SAML V2.0 Subject Identifier Attributes Profile Version 1.0

Committee Specification ~~Draft 03 /~~ ~~Public Review Draft 03~~01

~~25 September 2018~~

16 January 2019

Specification URIs

This version:

https://docs.oasis-open.org/security/saml-subject-id-attr/v1.0/cs01/saml-subject-id-attr-v1.0-cs01.odt (Authoritative)
https://docs.oasis-open.org/security/saml-subject-id-attr/v1.0/cs01/saml-subject-id-attr-v1.0-cs01.html
https://docs.oasis-open.org/security/saml-subject-id-attr/v1.0/cs01/saml-subject-id-attr-v1.0-cs01.pdf

Previous version:

http://docs.oasis-open.org/security/saml-subject-id-attr/v1.0/csprd03/saml-subject-id-attr-v1.0-csprd03.odt (Authoritative)
http://docs.oasis-open.org/security/saml-subject-id-attr/v1.0/csprd03/saml-subject-id-attr-v1.0-csprd03.html
http://docs.oasis-open.org/security/saml-subject-id-attr/v1.0/csprd03/saml-subject-id-attr-v1.0-csprd03.pdf ~~other format~~

~~**Previous version:**~~

~~http://docs.oasis-open.org/security/saml-subject-id-attr/v1.0/csprd02/saml-subject-id-attr-v1.0-csprd02.odt (Authoritative)~~
~~http://docs.oasis-open.org/security/saml-subject-id-attr/v1.0/csprd02/saml-subject-id-attr-v1.0-csprd02.html~~
~~http://docs.oasis-open.org/security/saml-subject-id-attr/v1.0/csprd02/saml-subject-id-attr-v1.0-csprd02.pdf~~

Latest version:

~~http://docs.oasis-open.org/security/saml-subject-id-attr/v1.0/saml-subject-id-attr-v1.0.odt (Authoritative)~~
~~http://docs.oasis-open.org/security/saml-subject-id-attr/v1.0/saml-subject-id-attr-v1.0.html~~
~~http://docs.oasis-open.org/security/saml-subject-id-attr/v1.0/saml-subject-id-attr-v1.0.pdf~~
https://docs.oasis-open.org/security/saml-subject-id-attr/v1.0/saml-subject-id-attr-v1.0.odt (Authoritative)
https://docs.oasis-open.org/security/saml-subject-id-attr/v1.0/saml-subject-id-attr-v1.0.html
https://docs.oasis-open.org/security/saml-subject-id-attr/v1.0/saml-subject-id-attr-v1.0.pdf

Technical Committee:

OASIS Security Services (SAML) TC

Chair:

Thomas Hardjono (hardjono@mit.edu), M.I.T.

Editor~~s~~:

Scott Cantor (cantor.2@osu.edu), Internet2

Additional artifacts:

This prose specification is one component of a Work Product that also includes:

~~XML schema: http://docs.oasis-open.org/security/saml-subject-id-attr/v1.0/csprd03/schema/saml-subject-id-attr-v1.0.xsd~~

- XML schema: https://docs.oasis-open.org/security/saml-subject-id-attr/v1.0/cs01/schema/saml-subject-id-attr-v1.0.xsd

Related work:

This specification is related to:

- eduPerson Object Class Specification (201602) http://software.internet2.edu/eduperson/internet2-mace-dir-eduperson-201602.html.

Declared XML namespace~~s~~:

- urn:mace:shibboleth:metadata:1.0

Abstract:

This specification standardizes two new SAML Attributes to identify security subjects, as a replacement for long-standing inconsistent practice with the `<saml:NameID>` and `<saml:Attribute>` constructs, and to address recognized deficiencies with the SAML V2.0 `urn:oasis:names:tc:SAML:2.0:nameid-format:persistent` Name Identifier format.

Status:

This document was last revised or approved by the OASIS Security Services (SAML) TC on the above date. The level of approval is also listed above. Check the “Latest version” location noted above for possible later revisions of this document. Any other numbered Versions and other technical work produced by the Technical Committee (TC) are listed at https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security#technical.

TC members should send comments on this specification to the TC’s email list. Others should send comments to the TC's public comment list, after subscribing to it by following the instructions at the "Send A Comment" button on the Technical Committee’s web page at https://www.oasis-open.org/committees/security/.

~~This Committee Specification Public Review Draft~~This specification is provided under the RF on RAND Terms Mode of the OASIS IPR Policy, the mode chosen when the Technical Committee was established. For information on whether any patents have been disclosed that may be essential to implementing this Work Product, and any offers of patent licensing terms, please refer to the Intellectual Property Rights section of the TC's web page (https://www.oasis-open.org/committees/security/ipr.php).

