Network Working Group	M. Wasserman
Internet-Draft	Painless Security
Intended status: Standards Track	H. Tschofenig
Expires: May 02, 2012	Nokia Siemens Networks
	October 30, 2011

Multihop Federations for Application Bridging for Federation Beyond the Web (ABFAB)
draft-mrw-abfab-multihop-fed-02.txt

Abstract

This document describes a mechanism for establishing trust across a multihop federation within the Application Bridging for Federation Beyond the Web (ABFAB) framework.
This document introduces a new entity, the Trust Router. Trust Routers exchange information about the availability of Trust Paths across a multihop federation. They can be queried by a Relying Party to obtain the best Trust Path to reach an Identity Provider. They also provide temporary identities that can be used by a Relying Party to traverse a Trust Path.

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction

This document describes a mechanism for establishing trust across a multihop federation within the Application Bridging for Federation Beyond the Web (ABFAB) framework [I-D.lear-abfab-arch].
This document introduces a new ABFAB entity, the Trust Router. Trust Routers exchange information about the availability of Trust Paths across a multihop federation. ABFAB entity, the Trust Router. These paths are used by RPs to contruct transitive trust chains across a federation to a AAA Server (a RADIUS, RadSec or Diameter Server) within a target IdP.
A Trust Path consists of one or more Trust Links. A Trust Link is an assertion that a specific Trust Router is capable of providing temporary identies that can be used to access another entity in the ABFAB system.

At this point, we anticipate that there will be two types of Trust Links in ABFAB: a Trust Link that indicates that one Trust Router can be used to reach another Trust Router, and a Trust Link that indicates that a Trust Router can be used to reach a AAA Server. The first type (Trust Router Links) are shown as A->B(T), which indicates that the Trust Router in realm A can create identities to reach the trust router in Realm B. The second type (AAA Links) are shown as A->B(R), to indicate that a trust router in Realm A can be used to reach a AAA Server in Realm B.

Trust Routers exchange information about available Trust Links within a federation, and each Trust Router maintains a tree of available paths to reach all of the IdPs within the federation that can be reached from the local realm of the Trust Router.

When an RP receives a request from a party within a realm that not known directly to the RP, the RP will query its local Trust Router to obtain the best Trust Path to reach that IdP. Note that we use the term 'best' here to highlight that there may well be multiple paths to reach an IdP from a given RP, and the selection of the 'best' path may involve several factors in addition to the length of the path, such as security and privacy practices, or monetary costs.

The RP will traverse the Trust Path obtained from it's local Trust Router. At each step, the RP will request a temporary identity to access the next step in the Trust Path, constructing a transitive chain of trust to a AAA Server within the target IdP.

To summarize, the Trust Router performs three functions:

* Trust Routers peer with other Trust Routers to exchange information about available Trust Links, and Trust Paths. This information is exchanged between Trust Routers using the Trust Router Protocol. The Trust Router Protocol is described in more detail in [I-D.mrw-abfab-trust-router].

* Trust Routers respond to queries from Relying Parties to make information about Trust Paths available. This exchange is referred to as a Trust Path Query Protocol, which is described in Section 6.

* To follow the Trust Path across a federation, the RP will use KNP to ask each Trust Router along the path to provision a temporary identity that can be used to gain access to the next step in the path. This mechanism is called a Temporary Identity Request, which is described in [I-D.howlett-radsec-knp].

2. Terminology

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 RFC 2119 [RFC2119].

This document introduces the following terms:

Trust Router:

This is a logical ABFAB entity that exchanges information about Trust Paths that Relying Parties can use to create transtitive chains of trust across multihop ABFAB federations.

Trust Link:

A Trust Link is an assertion that a given Trust Router is capable of providing a temporary identity to communicate with another ABFAB entity (either another Trust Router, or a AAA Server within an IdP).

Trust Path:

A Trust Path is a concatenation of Trust Links that can be used by an RP to contruct a transitive trust chain across a federation to a target IdP.

