ESG Security Technical Note:

OpenID Relying Party with SSL

|  |  |
| --- | --- |
| Author/Affiliation | Philip Kershaw, BADC |
| Date | 25 May 2010 |
| Issue | 1 Draft A |

Contents

[Acronyms 1](#_Toc262560407)

[Summary 1](#_Toc262560408)

[RP Hosted over HTTP 2](#_Toc262560409)

[RP Hosted over HTTPS 4](#_Toc262560410)

[Analysis 6](#_Toc262560411)

[Advantages to hosting the RP over HTTPS 6](#_Toc262560412)

[Drawbacks 6](#_Toc262560413)

# Acronyms

|  |  |
| --- | --- |
| OP | OpenID Provider (*Identity Provider* or *server*) |
| RP | OpenID Relying Party (*consumer*) |
| AX | OpenID Attribute Exchange |
| OPeNDAP | Open-source Project for a Network Data Access Protocol |
| TDS | THREDDS (Thematic Realtime Environmental Distributed Data Services) Data Server |
| SSL | Secure Sockets Layer |

# 

# Summary

The ESG Security architecture stipulates SSL for securing OpenID. By hosting OPs over a HTTPS connection the RP and OP can mutually authenticate at the Discovery stage and in addition, RPs can whitelist OPs constraining OPs to a list of trusted sites.

This document describes how this model may be augmented by hosting RPs over SSL also. It examines the relative merits of applying this method against the alternative of hosting the RP over HTTP. The following sections describe the two methods in detail using Sequence Diagrams; the last section some comparison and analysis.

# RP Hosted over HTTP

The following sequence diagram shows an OPeNDAP service protected by an OpenID RP interface hosted at a given Service Provider. The RP is implemented as server middleware which intercepts unauthenticated requests and invokes the OpenID RP sign in interface. The user enters their OpenID URI and the usual OpenID sign in process is followed. The RP requests user attributes from the OP via the AX interface. Connection to the OP is over HTTPS enabling mutual authentication between the OP and RP during the Discovery stage. Attributes are returned from the OP in a HTTP redirect request to the client. The client invokes the redirect location returning the user’s browser to the OPeNDAP URI originally invoked. In this stage, the return to address is HTTP based. Any AX attributes are sent over an unencrypted channel.

The RP processes the AX parameters returned from the OP and sets the authenticated state for the user via a session cookie. The request is passed on within the web application middleware framework to the OPeNDAP service. This serves the request and returns the data to the client.

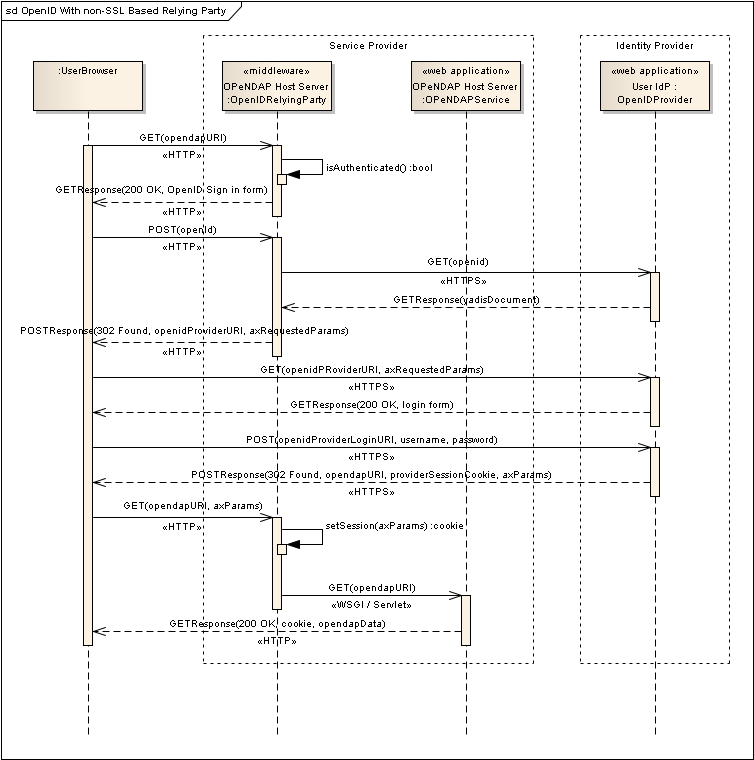


Figure OpenID Relying Party hosted over HTTP

# RP Hosted over HTTPS

This second sequence diagram shows browser based access to an OPeNDAP service secured with OpenID and PKI based authentication. The Service Provider hosts the OPeNDAP service along with an Authentication Service. The OPeNDAP service operates over HTTP whilst the Authentication Service runs over HTTPS. The OPeNDAP service is fronted with an AuthenticationRedirectFilter. This middleware intercepts client requests and redirects unauthenticated requests to the Authentication Service. In this instance, the authenticated state is determined by the presence of a session cookie.

The Authentication Service, hosts an OpenID Relying Party interface but is also capable of authenticating clients with SSL based authentication should the client pass a certificate in the SSL handshake. In the sequence shown, no certificate is passed. The Authentication Service defaults to display the OpenID Relying Party interface.