Note that any machine-readable content (aka Computer Language Definitions) declared Normative for this Work Product is provided in separate plain text files. In the event of a discrepancy between any such plain text file and display content in the Work Product's prose narrative document(s), the content in the separate plain text file prevails.

Citation format:

When referencing this Work Product the following citation format should be used:

[SAML-SubjectID-v1.0]

SAML V2.0 Subject Identifier Attributes Profile Version 1.0. Edited by Scott Cantor. ~~25 September 2018.~~16 January 2019. OASIS Committee Specification ~~Draft 03 / Public Review Draft 03.~~01. ~~http://docs.oasis-open.org/security/saml-subject-id-attr/v1.0/csprd03/saml-subject-id-attr-v1.0-csprd03.html.~~https://docs.oasis-open.org/security/saml-subject-id-attr/v1.0/cs01/saml-subject-id-attr-v1.0-cs01.html. Latest version: https://docs.oasis-open.org/security/saml-subject-id-attr/v1.0/saml-subject-id-attr-v1.0.html.

Notices

Table of Contents

1 Introduction

1.1 IPR Policy

This specification is provided under the RF on RAND Terms Mode of the OASIS IPR Policy, the mode chosen when the Technical Committee was established.

For information on whether any patents have been disclosed that may be essential to implementing this specification, and any offers of patent licensing terms, please refer to the Intellectual Property Rights section of the TC's web page (https://www.oasis-open.org/committees/security/ipr.php).

1.2 Terminology and Notation

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

Conventional XML namespace prefixes are used throughout the listings in this specification to stand for their respective namespaces as follows, whether or not a namespace declaration is present in the example:

Prefix	***XML Namespace***	***Comments***
`saml:`	urn:oasis:names:tc:SAML:2.0:assertion	This is the SAML V2.0 assertion namespace [SAML2Core].
`samlp:`	urn:oasis:names:tc:SAML:2.0:protocol	This is the SAML V2.0 protocol namespace [SAML2Core].
`md:`	urn:oasis:names:tc:SAML:2.0:metadata	This is the SAML V2.0 metadata namespace [SAML2Meta].
`mdattr:`	urn:oasis:names:tc:SAML:metadata:attributes	This is the SAML V2.0 metadata extension for entity attributes namespace [MetaAttr].
`shibmd:`	urn:mace:shibboleth:metadata:1.0	This is a SAML V2.0 metadata extension namespace defined by this document and its accompanying schema.
`xsd:`	http://www.w3.org/2001/XMLSchema	This namespace is defined in the W3C XML Schema specification [XMLSCHEMA-2].

This specification uses the following typographical conventions in text: `<ns:Element>`, `Attribute`, **Datatype**, `OtherCode`.

This specification uses the following typographical conventions in XML listings:

```
Listings of XML schemas appear like this.
```

```
Listings of XML examples appear like this. These listings are non-normative.
```

1.3 Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997. http://www.ietf.org/rfc/rfc2119.txt.

[RFC2234] Crocker, D, Overell, P., "Augmented BNF for Syntax Specifications: ABNF", RFC 2234, November 1997. http://www.ietf.org/rfc/rfc2234.txt.

[SAML2Core] *Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0.* Edited by Scott Cantor, John Kemp, Rob Philpott, Eve Maler. 15 March 2005. OASIS Standard. http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf

[MetaAttr] *SAML V2.0 Metadata Extension for Entity Attributes Version 1.0*. Edited by Scott Cantor. 4 August 2009. OASIS Committee Specification. http://docs.oasis-open.org/security/saml/Post2.0/sstc-metadata-attr-cs-01.pdf. Latest version: http://docs.oasis-open.org/security/saml/Post2.0/sstc-metadata-attr.pdf.

[SAML2Errata] *SAML V2.0 Errata.* Edited by Scott Cantor. 1 May 2012. OASIS Approved Errata. http://docs.oasis-open.org/security/saml/v2.0/errata05/os/saml-v2.0-errata05-os.pdf. Latest version: http://docs.oasis-open.org/security/saml/v2.0/sstc-saml-approved-errata-2.0.pdf

[SAML2Meta] *Metadata for the OASIS Security Assertion Markup Language (SAML) V2.0.* Edited by Scott Cantor, Jahan Moreh, Rob Philpot, Eve Maler. 15 **March 2005**. OASIS Standard. http://docs.oasis-open.org/security/saml/v2.0/saml-metadata-2.0-os.pdf

[SAML2Prof] *Profiles for the OASIS Security Assertion Markup Language (SAML) V2.0.* Edited by John Hughes, Scott Cantor, Jeff Hodges, Frederick Hirsch, Prateek Mishra, Rob Philpot, Eve Maler. 15 **March 2005**. OASIS Standard. http://docs.oasis-open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf

[XMLSCHEMA-2] *XML Schema Part 2: Datatypes Second Edition*. Paul V. Biron, A. Malhotra, Editors. W3C Recommendation. October 28, 2004. http://www.w3.org/TR/2004/REC-xmlschema-2-20041028/. Latest version: http://www.w3.org/TR/xmlschema-2/.

