "controller": "https://example.org/controllerInfo.json",
    "publicKeyMultibase": "z6MksnCpNjGV"
  },
  "assertionMethod": {
    "id":https://example.org/controllerDocument.json#z6MksnCpNjGV",
    "type": "Ed25519VerificationKey2020",
    "controller": "https://example.org/controllerDocument.json",
    "publicKeyMultibase": "z6MksnCpNjGVkL87mnaC2sbBbAZ"
  }
}
```

The verification method could be included by reference using an URL and its properties will need to be retrieved from elsewhere in the Controller Document or from another Controller Document. This is done by dereferencing the URL and searching the resulting resource for a verification method map with an id property whose value matches the URL.

The Controller Document can express also verification relationship between the controller and a verification method, enabling this to be used for different purposes.

Contemplated verification relationships are: Authentication, Assertion, keyAgreement, capabilityInvocation, capabilityDelegation.

The W3C Data integrity specification [i.1] suggests that any verification relation should be registered in the Data Integrity Specification Registries, which is not already defined.

At the actual state of the data integrity specification no other proof types that are not data integrity signatures are implemented.

4.3.1.2 Selective disclosure and aggregation

A secondary problem of the digital signature of a piece of data is that, in order to verify it, the entire document needs to be shared.

In many scenarios it can be useful or even necessary to choose what information can be shared without renouncing the integrity granted by the proof.

Also, in order to recreate provenance of data, it is commonly used the possibility to merge single data fragments without losing information of the original signatures.

Data integrity is a prerequisite, not only required for the final data, coming from multiple merging processes, but also required for every single original data fragment.

One possible solution can be the signature aggregation mechanism.

This property will guarantee the possibility to create an aggregated signature from every single message signature which can be verified by an aggregated public key created from all signer's public keys.

This process does not require the aggregator's signature, so whoever performs the aggregation does not enter inside the trust model.

The overhaul data integrity is guaranteed because only the aggregated public key made out of all original signers can verify the aggregated signature.

Additional information about the signer's identity, or its public key, can be added to the original fragments and signed in order to, subsequently, prove which fragment was signed by whom.

In addition, with Data Integrity Specification it is possible to utilize signature suites that allow the implementation of a selective disclosure, with Zero Knowledge proof technology and signature aggregation, such as BBS signature suite or Cl signature suite.

4.3.1.3 W3C BBS+ Signature suite

The BBS signature was invented by Dan Boneh, Ben Lynn and Hovav Shacham.

It uses elliptic curve pairings and guarantees the following benefits:

- is very simple to use (aside the extreme complexity of elliptic curve pairing)
- is deterministic and verifiably
- is able to provide signatures aggregations
- Signatures and keys can be 32 bytes long, but not at the same time (BLS12-381 48 bytes)

The W3C BBS+ Signatures 2020 specification [i.2], which is not a W3C standard, describes the utilization of BBS+ signatures to provide the capability of zero-knowledge proof disclosures.

Using traditional digital signatures, starting with a message and the issuer private key, it is possible to create a signature.

On verification, starting with the original message and issuer public key, it is possible to establish if the signature is valid or not.

Signature validation tells if the message is untampered with and it gives origin authenticity, in the sense that it is possible to know who produced that signature.

Traditional signature scheme can be represented as in Figure 4.3.1.3-1.

