Network Working Group J. Richer, Ed.
Internet-Draft Bespoke Engineering
Intended status: Experimental L. Johansson
Expires: July 20, 2017 Swedish University Network
January 16, 2017

# Vectors of Trust
## draft-richer-vectors-of-trust-04

Abstract

This document defines a mechanism for describing and signaling several aspects that are used to calculate trust placed in a digital identity transaction.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

Table of Contents

## 1. Introduction

This document defines a mechanism for measuring and signaling several aspects of digital identity and authentication transactions that are used to determine a level of trust in that transaction. In the past, there have been two extremes of communicating authentication transaction information.

At one extreme, all attributes can be communicated with full provenance and associated trust markings. This approach seeks to create a fully-distributed attribute system to support functions such as attribute based access control (ABAC). These attributes can be used to describe the end user, the identity provider, the relying party, or even the transaction itself. While the information that can be expressed in this model is incredibly detailed and robust, the complexity of such a system is often prohibitive to realize, especially across security domains. In particular, a large burden is placed on relying parties needing to process the sea of disparate attributes when making a security decision.

At the other extreme there are systems that collapse all of the attributes and aspects into a single scalar value that communicates, in sum, how much a transaction can be trusted. The NIST special publication 800-63 [SP-800-63-2] version 2 defines a linear scale Level of Assurance (LoA) measure that combines multiple attributes about an identity transaction into such a single measure. While this definition was originally narrowly targeted for a specific set of government use cases, the LoA scale appeared to be applicable with a wide variety of authentication scenarios in different domains. This has led to a proliferation of incompatible interpretations of the same scale in different contexts, preventing interoperability between each LoA definition in spite of their common measurement. LoA is artificially limited due to the original goal of creating a single linear scale. Since identity proofing strength increases linearly along with credential strength in the LoA scale, this scale is too limited for describing many valid and useful forms of an identity transaction that do not fit the government's original model. For example, an anonymously assigned hardware token can be used in cases where the real world identity of the subject cannot be known for privacy reasons, but the credential itself can be highly trusted. This is in contrast with a government employee accessing a government system, where the identity of the individual would need to be highly proofed and strongly credentialed at the same time.

The Vectors of Trust (VoT) effort seeks to find a balance between these two extremes by creating a data model that combines attributes of the user and aspects of the authentication context into several values that can be communicated separately but in parallel with each

other. This approach is both coarser grained than the distributed attributes model and finer grained than the single scalar model, with the hope that it is a viable balance of expressibility and processability. Importantly, these three levels of granularity can be mapped to each other. The information of several attributes can be folded into a vector component, while the vector itself can be folded into an assurance category. As such, the vectors of trust seeks to complement, not replace, these other identity and trust mechanisms in the larger identity ecosystem while providing a single value for RPs to process.

## 1.1. Terminology

Identity Provider (IdP) A system that manages identity information and is able to assert this information across the network through an identity API.

Identity Subject The person (user) engaging in the identity transaction, being identified by the identity provider and identified to the relying party.

Primary Credential The means used by the identity subject to authenticate to the identity provider.

Federated Credential The assertion presented by the IdP to the RP across the network to authenticate the user.

Relying Party (RP) A system that consumes identity information from an IdP for the purposes of authenticating the user.

Trust Framework A document containing business rules and legal clauses that defines how different parties in an identity transaction may act.

Trustmark A verifiable attestation that a party has proved to follow the constraints of a trust framework.

Trustmark Provider A system that issues and provides verification for trustmarks.

Vector A multi-part data structure, used here for conveying information about an authentication transaction.

Vector Component One of several constituent parts that make up a vector.

### 1.2. An Identity Model

This document assumes the following model for identity based on identity federation technologies:

The identity subject (also known as the user) is associated with an identity provider which acts as a trusted third party on behalf of the user with regard to a relying party by making identity assertions about the user to the relying party.

The real-world person represented by the identity subject is in possession of a primary credential bound to the identity subject by the identity provider (or an agent thereof) in such a way that the binding between the credential and the real-world user is a representation of the identity proofing process performed by the identity provider (or an agent thereof) to verify the identity of the real-world person. This is all carried by an identity assertion across the network to the relying party during the authentication transaction.