1.4 Non-Normative References

[eduPerson] Internet2, “eduPerson Object Class Specification (201602)”, February 2016. http://software.internet2.edu/eduperson/internet2-mace-dir-eduperson-201602.html.

[RFC4648] Josefson, S., “The Base16, Base32, and Base64 Data Encodings”, RFC 4648, October 2006. http://www.ietf.org/rfc/rfc4648.txt.

[ShibMetaExt] Shibboleth Project, “Shibboleth Metadata Extensions V1.0”, July 2018. https://wiki.shibboleth.net/confluence/x/QACt.

2 Motivation

2.1 Problem Statement

Identification of subjects in security protocols and applications has a fraught history of inconsistent syntax, bugs, terrible but deeply cemented practices such as misuse of email addresses, vertical market-specific approaches, and failure to precisely communicate intended semantics and constraints. These problems lead to overly complex burdens on both asserting and relying parties to issue and consume a variety of different identifiers in different formats, many of which work poorly with off the shelf applications. Much of this is self-inflicted fragmentation due to the constant tension between fixing problems with new solutions and avoiding new solutions to ensure wider adoption.

SAML itself has its origins in a design philosophy that tried to avoid breaking new ground in this area, and instead attempted to design for generality, which is valuable, but did not ease adoption due to a lack of guidance. SAML also complicates itself by providing an optional, singly-appearing construct for identification (the `<saml:NameID>` element) *and* a more general multiply-appearing `<saml:Attribute>` construct that inherently overlap.

This, together with inconsistent technical precision by implementers and deployers, creates complexity. Deployment experience has shown that use of the NameID feature is confusing in many implementations. It also, through its presence in the SAML Single Logout protocol, potentially appears (indirectly but recoverably) in web access logs, leading to the added complexity of encryption when privacy is a consideration.

There is a general consensus by most federated identity practitioners around a few common requirements:

- Identifiers should be as stable as possible and should have little or no risk of reassignment to different subjects due to the lack of tight synchronization[1] inherent between loosely-coupled systems.
- Opaque (i.e., superficially random) identifiers are inherently more stable than name-based identifiers or email addresses in many organizations.
- Identifiers should be compact and simple to handle and manipulate.
- The ability to clearly express the scope of an identifier's uniqueness and enforce policy stipulating the asserting parties permitted to issue an identifier is crucial to federated systems and the lack of such policy has led to widely-publicized breaches.

Another requirement perhaps more common to education and research is the ability for different asserting parties to issue the same identifier. This is facilitated by ensuring the scope of an identifier is part of its value and not implicit in a protocol-specific construct specific to an asserting party.

SAML does not define an identifier that meets all of these requirements well. It does standardize a kind of NameID termed "persistent" that meets some of them in the particular case of so-called "pairwise" identification, where ~~an~~a given subject's identifier varies by relying party. It has seen minimal adoption outside of a few contexts, and fails at the "compact" and "simple to handle" criteria above, on top of the disadvantages inherent with all NameID usage.

Pairwise identification may help meet certain privacy and regulatory requirements (though this is far from clear to date), but does not address many common use cases that demand cross-system correlation without the friction of complex linking protocols and the involvement of the data subject.

In addition, it has come to light that many, if not most, applications have a predisposition to handle identifiers case-insensitively, partly due to a long-standing, though factually untrue, assumption that e-mail address mailbox names are case-insensitive data. SAML's "persistent" NameID definition explicitly requires case-sensitive handling, making them impossible to use safely with such applications without resorting to additional layers of profiling. Note that any other specification promulgating such identifiers is potentially unsafe in combination with such applications and should be used with caution.

[1] It's worth noting that SAML actually defines a protocol for managing changes to NameID values, but it has seen very little adoption, further demonstrating the lack of value of NameID usage.

For all of these reasons, this profile attacks these problems by taking a clean-slate approach that abandons existing practice instead of attempting to layer more profiling and out of band agreements on top of existing solutions, an approach that has seemingly reached its breaking point.

2.2 Relationship to Existing Work

A clean slate notwithstanding, this profile is based on a thorough review of practice within the higher education sector, which has seen extensive adoption of SAML and partially-successful efforts to standardize subject identification and avoid the “email address” trap that most of the technical world fell into many years ago.

Among the significant work in this space, the [eduPerson] schema includes a number of identifier attributes, some widely adopted and some less so. This profile is particularly influenced by:

- Experience with the SAML “persistent” NameID construct and the related eduPersonTargetedID attribute.
- The eduPersonPrincipalName and eduPersonUniqueId attributes, the former successful but deeply flawed, the latter less successful but more carefully defined.
- Success with DNS domain-based scoping of values and managing policy around their use in SAML.
- Challenges in the adoption of profiles required to accommodate the limitations of widely deployed identifiers.