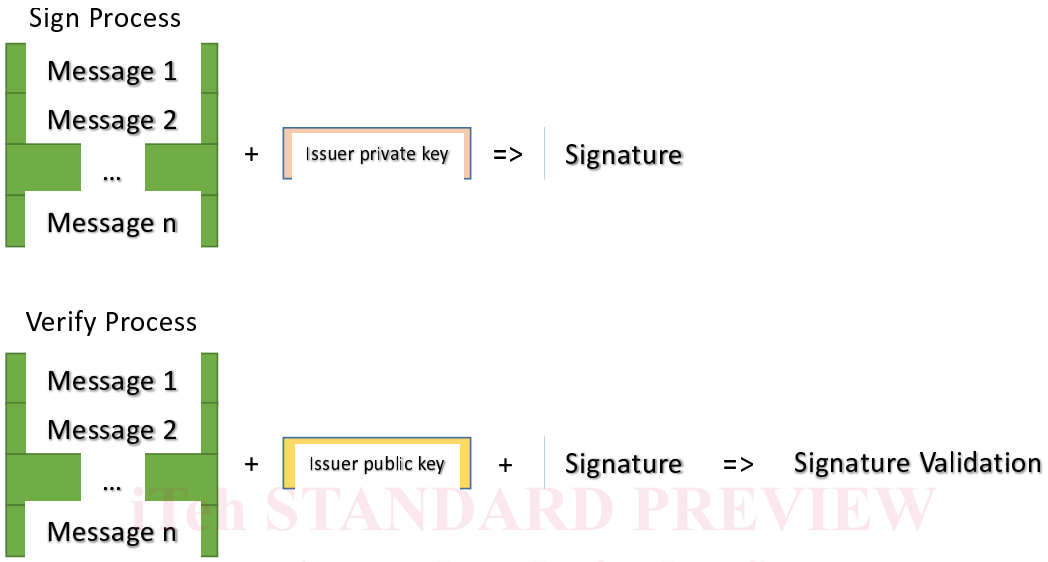

Figure 4.3.1.3-1: Traditional signature scheme

On the other side, BBS+ signature is a multi-message digital signature scheme, represented as in Figure 4.3.1.3-2.

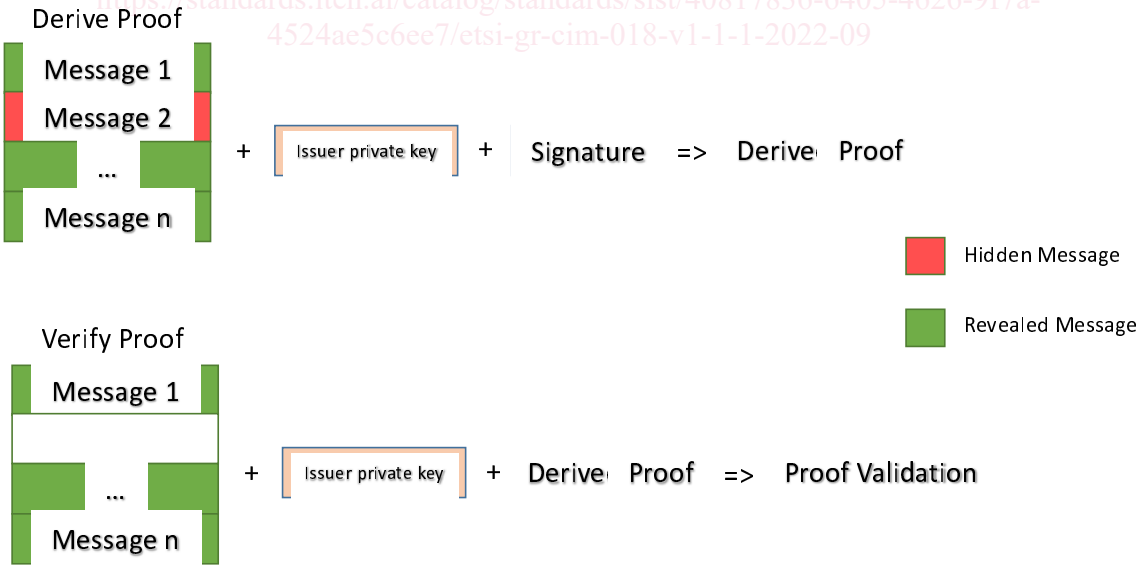

Figure 4.3.1.3-2: BBS signature scheme

Instead of a single message on input it is possible to have many of them. A group of messages are signed with the issuer's private key, and can be verified in groups as always.

In addition to traditional digital signatures it is possible to create what is called a derived proof.

Choosing one or more messages, from the original group, that has to be revealed, with issuer's public key and the original signature it is possible to create a derived proof.