Trust Router Protocol:

The Trust Router Protocol is the mechanism used by two Trust Routers to exchange information about Trust Links and Trust Paths.

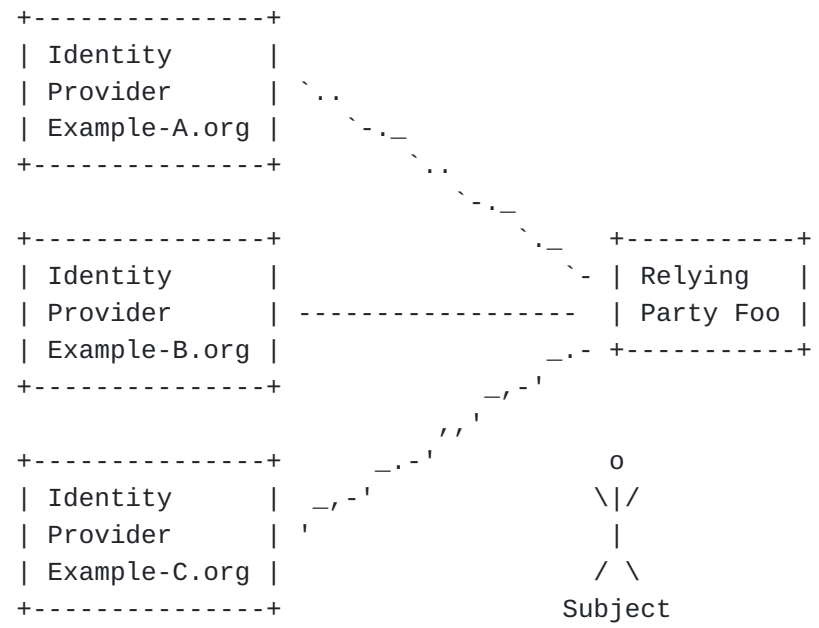
Community of Interest:

A Community of Interest defines a group of Services and IdPs that have agreed to cooperate to provide access to a specific set of services only to those users within a particular community. Communities of Interest can be layered on top of the base Trust Router infrastructure to allow selected access to IdPs that have joined a specific group, or agreed to a set of community-specific policies.

The terms Identity Provider (IdP), Relying Party (RP), Subject, and Federation are used as defined in [I-D.lear-abfab-arch].

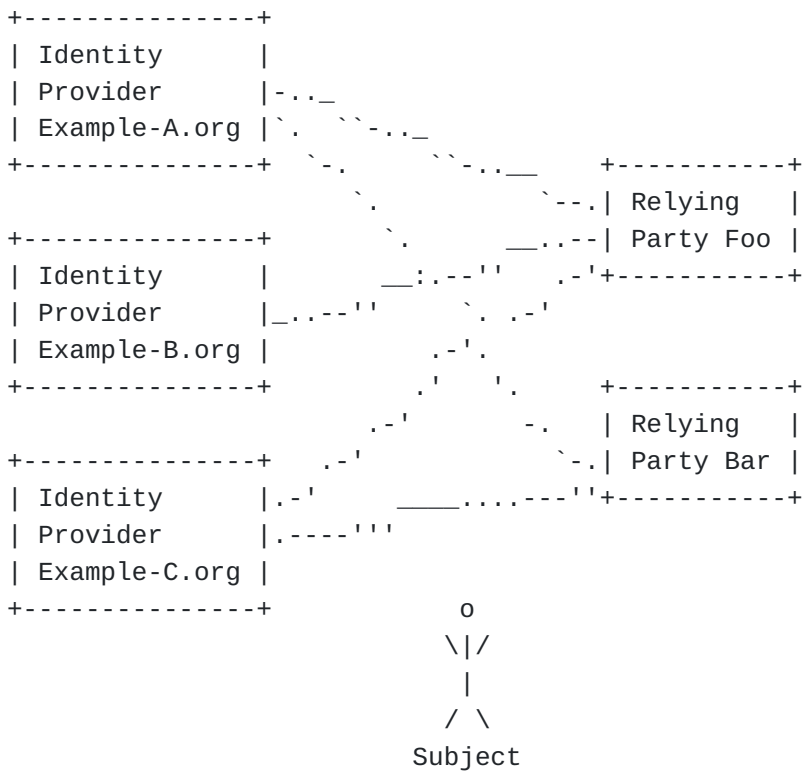
3. Motivation

Figure 1 shows an example federation where the Relying Party Foo, has established relationships with various Identity Providers.