### 1.3. Component Architecture

The term Vectors of Trust is based on the mathematical construct of a vector, which is defined as an item composed of multiple independent values.

An important goal for this work is to balance the need for simplicity (particularly on the part of the relying party) with the need for expressiveness. As such, this vector construct is designed to be composable and extensible.

All components of the vector construct MUST be orthogonal such that no aspect of a component overlaps an aspect of another component, as much as is possible.

## 2. Component Definitions

This specification defines four orthogonal components: identity proofing, primary credential usage, primary credential management, and assertion presentation. These dimensions MUST be evaluated by the RP in the context of a trust framework and SHOULD be combined with other information when making a trust and authorization decision.

This specification also defines values for each component to be used in the absence of a more specific trust framework in Section 3. It is expected that trust frameworks will provide context, semantics, and mapping to legal statutes and business rules for each value in

each component. Consequently, a particular vector value can only be compared with vectors defined in the same context. The RP MUST understand and take into account the trust framework context in which a vector is being expressed in order for it to be processed securely.

Each component is identified by a demarcator consisting of a single uppercase ASCII letter in the range "[A-Z]". The demarcator SHOULD reflect the category with which it is associated in a natural manner. Demarcators for components MUST be registered as described in Section 9. It is anticipated that trust framework definitions will use this registry to define specialized components, though it is RECOMMENDED that trust frameworks re-use existing components wherever possible.

The value for a given component within a vector of trust is defined by its demarcator character followed by a single digit or lowercase ASCII letter in the range "[0-9a-z]". Categories which have a natural ordering SHOULD use digits, with "0" as the lowest value. Categories which do not have a natural ordering, or which can have an ambiguous ordering, SHOULD use letters. Categories MAY use both letter style and number style value indicators. For example, a category could define "0" as a special "empty" value while using letters such as "a", "b", "c" for normal values can to differentiate between these types of options.

Regardless of the type of value indicator used, the values assigned to each component of a vector MUST NOT be assumed always to have inherent ordinal properties when compared to the same or other components in the vector space. In other words, "1" is different from "2", but it is dangerous to assume that "2" is always better than "1" in a given transaction.

## 2.1. Identity Proofing

The Identity Proofing dimension defines, overall, how strongly the set of identity attributes have been verified and vetted. In other words, this dimension describes how likely it is that a given digital identity transaction corresponds to a particular (real-world) identity subject.

This dimension SHALL be represented by the "P" demarcator and a single-character level value, such as "P0", "P1", etc. Most definitions of identity proofing will have a natural ordering, as more or less stringent proofing can be applied to an individual. In such cases it is RECOMMENDED that a digit style value be used for this component.

## 2.2. Primary Credential Usage

The primary credential usage dimension defines how strongly the primary credential can be verified by the IdP. In other words, how easily that credential could be spoofed or stolen.

This dimension SHALL be represented by the "C" demarcator and a single-character level value, such as "Ca", "Cb", etc. Most definitions of credential usage will not have an overall natural ordering, as there may be several equivalent classes described within a trust framework. In such cases it is RECOMMENDED that a letter style value be used for this component. Multiple credential usage factors MAY be communicated simultaneously, such as when Multi-Factor Authentication is used.

## 2.3. Primary Credential Management

The primary credential management dimension conveys information about the expected lifecycle of the primary credential in use, including its binding, rotation, and revocation. In other words, the use and strength of policies, practices, and security controls used in managing the credential at the IdP and its binding to the intended individual.

This dimension SHALL be represented by the "M" demarcator and a single-character level value, such as "Ma", "Mb", etc. Most definitions of credential management will not have an overall natural ordering, though there can be preference and comparison between values in some circumstances. In such cases it is RECOMMENDED that a letter style value be used for this component.