Portions of this specification are borrowed liberally from the [eduPerson] specification in a deliberate desire to remain consistent with the formulation of the eduPersonUniqueId attribute.

This specification also incorporates the relevant subset of a SAML Metadata extension schema, originally defined by the Shibboleth Project [ShibMetaExt]. This extension has seen extensive adoption, and is included here to support centralizing and automating policy for authorizing asserting parties to issue identifiers in particular scopes. The XML namespace of this extension (a URN issued by the Shibboleth Project) is maintained to remain compatible with existing implementations and deployments dating back many years.

3 SAML V2.0 Subject Identifier Attributes Profile Version 1.0

3.1 Required Information

Identification: urn:oasis:names:tc:SAML:profiles:subject-id

Contact information: security-services-comment@lists.oasis-open.org

Description: Given below.

Updates: None.

3.2 Overview

This profile defines a pair of SAML Attributes providing for unique identification of security subjects (which are generally but not exclusively people). One is designed for general use as a ~~correlatable~~globally-unique identifier, and the other is a pairwise identifier suitable for more specialized uses.

Both SAML Attributes are limited to a single value when expressed in SAML assertions and other constructs. They may be mapped to and from other technical forms (e.g., LDAP attributes) but this profile does not include such mappings.

In the terminology used in this profile:

- "asserting party" refers to a uniquely-named SAML entity that issues assertions containing one or both of these Attributes
- "relying party" refers to one or more uniquely-named SAML entities that receive assertions containing one or both of these Attributes

In addition, this profile defines a signaling mechanism for a relying party to express its subject identification requirements via SAML metadata [SAML2Meta], by means of the `<mdattr:EntityAttributes>` extension [MetaAttr]. This allows asserting parties to unambiguously understand the requirements of a peer and facilitates deployment profiles that wish to mandate support for one or both of these Attributes, while maintaining appropriate privacy expectations.

Finally, this profile incorporates and re-publishes in a standards-based context an existing SAML metadata extension element that documents attribute "scopes" an asserting party is authorized to use for its SAML Attributes (according to the issuer of that metadata).

3.3 General Purpose Subject Identifier

For general purpose identification of subjects, the following SAML Attribute is defined:

Name: `urn:oasis:names:tc:SAML:attribute:subject-id`

NameFormat: `urn:oasis:names:tc:SAML:2.0:attrname-format:uri`

This is a long-lived, non-reassignable, omni-directional identifier suitable for use as a globally-unique external key. Its value for a given subject is independent of the relying party to whom it is given.

3.3.1 Syntax and Handling

The `<saml:Attribute>` element MUST contain exactly one `<saml:AttributeValue>` element, whose `xsi:type` SHOULD be absent or if present MUST BE bound to the XML Schema `xsd:string` data type [XMLSCHEMA-2].

Any leading or trailing whitespace, as defined by XML (ASCII 32, ASCII 9, ASCII 10, ASCII 13), present in the `<saml:AttributeValue>` element's content is not significant and MUST be stripped by the relying party prior to evaluation or comparison.

The value consists of two substrings (termed a "unique ID" and a "scope" in the remainder of this definition) separated by an `@` symbol (ASCII 64) as an inline delimiter.

The unique ID consists of 1 to 127 ASCII characters, each of which is either an alphanumeric ASCII character, an equals sign (ASCII 61), or a hyphen (ASCII 45). The first character MUST be alphanumeric.

The scope consists of 1 to 127 ASCII characters, each of which is either an alphanumeric ASCII character, a hyphen (ASCII 45), or a period (ASCII 46). The first character MUST be alphanumeric. The scope deliberately resembles, and often is, a DNS domain name, but is drawn from a more limited character set due to case folding considerations, and no attempt is made to limit the allowable grammar to legal domain names (e.g., it allows consecutive periods).

The ABNF [RFC2234] grammar is therefore:

```
<value> = <uniqueID> "@" <scope>
<uniqueID> = (ALPHA / DIGIT) 0*126(ALPHA / DIGIT / "=" / "-")
<scope> = (ALPHA / DIGIT) 0*126(ALPHA / DIGIT / "-" / ".")
```

Value comparison MUST be performed case-insensitively (that is, values that differ only by case are the same, and MUST refer to the same subject).

In the grammar above, ~~only~~ the ALPHA production contains characters that can be expressed in both upper and lower case. It is RECOMMENDED that the unique ID be exclusively upper- or lower-case when expressed or stored to facilitate ease of comparison. Further, it is RECOMMENDED that scopes be expressed in lower case, since they are generally chosen independently of more “entrenched” decisions and are frequently, though not required to be, in the form of DNS domains. See also Section 3.5.2.2 for additional motivation.