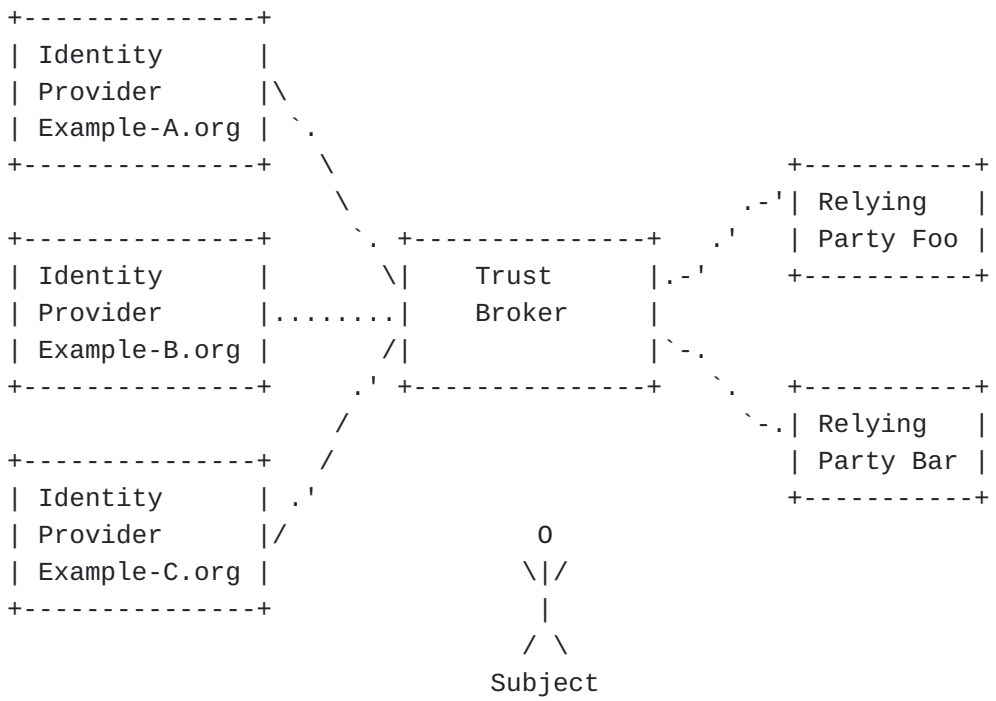


When an RP receives a request to access a protected resource (or requires authentication and authorization for other purposes) the request includes a realm name that indicates the IdP the Subject has selected for this exchange. Offering the Subject the ability to choose among many different IdPs is necessary because a Subject may have, and want to maintain, uncorrelated identities in several different realms within a single federation (i.e. work, school, social networking, etc.). However, this also places a burden on the RPs to establish and maintain business agreements and exchange security credentials with a potentially large number of Identity Providers.

In order for a single-hop federation to function, each IdP needs to maintain business agreements and exchange credentials with every RP that its Subjects are authorized to access. Figure 2, shows the likely outcome, which is that a single-hop federation will come to resemble a dense mesh topology.