Note that the OpenID Relying Party interface returns a HTTP 401 Unauthorized response status. This is to enable non-browser based clients a convenient means to detect programmatically that the request has not been authenticated. In the example sequence though, the client is a browser. This displays to the user an OpenID sign in form. Interactions between the Service Provider and client browser are now over HTTPS. The user enters their OpenID. On receipt of this, the Relying Party performs the Discovery stage communicating directly with the user’s Provider and retrieving the Yadis document.

As agreed for ESG security, the Relying Party can at this stage apply whitelisting of Providers and reject a response from an untrusted Provider site.

Assuming the Provider is acceptable, the user is redirected to the Provider for login. They enter their details and perform any further steps required for authentication and perhaps to verify that they are happy to return any of the requested AX (Attribute Exchange) parameters back to the Relying Party.

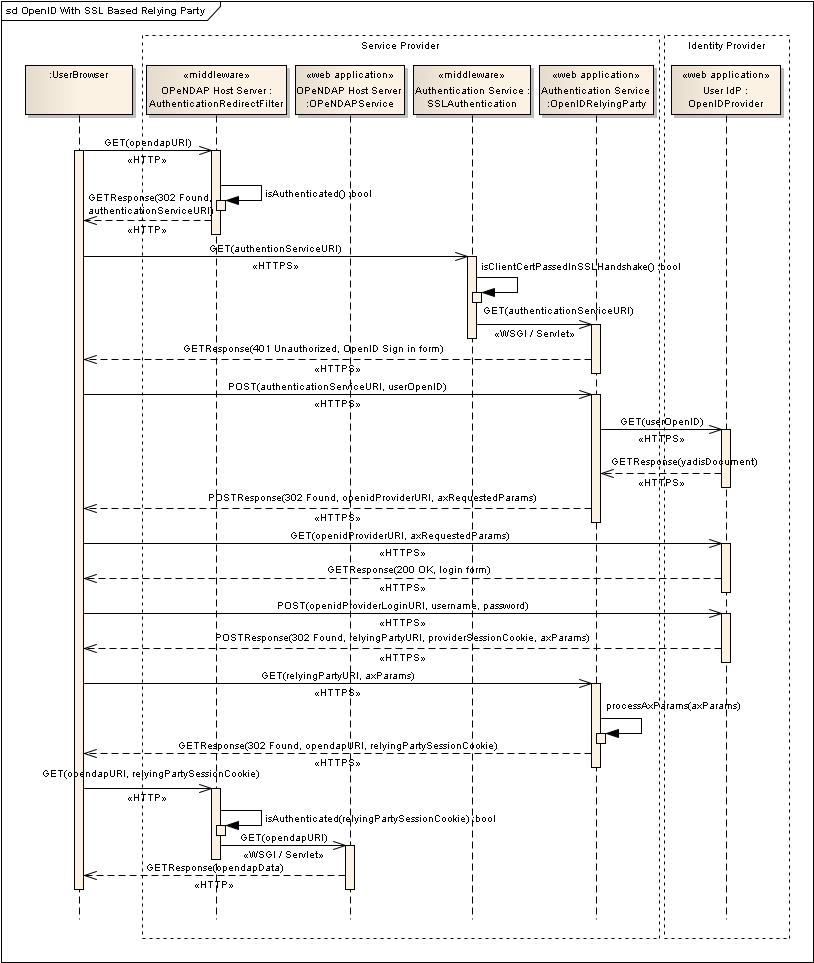


Figure : OpenID sign in with OpenID Relying Party hosted over HTTPS

# Analysis

## Advantages to hosting the RP over HTTPS

1. The ability to preserve confidentiality of returned AX (Attribute Exchange) parameters in client to RP transit: when login is completed at the OP any AX values requested by the RP are returned by the OP are returned to the RP by means of HTTP redirect response to the client[[1]](#footnote-2). When the client has received this and then invokes the redirect URI, this is executed over a HTTP connection. The information could be intercepted by a third party. This is explored further in the following sequence diagram.
2. The ability for the OP to execute a validation check of the RP return to address. Following login at the OP, the OP returns a redirect request to the client to redirect it to the RP. If the RP return to address is HTTPS based, the OP could probe the RP address by initiating an SSL handshake to verify the RP’s SSL certificate.
3. Serve PKI based and OpenID based authentication from the same endpoint. Services based on OPeNDAP e.g. TDS are typically accessed by specialist HTTP clients. These are more easily adapted to a PKI based solution than OpenID based one. By serving PKI and OpenID based authentication from the same endpoint, clients can select their preferred authentication scheme whilst the server remains agnostic as to the client type.

Point 3.) is key to the ESG deployment of OPeNDAP service. Most access is likely to be via specialised HTTP clients and not via a browser.

## Drawbacks

1. An additional redirect from HTTPS based RP endpoint to HTTP is required. The redirect could be exploited but since the redirect request is over HTTPS it could not be modified in transit. It would have to occur in the browser client.
2. RP running over HTTPS sets a cookie which must be visible over HTTP in the same domain e.g. <http://mydomain> and <https://mydomain> can see the same cookie. (This has been tested with Python/Apache and Java/Apache Tomcat based implementations). Wildcard sub-domains have not been tested.
3. It places an additional burden of certificate management on the RP host organisation. RPs must have certificates to support all the different domains in which services are to be hosted within a given organisation. SSL certificates with Subject Alt Names could help with this issue.

1. If the AX content is such that the URL length limit is exceeded, the OP switches to the POST method to return attributes. [↑](#footnote-ref-2)