3.3.2 Semantics and Practices

A value (the unique ID and scope together) MUST be bound to one and only one subject, but the same unique ID given a different scope may refer to the same or (far more likely) a different subject.

The relationship between an asserting party and a scope is an arbitrary one and does not reflect any assumed relationship between a scope in the form of a domain name and a domain found in a given SAML entity identifier. This indirect rela~~la~~tionship is formally expressible in SAML metadata via the extension defined in Section 3.5.2.

A value MUST NOT be assigned to more than a single subject over its lifetime of use under any circumstances. The unique ID should therefore be constructed in a fashion that reduces the probability of non-technical or political considerations leading to a violation of this requirement, and any such violation should be treated as a potential security risk to the relying parties to which the value may have been given.

Relying parties should not treat this identifier as an email address for the subject as it is unlikely (though not precluded) for it to be valid for that purpose. Most organizations will find that existing email address values will not serve well as values for this Attribute.

The unique ID should not change as a result of a change to any other data associated with the subject (e.g., name, email address, age, organizational role).

A given value MUST identify the same subject regardless of the context of use or the relying parties to which the Attribute is given. It is therefore to be assumed by relying parties that receive a given value that the same subject has been identified.

Note that, policy permitting, a given value could be provided by any asserting party, and the requirement still holds: identical values correspond to the same subject. While it will be common in many deployments to limit values with a given scope to a single asserting party, this is ultimately left to the discretion of the relying party and the use case.

A single subject MAY be identified simultaneously by a single asserting party by multiple values, but this should be minimized to the extent possible.

3.3.3 Example

The following is an example of the SAML Attribute defined in this section:

```
<saml:Attribute Name="urn:oasis:names:tc:SAML:attribute:subject-id"
        NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri">
    <saml:AttributeValue>idm123456789@example.com</saml:AttributeValue>
```

```
</saml:Attribute>
```

3.4 Pairwise Subject Identifier

For pairwise identification of subjects, the following SAML Attribute is defined:

Name: `urn:oasis:names:tc:SAML:attribute:pairwise-id`

NameFormat: `urn:oasis:names:tc:SAML:2.0:attrname-format:uri`

This is a long-lived, non-reassignable, uni-directional identifier suitable for use as a unique external key specific to a particular relying party. Its value for a given subject depends upon the relying party to whom it is given, thus preventing unrelated systems from using it as a basis for correlation.

3.4.1 Syntax and Handling

The requirements for this Attribute are identical to those described in Section 3.3.1. That is, values of this Attribute are indistinguishable, lacking the context, from the other.

3.4.2 Semantics and Practices

Given a particular relying party, a value (the unique ID and scope together) MUST be bound to only one subject, but the same unique ID given a different scope may refer to the same or (far more likely) a different subject. The same value provided to different relying parties MAY refer to different subjects, and indeed that is the primary distinguishing characteristic of this identifier Attribute.

The relationship between an asserting party and a scope is an arbitrary one and does not reflect any assumed relationship between a scope in the form of a domain name and a domain found in a given SAML entity identifier. This indirect rela~~la~~tionship is formally expressible in SAML metadata via the extension defined in Section 3.5.2.

A value MUST NOT be assigned to more than a single subject over its lifetime of use under any circumstances. The unique ID should therefore be constructed in a fashion that reduces the probability of non-technical or political considerations leading to a violation of this requirement, and any such violation should be treated as a potential security risk to the relying parties to which the value may have been given.

The value MUST NOT be mappable by a relying party into a non-pairwise identifier for the subject through ordinary effort. This precludes the degenerate case of providing a non-pairwise value to all relying parties for a given subject.

Relying parties should not treat this identifier as an email address for the subject as it is unlikely (though not precluded) for it to be valid for that purpose. Most organizations will find that existing email address values will not serve well as values for this Attribute.

The unique ID should not change as a result of a change to any other data associated with the subject (e.g., name, email address, age, organizational role).

Assuming a particular scope, a given subject MUST be identified with a different, though consistent, unique ID for each relying party to which a value is provided; however, the relationship between relying parties and SAML entities is not defined by this profile and is interpreted from the perspective of the asserting party. For example, in the context of the SAML Web Browser SSO profile [SAMLProf] it would be typical for an Identity Provider to base its notion of a relying party boundary on a single Service Provider's entity identifier, but that is not specifically required by this profile. The boundary MAY be larger or even smaller, at the Identity Provider's discretion or as addressed by additional profiles.

While it will be common in many deployments to limit values with a given scope to a single asserting party, this is ultimately left to the discretion of the relying party and the use case. It is unspecified by this profile whether a given value provided by two or more asserting parties correspond to the same subject. This would depend on out of band arrangements made between the parties. ~~But, in such cases, the "standard"~~If you want a relying party to understand that two or more asserting parties are referring to the same subject, using the general-purpose subject identifier defined in Section 3.3 is likely to be a much better choice.