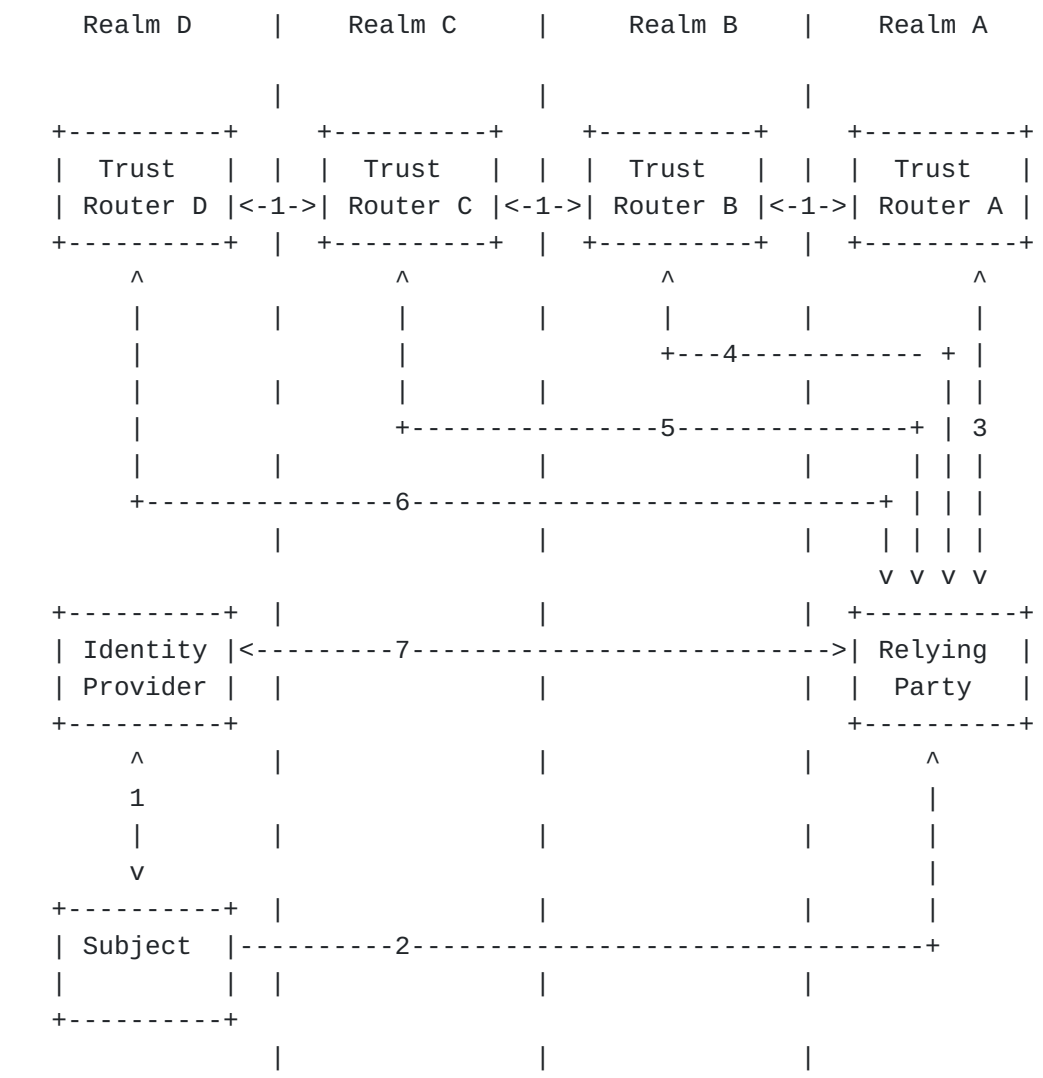
As discussed in section 2.1.1 of [I-D.lear-abfab-arch], as the number of organizations involved in a ABFAB federation increase, static configuration may not scale sufficiently. Also, using a Trust Broker to establish keys between entities near the RP and entities near the IDP with improve the security and privacy of an ABFAB federation. Figure 3 shows the structure of a federation where each IdP and RP has a single connection to the Trust Router infrastructure.



To improve the operational scalability and security of large ABFAB federations, this document proposes a Trust Broker solution consisting of of a set of Trust Routers, as described in this document, and the Key Negotiation Protocol (KNP), as described in [I-D.howlett-radsec-knp].