On verification, a verifier using the sets of messages revealed with the issuer's public key and the derived proof can verify if the proof is valid or not.

In this scenario there is the same integrity for all messages, in addition to the possibility of sensitive disclosure of information.

The BLS Signature Proof is a proof of knowledge of the original BLS signature. This means that the information of the original signature is not shared, but only that its existence and knowledge is proved.

The signature aggregation property is guaranteed using the BLS signature.

An aggregate signature is a shorter representation of n signatures provided by different users on different messages.

The linearity property of elliptic curve pairing guarantees the possibility to verify in batch a group of messages following the schema of Figure 4.3.1.3-3.

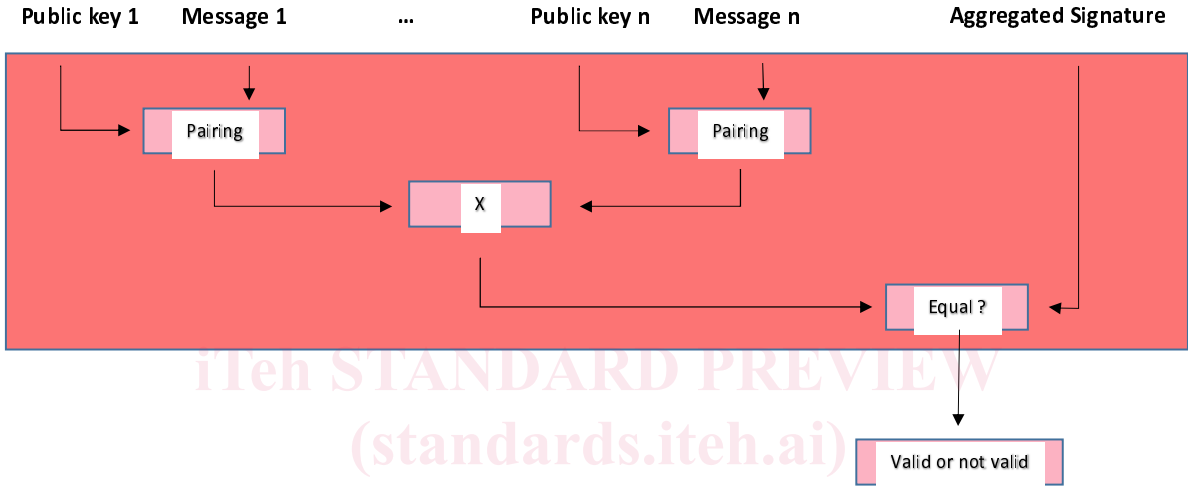

Figure 4.3.1.3-3: BBS multi-signed message Verification scheme

4.3.1.4 Bls12381G2Key2020

Is a Key Pair Standard based on elliptic curve pairing or bilinear map.

This proof has to contain, over the already defined Data integrity proof properties, a new property:

`requiredRevealStatements:` which has to be an array of unsigned integers representing the indices of the statements in the canonical form that has always to be revealed in a derived proof.

Here is an example of Bls Signature 2020 generated with Bls12381G2Key2020:

```
"proof": {
        "type": "BbsBlsSignature2020",
        "created": "2020-04-25",
        "verificationMethod": "did:example:489398593#test",
        "proofPurpose": "assertionMethod",
        "proofValue":"F9uMuJzNBqj4j+HPTvWjUN/MNoe6KRH0818WkvDn2Sf7kg1P17YpN",
        "requiredRevealStatements": [ 4, 5 ]
}
```

The main characteristic of a BBS signed document is the possibility to create a derived document (revealed document), containing revealed statements from the original document and a derived proof.

Implementing in JSON-LD the derived document is created via JSON-LD frame [i.3].

This is possible due to BBS proof of knowledge linked data proof which is a proof that is derived from a BbsBlsSignature2020 linked data proof, where a subset of the original statements is revealed.

A derived proof has to contain a type attribute that has a type equal to BbsSignatureProof2020.

A derived proof, over the already defined Data integrity proof properties, has to contain a nonce attribute.