3.4.3 Implementation Strategies

Supporting pairwise identifiers typically involves either the generation and storage of random values, or the computation of reproducible values that can be produced on demand but need not be stored. This profile does not require any specific approach, but implementers should be aware that some techniques for computing values may result in an unacceptable risk of case conflicts. For example, a salted hash over a seed identifier together with a relying party identifier produces a "safe" generated value, but becomes unsafe when encoded in Base64 [RFC4648] (and the allowable character set is defined in part to preclude this choice). However, encoding hashes in Base32 [RFC4648] is a safe choice, and the equals sign is included in the allowable character set to accommodate this.

3.4.4 Differences from "persistent" NameIDs

This Attribute is a direct replacement for the `urn:oasis:names:tc:SAML:2.0:nameid-format:persistent` NameID Format defined in SAML [SAML2Core]. There are obvious syntactic differences, in a deliberate attempt at simplification. The XML syntax and data "triple" are replaced with a simpler id/scope pair encoded into a string, and the awkward use of a pair of URIs to qualify the value is replaced with a simpler, shorter, and more flexible approach that more easily emulates the email address syntax required by many applications, and decouples identifier scoping from SAML entity naming.

One functional gap is the interoperable mechanism of SAML "affiliations" to group entities for the purpose of targeting pairwise identifiers to multiple Service Providers, which was ~~baked~~designed into the SAML protocol. It has been left out of this profile due to the general lack of adoption by implementers or deployers in the intervening years since the publication of the standard. Were there demand, it could be incorporated into a future revision.

3.4.5 Example

The following is an example of the SAML Attribute defined in this section:

```
<saml:Attribute Name="urn:oasis:names:tc:SAML:attribute:pairwise-id"
        NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri">
    <saml:AttributeValue>
HA2TKNZZGE2TOZDCGMZWKOLDHBQWIMBSGM4TGZBYGUYGINRQHAYTINBZGYZDOZBZMZRGKNZTME3TMN
BXGYYTIOBYGMYWKNLFMYYDAYY=@osu.edu
    </saml:AttributeValue>
</saml:Attribute>
```

3.5 Considerations for SAML Profiles

The Attributes defined in this profile are designed to be used in conjunction with any SAML profiles that support the use of SAML Attributes, though its predominant expected use is with the various SAML single sign-on profiles [SAML2Prof] such as the Web Browser SSO Profile and Enhanced Client or Proxy (ECP) Profile.

3.5.1 Requirements Signaling

In the event that SAML metadata [SAML2Meta] is used, a relying party MUST express its identifier requirements by including an `<mdattr:EntityAttribute>` extension [MetaAttr] in its metadata containing the following Attribute:

Name: `urn:oasis:names:tc:SAML:profiles:subject-id:req`

NameFormat: `urn:oasis:names:tc:SAML:2.0:attrname-format:uri`

This Attribute, MUST contain exactly one `<saml:AttributeValue>` element, whose `xsi:type` SHOULD be absent or if present MUST BE bound to the XML Schema `xsd:string` data type [XMLSCHEMA-2].

The value MUST be one of the following, signaling the corresponding requirement:

- `subject-id`
    - The relying party requires the standard identifier Attribute defined in Section 3.3.

- `pairwise-id`
  - The relying party requires the pair-wise identifier Attribute defined in Section 3.4.
- `none`
  - The relying party does not require any subject identifier and is designed to operate without a specific user identity (e.g., with authorization based on non-identifying data).
- `any`
  - The relying party will accept any of the identifier Attributes defined in this profile but requires at least one.

This profile does not define specific normative behavior on the part of asserting parties in response to this metadata, but it is expected that other profiles will do so in the future.

This profile does not provide (nor preclude) any guidance around the use of the `<md:RequestedAttribute>` element for signaling requirements, but notably it is impossible without additional specification work to reflect the semantics of the `any` value defined above using that mechanism.

3.5.2 Scope Filtering

A critical obligation of any federated relying party is to limit the ability of asserting parties to supply identifiers they are not authorized to assert. While this is commonly done in SAML based on the asserting party's entityID, that approach generally requires artificially combining an identifier's value with the entityID for storage and comparison. The Attributes defined in this specification include a scope expression in their values that makes this step unnecessary but introduce the need for a binding between scopes and asserting parties.

In the event that SAML metadata [SAML2Meta] is used, an asserting party MUST express the scope(s) within which it will issue subject identifiers by including one or more `<shibmd:Scope>` elements (defined below) in its metadata.