## 2.4. Assertion Presentation

The Assertion Presentation dimension defines how well the given digital identity can be communicated across the network without information leaking to unintended parties, and without spoofing. In other words, this dimension describes how likely it is that a given digital identity was actually asserted by a given identity provider for a given transaction. While this information is largely already known by the RP as a side effect of processing an identity assertion, this dimension is still very useful when the RP requests a login (Section 5) and when describing the capabilities of an IdP (Section 7).

This dimension SHALL be represented by the "A" demarcator and a level value, such as "Aa", "Ab", etc. Most definitions of assertion presentation will not have an overall natural ordering. In such

cases, it is RECOMMENDED that a letter style value be used for this component.

## 3. Vectors of Trust Initial Component Value Definitions

This specification defines the following general-purpose component definitions, which MAY be used when a more specific set is unavailable. These component values are referenced in a trustmark definition defined by [[ this document URL ]].

It is anticipated that trust frameworks and specific applications of this specification will define their own component values. In order to simplify processing by RPs, it is RECOMMENDED that trust framework definitions carefully define component values such that they are mutually exclusive or subsumptive in order to avoid repeated vector components where possible.

### 3.1. Identity Proofing

The identity proofing component of this vector definition represents increasing scrutiny during the proofing process. Higher levels are largely subsumptive of lower levels, such that "P2" fulfills requirements for "P1", etc.

P0 No proofing is done, data is not guaranteed to be persistent across sessions

P1 Attributes are self-asserted but consistent over time, potentially pseudonymous

P2 Identity has been proofed either in person or remotely using trusted mechanisms (such as social proofing)

P3 There is a binding relationship between the identity provider and the identified party (such as signed/notarized documents, employment records)

### 3.2. Primary Credential Usage

The primary credential usage component of this vector definition represents distinct categories of primary credential that MAY be used together in a single transaction.

C0 No credential is used / anonymous public service

Ca Simple session cookies (with nothing else)

Cb Known device

Cc Shared secret such as a username and password combination

Cd Cryptographic proof of key possession using shared key

Ce Cryptographic proof of key possession using asymmetric key

Cf Sealed hardware token / trusted biometric / TPM-backed keys

### 3.3. Primary Credential Management

The primary credential management component of this vector definition represents distinct categories of management that MAY be considered separately or together in a single transaction. Many trust framework deployments MAY use a single value for this component as a baseline for all transactions and thereby omit it.

Ma Self-asserted primary credentials (user chooses their own credentials and must rotate or revoke them manually) / no additional verification for primary credential issuance or rotation

Mb Remote issuance and rotation / use of backup recover credentials (such as email verification) / deletion on user request

Mc Full proofing required for each issuance and rotation / revocation on suspicious activity

### 3.4. Assertion Presentation

The assertion presentation component of this vector definition represents distinct categories of assertion which are RECOMMENDED to be used in a subsumptive manner but MAY be used together.

Aa No protection / unsigned bearer identifier (such as a session cookie in a web browser)

Ab Signed and verifiable assertion, passed through the user agent (web browser)

Ac Signed and verifiable assertion, passed through a back channel

Ad Assertion encrypted to the relying parties key and audience protected

## 4. Communicating Vector Values to RPs

A vector of trust is designed to be used in the context of an identity and authentication transaction, providing information about the context of a federated credential. The vector therefore needs to be able to be communicated in the context of the federated credential in a way that is strongly bound to the assertion representing the federated credential.

This vector has several requirements for use.

- All applicable vector components and values need to be combined into a single vector.

- The vector can be communicated across the wire unbroken and untransformed.

- All vector components need to remain individually available, not "collapsed" into a single value.

- The vector needs to be protected in transit.

- The vector needs to be cryptographically bound to the assertion which it is describing.

These requirements lead us to defining a simple string-based representation of the vector that can be incorporated within a number of different locations and protocols without further encoding.

### 4.1. On the Wire Representation

The vector MUST be represented as a period-separated ('.') list of vector components, with no specific order. A vector component type MAY occur multiple times within a single vector, with each component separated by periods. Multiple values for a component are considered a logical AND of the values. A specific value of a vector component MUST NOT occur more than once in a single vector. That is, while "Cc.Cd" is a valid vector, "Cc.Cc" is not.