4. Multihop Federation Example

The diagram below shows an example of a successful exchange in a multihop federation using the Trust Router Protocol and KNP:



A multihop federation exchange matching the above diagram can be summarized as follows:

1. We start with a single federation including four realms, each containing a single Trust Router. The Trust Routers are peered, such that their interconnections form a multihop federation.

2. A Subject (with an identity in Realm D) attempts to access a service provided by a Relying Party in Realm A.

3. The Relying Party does not have direct access to a AAA Server in Realm D that it can use to authenticate the Subject, so it asks its local Trust Router for a Trust Path to reach Realm D. The Trust Router in Realm A returns the path A->B(T)->C(T)->D(T)->D(R), which indicates that the Relying Party should use the Trust Routers in Realms B, C and D to reach a AAA Server in Realm D, which could then be used to authenticate the Subject.

4. The Relying Party contacts a Trust Router in Realm B (using its permanent identity in Realm A), and requests the creation of a temporary identity that can be used to communicate with the Trust Router in Realm C.

5. The Relying Party then contacts the Trust Router in Realm C (using the temporary identity returned in the previous step), and asks for a temporary identity that can be used to communicate with the Trust Router in realm D.

6. The Relying Party then contacts the Trust Router in Realm D (using the temporary identity returned in the previous step), and asks the Trust Router to provision an identity that it can use to speak to the AAA Server in Realm D (which is part of Realm D's Identity Provider).

7. At this point, the Relying Party can reach the Subject's Identity provider, and the rest of the ABFAB exchange can continue, as described in [I-D.lear-abfab-arch].

5. Trust Router Protocol

Trust Routers use the Trust Router Protocol to exchange information about available Trust Links, and Trust Paths across a federation. The Trust Router Protocol differs from an Internet Routing Protocol in a couple of important ways:

*Trust Links are unidirectional. It can not be assumed that the fact that a Trust Router in Realm A is authorized to create temporary identities to access a Trust Router in realm B, that the opposite is also true (A -> B(T) does not imply B->A(T)).

*Realm names are not necessarily hierarchical. Although aggregation might be possible as a later optimization, the ability to aggregate realm names based on shared roots is not currently assumed.

In addition to the existence of the links themselves, Trust Links have a set of associated attributes that can be used for filtering and tree computation, including:

*The cost of the link.

*Any security and privacy characteristics associated with the link.

*Information indicating how/if the link should be propagated across the federation.

Current thinking is that we will use a BGP-based algorithm for computation of the local tree at each Trust Router, and that we will

communicate a similar set of information between Trust Routers as would be communicated between Internet Routers running BGP.

6. Trust Path Query

A Trust Path Query is generated by a RP to request a Trust Path to reach a specific realm within a given Community of Interest. If possible, the Trust Router will reply with a Trust Path that consists of zero or more Trust Router steps and ends with a AAA Server (or a path of multiple AAA Servers) within the IdP for the indicated realm.
The Trust Path Query is initiated by the RP, and the initial query message will contain the destination realm and Community of Interest. When a Trust Path Query is received by a Trust Router, the router will first authenticate the RP, and check local policy information to determine whether or not to reply.
Assuming that the RP is successfully authenticated and the request passes local policy checks, the Trust Router will search it's tree of Trust Path information to determine whether a Trust Path exists that will reach the destination Realm within the indicated Community of Interest. If so, the shortest/best Trust Path will be returned to the Relying Party.
A Trust Path will consist of a list of steps, each of which will contain: The type of the step (Trust Router or AAA Server), the Community of Interest associated with each step, information needed to reach the indicated Trust Router or server (domain name or IP address), and any special attributes associated with that step.