The `<shibmd:Scope>` element MUST appear within the `<md:Extensions>` element of an `<md:EntityDescriptor>` element or the `<md:Extensions>` element of an assertion-issuing role descriptor element (such as `<md:IDPSSODescriptor>` or `<md:AttributeAuthorityDescriptor>`). The use of the `<shibmd:Scope>` element outside of these contexts is undefined.

When a `<shibmd:Scope>` element appears in the `<md:Extensions>` element of an `<md:EntityDescriptor>` element it applies to all descendant role descriptor elements. That is to say, this usage is equivalent to putting an identical `<shibmd:Scope>` on every descendant role descriptor.

In processing the identifiers defined in this specification, the scope component is intended to be compared against the collection of scopes designated as permissible for the asserting party in its metadata. Any values whose scope is not permissible SHOULD be discarded, thus ensuring that all scoped identifier values accepted by the relying party and passed to an application will have permissible scopes.

The final arbiter of any such policy is the relying party, and metadata-based policy via this extension MAY be supplemented or overridden by local policy.

This profile does not mandate a particular exchange or trust model by which the metadata and its content are expected to be verified, but it is common for metadata containing this extension to come from a trusted third party able to independently validate an asserting party's right to the claimed scope(s).

For compatibility reasons, the matching between values of this extension and the scope component of the identifiers defined in this specification is done in a case-sensitive manner. To avoid unintentional mismatches, it is RECOMMENDED that scopes be expressed in lower case (both in this extension and in the values themselves, per Section 3.3.1).

Finally, note that the concept of scope and scope filtering need not be limited to the Attributes defined in this specification, but such applicability is outside the purview of this specification.

3.5.2.1 Element <shibmd:Scope>

This element extends the **xsd:string** schema type with the following attribute:

regexp [Optional]

Boolean regular expression indicator

Each <shibmd:Scope> element's text content identifies a permissible identifier scope for the issuing entity/role, per the definition of “scope” in Section 3.3.1.

If regexp is "false" or "0" or absent, the text content of the <shibmd:Scope> element is interpreted as the literal scope value (matched case-sensitively for compatibility reasons, see below).

If regexp is "true" or "1", the text content of the <shibmd:Scope> element is interpreted as specifying a regular expression (also see below).

The schema for the <shibmd:Scope> element is as follows:

```
<element name="Scope">
  <complexType>
    <simpleContent>
      <extension base="string">
        <attribute name="regexp" type="boolean" use="optional"
default="false"/>
      </extension>
    </simpleContent>
  </complexType>
</element>
```

3.5.2.2 Usage Considerations

Because this extension has an extensive history of use, its definition is not optimal and there are some important caveats.

Comparison of literal scope values expressed via this extension is defined to be case-sensitive, despite the overall rule for comparison of the Attributes defined in this specification as case-insensitive. This is for reasons of historical compatibility and generality, and is easily addressed by adhering to this specification's guidance to express scopes in lower-case.

The XML Schema definition of the <shibmd:Scope> element includes an explicit default value for the regexp attribute. One effect of this is that the meaning of an omitted regexp attribute will be different for a schema-validating processor than for one which does not schema-validate. If a document containing a <shibmd:Scope> element with an omitted regexp attribute is digitally signed, the signature value will therefore depend on whether the signer schema-validates, and validation of such a signature will only succeed if the validator has chosen to take the same approach.

To ensure interoperability between signers and validators no matter whether each schema validates or does not, it is therefore strongly RECOMMENDED that any <shibmd:Scope> element appearing in a metadata document that is to be digitally signed incorporate an explicit regexp attribute (i.e., regexp="false" or regexp="0" SHOULD always be used instead of an omitted regexp attribute).

Furthermore, great care should be taken in using regexp="true" as it is extremely easy to write regular expressions which match the desired patterns but also permit additional, sometimes surprising, matches. This can lead to an asserting party being permitted a wider range of scopes than intended. Common mistakes are not appropriately quoting meta-characters such as ".", and not appropriately anchoring the ends of the match.

Additionally, regular expressions are implemented with a degree of inconsistency in specifics and features and this extension does not include a formal reference to any single "standard" version of regular expressions because it would be impractical to force SAML implementations to follow only one.

As a result, deployments SHOULD avoid the use of regular expressions and implementations MAY omit support for this capability and reject its use. Its presence is again an issue of legacy compatibility ~~moreso~~more-so than current practice.

3.5.3 NameID Considerations

While the Attributes defined in this profile have as a goal the explicit replacement of the `<saml:NameID>` element as a means of subject identification, it is certainly possible to compose them with existing NameID usage provided the same subject is being identified. This can also serve as a migration strategy for existing applications.

Some profiles such as the Single Logout Profile [SAML2Prof] require the use of a `<saml:NameID>` element, which implies the earlier use of a NameID. In such cases, it is RECOMMENDED that the `urn:oasis:names:tc:SAML:2.0:nameid-format:transient` NameID Format be used.