Vector components MAY be omitted from a vector. No holding space is left for an omitted vector component. If a vector component is omitted, the vector is making no claim for that component. This MAY be distinct from a specific component value stating that a component was not used.

Vector values MUST be communicated along side of a trustmark definition to give the components context. A vector value without context is unprocessable, and vectors defined in different contexts

are not directly comparable as whole values. Different trustmarks MAY re-use component definitions (including their values), allowing comparison of individual components across contexts without requiring complete understanding of all aspects of a context. The proper processing of such cross-context values is outside the scope of this specification.

For example, the vector value "P1.Cc.Ab" translates to "pseudonymous, proof of shared key, signed browser-passed verified assertion, and no claim made toward credential management" in the context of this specification's definitions (Section 3). The vector value of "Cb.Mc.Cd.Ac" translates to "known device, full proofing require for issuance and rotation, cryptographic proof of possession of a shared key, signed back-channel verified assertion, and no claim made toward identity proofing" in the same context.

## 4.2. In OpenID Connect

In OpenID Connect [OpenID], the IdP MUST send the vector as a string within the "vot" (vector of trust) claim in the ID token. The trustmark (Section 6) that applies to this vector MUST be sent as an HTTPS URL in the "vtm" (vector trust mark) claim to provide context to the vector.

For example, the body of an ID token claiming "pseudonymous, proof of shared key, signed back-channel verified token, and no claim made toward credential management" could look like this JSON object payload of the ID token.


```
{
    "iss": "https://idp.example.com/",
    "sub": "jondoe1234",
    "vot": "P1.Cc.Ac",
    "vtm": "https://trustmark.example.org/trustmark/idp.example.com"
}
```


The body of the ID token is signed and optionally encrypted using JOSE, as per the OpenID Connect specification. By putting the "vot" and "vtm" values inside the ID token, the vector and its context are strongly bound to the federated credential represented by the ID token.

## 4.3. In SAML

In SAML, a vector is communicated as an AuthenticationContextDeclRef. A vector is represented by prefixing it with the urn urn:ietf:param:[TBD] to form a full URN. The AuthenticationContextDeclaration corresponding to a given vector is a

AuthenticationContextDeclaration element containing an Extension element with components of the vector represented by the following XML schema:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
    targetNamespace="urn:ietf:param:[TBD]:schema"
    xmlns:xs="http://www.w3.org/2001/XMLSchema">
   <xs:element name="Vector">
      <xs:annotation>
         <xs:documentation>This represents a set of vector components.</
xs:documentation>
      </xs:annotation>
      <xs:simpleType>
         <xs:restriction base="xsd:token">
            <xs:pattern value="([A-Z][a-z0-9])(\.[A-Z][a-z0-9])*"/>
         </xs:restriction>
      </xs:simpleType>
   </xs:element>
</xs:schema>
```

For instance the vector P1.Cc.Ac is represented by the AuthenticationContextDeclRef URN urn:ietf:param:[TBD]:P1.Cc.Ac (or urn:ietf:param:[TBD]:Cc.P1.Ac or ...) which corresponds to the following AuthenticationContextDeclaration:

```
<?xml version="1.0" encoding="UTF-8"?>
<AuthenticationContextDeclaration xmlns="urn:oasis:names:tc:SAML:2.0:ac">
   <Extension>
      <vot:Vector>P1.Cc.Ac</vot:Vector>
   </Extension>
</AuthenticationContextDeclaration>
```

## 5. Requesting Vector Values

In some identity protocols, the RP can request that particular vector components be applied to a given identity transaction. Using the same syntax as defined in Section 4.1, an RP can indicate that it desires particular aspects be present in the authentication. Processing and fulfillment of these requests are in the purview of the IdP and details are outside the scope of this specification.

### 5.1. In OpenID Connect

In OpenID Connect [OpenID], the client MAY request a set of acceptable VoT values with the "vtr" (vector of trust request) claim request as part of the Request Object. The value of this field is an array of JSON strings, each string identifying an acceptable set of vector components.  The component values within each vector are ANDed together while the separate vectors are ORed together. For example, a list of vectors in the form "["P1.Cb.Cc.Ab", "Ce.Ab"]" is stating that either the full set of "P1 AND Cb AND Cc AND Ab" simultaneously OR the set of "Ce AND Ab" simultaneously are acceptable to this RP for this transaction.

Vector request values MAY omit components, indicating that any value is acceptable for that component category, including omission of that component in the response vector.

The mechanism by which the IdP processes the "vtr" and maps that to the authentication transaction are out of scope of this specification.

### 5.2. In SAML

In SAML (Section 4.3) the client can request a set of acceptable VoT values by including the corresponding AuthenticationContextDeclRef URIs together with an AuthenticationContextClassRef corresponding to the trust mark (cf below). The processing rules in Section 4.3 apply.

## 6. Trustmark

When an RP receives a specific vector from an IdP, it needs to make a decision to trust the vector within a specific context. A trust framework can provide such a context, allowing legal and business rules to give weight to an IdP's claims. A trustmark is a verifiable claim to conform to a specific component of a trust framework, such as a verified identity provider. The trustmark conveys the root of trustworthiness about the claims and assertions made by the IdP, including the vector itself.

The trustmark MUST be available from an HTTPS URL served by the trust framework provider. The contents of this URL are a JSON [RFC7159] document with the following fields:

idp The issuer URL of the identity provider that this trustmark pertains to. This MUST match the corresponding issuer claim in the identity token, such as the OpenID Connect "iss" field. This MUST be an HTTPS URL.

trustmark_provider The issuer URL of the trustmark provider that issues this trustmark. The URL that a trustmark is fetched from MUST start with the "iss" URL in this field. This MUST be an HTTPS URL.

P Array of strings containing identity proofing values for which the identity provider has been assessed and approved.

C Array of strings containing primary credential usage values for which the identity provider has been assessed and approved.

M Array of strings containing primary credential management values for which the identity provider has been assessed and approved.

A Array of strings containing assertion strength values for which the identity provider has been assessed and approved.

Additional vector component values MUST be listed in a similar fashion using their demarcator.

For example, the following trustmark provided by the trustmark.example.org organization applies to the idp.example.org identity provider:

```
{
  "idp": "https://idp.example.org/",
  "trustmark_provider": "https://trustmark.example.org/",
  "P": ["P0", "P1"],
  "C": ["C0", "Ca", "Cb"],
  "M": ["Mb"],
  "A": ["Ab", "Ac"]
}
```

An RP wishing to check the claims made by an IdP can fetch the information from the trustmark provider about what claims the IdP is allowed to make in the first place and process them accordingly. The RP MAY cache the information returned from the trustmark URL.

The operational aspects of the IdP MAY be included in the trustmark definition. For example, if a trustmark can indicate that an IdP uses multiple redundant hosts, encrypts all data at rest, or other operational security mechanisms that affect the trustworthiness of assertions made by the IdP. The definition of these additional aspects is outside the scope of this specfication.

The means by which the RP decides which trustmark providers it trusts is out of scope for this specification and is generally configured out of band.

Though most trust frameworks will provide a third-party independent verification service for components, an IdP MAY host its own trustmark. For example, a self-hosted trustmark would look like:

```
{
  "idp": "https://idp.example.org/",
  "trustmark_provider": "https://idp.example.org/",
  "P": ["P0", "P1"],
  "C": ["C0", "Ca", "Cb"],
  "M": ["Mb"],
  "A": ["Ab", "Ac"]
}
```


## 7. Discovery

The IdP MAY list all of its available trustmarks as part of its discovery document, such as the OpenID Connect Discovery server configuration document. In this context, trustmarks are listed in the "trustmarks" element which contains a single JSON [RFC7159] object. The keys of this JSON object are trustmark provider issuer URLs and the values of this object are the corresponding trustmark URLs for this IdP.