7. Temporary Identity Request

A Temporary Identity Request is issued by a Relying Party in order to obtain an identity that can be used to traverse each step in the Trust Path. When a Temporary Identity is requested, a Trust Router will provision a new identity in its local AAA infrastructure that can be used by the Relying Party to communicate with the Trust Router or AAA Server that represents the next step in the Trust Path. Current thinking is that KNP will be used as the protocol mechanism for these requests. These Temporary Identities will have a finite lifetime and, when authenticated, will include a Radius Attribute/Diameter AVP indicating that they were generated based on a Temporary Identity Request. This attribute will inlcude the chain of identities that preceeded the current identity in the traversal of the Trust Path.
The details of how these messages will be encoded has not yet been determined. However, it is expected that, for each Trust Router step in the Trust Path, the following actions will take place:

1. The Relying Party will send a Temporary Identity Request message to the Trust Router, containing the identity of the next step in the Trust Path, the destination realm that it is trying to reach, and the Community of Interest in use. This request will

be sent using the identity that the Trust Router obtained from the previous step in the Trust Path (or the Trust Router's permanent identity in it's home realm, if this is the first step).

2. The Trust Router will authenticate the Relying Party.

3. If the authentication is successful, the Trust Router will check local policy to determine whether it should provision an identity for the Relying Party for the indicated purpose (details of this check may be implementation dependent).

4. If the request passes any policy requirements, the Trust Router will provision a temporary identity for the Relying Party within the Trust Router's local realm that can be used to access the next-hop Trust Router or AAA Server in the Trust Path.

8. Security Considerations

This document describes an architecture for the establishment of transitive trust across an ABFAB federation. It describes, at a high level, the entities and protocols that will be used to establish transitive trust, but it does not describe the actual protocols that will be used in detail. Those details, and the detailed Security Considerations associated with them are described in separate documents. It is important to note that the trust established using a transitive trust mechanism described in this document will only be as good as the weakest link in the transitive trust chain. To service the needs of a highly sensitive Community of Interest, stringent criteria must be applied to join the Community, sites must be monitored to ensure that they are adhering to the Community's standards, and local policy may be required to ensure that the chain of trust does not traverse any untrusted, or insufficiently trusted, realms.

9. IANA Considerations

There are no IANA actions required for this document.

10. Acknowledgements

This document was written using the xml2rfc tool described in RFC 2629 [RFC2629].

11. Change Log

11.1. Changes from -01 to -02

*Changed the term "Policy Regime" to "Community of Interest" throughout the document.

*Replaced explicit references to RADIUS and Diameter servers with more generic references to AAA Servers.

*Minor editorial changes.

11.2. Changes from -00 to -01

*Editorial changes, and additional text throughout document.

12. References

12.1. Normative References

[RFC2119]	Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
[I-D.lear-abfab-arch]	Howlett, J, Hartman, S, Tschofenig, H and E Lear, "Application Bridging for Federated Access Beyond Web (ABFAB) Architecture", Internet-Draft draft-lear-abfab-arch-02, March 2011.
[I-D.howlett-radsec-knp]	Howlett, J and S Hartman, "Key Negotiation Protocol (KNP)", Internet-Draft draft-howlett-radsec-knp-02, October 2011.
[I-D.mrw-abfab-trust-router]	Wasserman, M, Hartman, S and J Howlett, "Application Bridging for Federation Beyond the Web (ABFAB) Trust Router Protocol", Internet-Draft draft-mrw-abfab-trust-router-01, October 2011.

12.2. Informative References

[RFC2629]	Rose, M.T., "Writing I-Ds and RFCs using XML", RFC 2629, June 1999.

Authors' Addresses

Margaret Wasserman Wasserman Painless Security 356 Abbott Street North Andover, MA 01845 USA Phone: +1 781 405 7464 EMail: mrw@painless-security.com URI: http://www.painless-security.com

Hannes Tschofenig Tschofenig Nokia Siemens Networks Linnoitustie 6 Espoo, 02600 Finland Phone: +358 (50) 4871445 EMail: Hannes.Tschofenig@gmx.net URI: http://www.tschofenig.priv.at