This specification does not define any syntax by which the SAML Attributes defined within would be used directly within the NameID construct. Such use is discouraged, but is not precluded by this specification. In practice, the most appropriate mechanism to express any string-valued SAML Attribute in a `<saml:NameID>` element is to express the Attribute's `Name` as a `Format` and omit any qualifiers, and such an approach is safe to use with the Attributes defined in this specification.

3.5.4 Security Considerations

All identifiers have inherent and generally well-understood concerns; most applications traditionally associate users directly with resources, privileges, and/or data by uniquely identifying those users and remembering them during subsequent interactions. Federated protocols don't alter these concerns, but can complicate them because of the particular issues introduced by multiple asserting parties that may (but usually do not) share a common identifier namespace.

Applications not originally designed to support federation often treat each asserting party as a kind of silo of identity, and the identifiers used are inherently segregated by these silos such that global uniqueness (or lack thereof) is irrelevant. In such cases, the asserting party's own identifier acts as an implicit "scope" for all of the identifiers it asserts. In some cases, a lack of this implicit enforcement of scope has led to security vulnerabilities involving impersonation of users across asserting parties, demonstrating that, no matter what kind of identifier is used, some form of scoping of user identifiers is an absolute necessity in federated systems. This requirement is more obvious when applications are truly federated and combine identifiers from multiple asserting parties within a data set.

The identifier attributes defined in this specification contain an explicit scope as part of their syntax, providing globally uniqueness, but, more subtly, creating indirection between the scopes and the asserting party or parties that provide them. That is, the scope is explicit, but the relationship between that scope and an asserting party is indirect, at least when looking solely at the identifier. This indirection adds power, in that use cases involving identity linking between asserting parties become simpler to support, and it adds simplicity from the point of view of safe handling of identifier values since the scope is harder to "lose" or ignore. But this also adds complexity because a policy decision is required to authorize an asserting party to supply identifiers in a given scope.

As an example, consider an identifier such as "abcdef123@osu.edu"; SAML doesn't define anything in its core machinery that associates "osu.edu" with the Identity Provider representing The Ohio State University. Domain ownership proofs are of course a common and sensible practice to use to establish this association, but nothing in SAML specifies that, so it's an additional step and is not represented "in-band".

This specification does not impose a single such policy layer, but does standardize (in Section 3.5.2) a long-standing SAML metadata extension that associates authorized scope values with asserting parties. By using SAML metadata, the problem of self-assertion is addressed; if an asserting party were able to self-authorize its ability to supply an identifier in a different asserting party's scope, impersonation becomes easy. Communities that rely on curated, third-party sources of metadata have a vehicle for automating policy around scopes, and for off-loading domain/scope verification. Thus, use of metadata in this fashion and use of scoped identifiers become mutually reinforcing.

4 Conformance

4.1 Conformance Clause 1: Asserting Party Implementations

An asserting party implementation conforms to this specification if it can be configured to produce both identifier Attributes conforming to the normative requirements in Sections 3.3 and 3.4.

If the asserting party implementation provides a mechanism for generation and/or publication of SAML metadata, then it MUST support the inclusion of the extension defined in Section 3.5.2.

4.2 Conformance Clause 2: Relying Party Implementations

A relying party implementation conforms to this specification if it can be configured to consume neither, either, and both of the two identifier Attributes conforming to the normative requirements in Sections 3.3 and 3.4.

If the relying party implementation provides a mechanism for generation and/or publication of SAML metadata, then it MUST support the inclusion of the extension defined in Section 3.5.1.

If the relying party supports the consumption of SAML metadata, then it MUST support configuring its acceptance of values of the Attributes defined in this specification based on authorization of their scopes via the extension defined in Section 3.5.2.

Appendix A Acknowledgments

The following individuals have participated in the creation of this specification and are gratefully acknowledged:

Scott Cantor, Internet2
Thomas Hardjono, MIT
Mohammad Jafari, Veterans Health Administration
Hal Lockhart, Oracle Corporation
Madalina Sultan, Connectis

Contributors to the InCommon Deployment Profile Working Group
Past contributors to the Shibboleth Project

Appendix B Revision History

Revision	Date	Editor	Changes Made
WD 01	30 Aug 2017	Scott Cantor	Initial draft
WD 02	13 Sep 2017	Scott Cantor	Added considerations for other profiles
WD 03	15 Sep 2017	Scott Cantor	Added hyphen as legal character in unique ID
WD 04	1 Feb 2018	Scott Cantor	Many nits, missing references, clarifying changes in response to public review
WD 05	3 Jul 2018	Scott Cantor	Second public review updates
WD 06	5 Sep 2018	Scott Cantor	Expansion of scope to include, umm, Scope
WD 07	16 Nov 2018	Scott Cantor	Editorial nits and corrections for final vote