```
{
    "iss": "https://idp.example.org/",
    "trustmark": {
         "https://trustmark.example.org/": "https://trustmark.example.org/
trustmark/idp.example.org/"
    }
}
```


## 8. Acknowledgements

The authors would like to thank the members of the Vectors of Trust mailing list in the IETF for discussion and feedback on the concept and document, and the members of the ISOC Trust and Identity team for their support.

## 9. IANA Considerations

This specification creates one registry and registers several values into an existing registry.

### 9.1. Vector Of Trust Components Registry

The Vector of Trust Components Registry contains the definitions of vector components and their associated demarcators.

- Demarcator Symbol: P

- Description: Identity proofing

- Document: [[ this document ]]

- o Demarcator Symbol: C
- o Description: Primary credential usage
- o Document: [[ this document ]]
- o Demarcator Symbol: M
- o Description: Primary credential management
- o Document: [[ this document ]]
- o Demarcator Symbol: A
- o Description: Assertion presentation
- o Document: [[ this document ]]

## 9.2. Additions to JWT Claims Registry

This specification adds the following values to the JWT Claims Registry.

- o Claim name: vot
- o Description: Vector of Trust value
- o Document: [[ this document ]]
- o Demarcator Symbol: vtm
- o Description: Vector of Trust Trustmark
- o Document: [[ this document ]]
- o Demarcator Symbol: vtr
- o Description: Vector of Trust Request
- o Document: [[ this document ]]

## 10. Security Considerations

The vector of trust value MUST be cryptographically protected in transit, using TLS as described in [BCP195]. The vector of trust value MUST be associated with a trustmark marker, and the two MUST be carried together in a cryptographically bound mechanism such as a

signed identity assertion. A signed OpenID Connect ID Token and a signed SAML assertion both fulfil this requirement.

The VoT framework provides a mechanism for describing and conveying trust information. It does not define any policies for asserting the values of the vector, nor does it define any policies for applying the values of a vector to an RP's security decision process. These policies MUST be agreed upon by the IdP and RP, and they SHOULD be expressed in detail in an associated trust framework.

## 11. Privacy Considerations

By design, vector of trust values contain information about the user's authentication and associations that can be made thereto. Therefore, all aspects of a vector of trust contain potentially privacy-sensitive information and MUST be guarded as such. Even in the absence of specific attributes about a user, knowledge that the user has been highly proofed or issued a strong token could provide more information about the user than was intended. It is RECOMMENDED that systems in general use the minimum vectors applicable to their use case in order to prevent inadvertent information disclosure.

## 12. References

### 12.1. Normative References

[OpenID] Sakimura, N., Bradley, J., and M. Jones, "OpenID Connect Core 1.0", November 2014.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC7159] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", RFC 7159, DOI 10.17487/RFC7159, March 2014, <http://www.rfc-editor.org/info/rfc7159>.

### 12.2. Informative References

[BCP195] Sheffer, Y., Holz, R., and P. Saint-Andre, "Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)", BCP 195, RFC 7525, DOI 10.17487/RFC7525, May 2015, <http://www.rfc-editor.org/info/bcp195>.

[SP-800-63-2]
, , , , , , and , "Electronic Authentication Guideline", August 2013.

## Appendix A. Document History

-04

- Updated SAML example to be consistent.

-03

- Clarified language of LoA's in introduction.

- Added note on operational security in trustmarks.

- Removed empty sections and references.

-02

- Converted C, M, and A values to use letters instead of numbers in examples.

- Updated SAML to a structured example pending future updates.

- Defined guidance for when to use letters vs. numbers in category values.

- Restricted category demarcators to uppercase and values to lowercase and digits.

- Applied clarifying editorial changes from list comments.

- 01

- Added IANA registry for components.

- Added preliminary security considerations and privacy considerations.

- Split "credential binding" into "primary credential usage" and "primary credential management".

- 00

- Created initial IETF drafted based on strawman proposal discussed on VoT list.

- o Split vector component definitions into their own section to allow extension and override.

- o Solidified trustmark document definition.

Authors' Addresses

Justin Richer (editor)
Bespoke Engineering

Email: ietf@justin.richer.org

Leif Johansson
Swedish University Network
Thulegatan 11
Stockholm
Sweden

Email: leifj@sunet.se
URI: http://www.sunet.se