EduSource Communication Layer Security Model

DRAFT 0.5
September 25, 2004

Dr. Marek Hatala and Ashok Shah

School of Interactive Arts and technology
Simon Fraser University
Surrey, British Columbia, Canada
{mhatala, ashaha}@sfu.ca

Overview

EduSource Communication Layer (ECL) [ECL] consists of ECL protocol and supporting software and infrastructure components. ECL Connector is a middleware component that implements ECL Protocol and provides and API for connecting repositories and tools to ECL Network. The ECL Gateway is an infrastructure component that allows communication between ECL Network and other repositories and networks by transforming between ECL protocol and other protocols. Finally, ECL registry is a registry that allows ECL Clients to discover other ECL-based repositories and services[1].

The ECL protocol implements four main functions as defined by the IMS DRI [IMSDRI] specification: Search/Expose, Gather/Expose, Request/Deliver, and Submit/Store. The ECL protocol is SOAP based protocol and uses document type approach to web services [DOCWS]. This means that all ECL protocol actions are defined as XML documents and can be validated by XML schemas which are part of the ECL protocol definition. The ECL protocol messages are exchanged between ECL connectors as a payload of SOAP messages.

The ECL connector provides the API for ECL protocol functions and for access to services of ECL infrastructure. The API effectively hides the complexity of the ECL protocol. Each ECL service provided by the repository connected on the ECL network is described by the endpoint and service parameters. The ECL registry is a UDDI based registry allowing end-users to search and select services they wish to use. Pre-selected services are stored internally by the ECL connector that directs ECL messages to those services. The user can modify the selection of services anytime. For example, when user initiates the search request for objects the ECL connector federates the request to all repositories pre-selected by the user. The connector then receives search results and makes them available via API. ECL Connector configuration manager simplifies the implementation, configuration and deployment/registration steps when connecting repository to the ECL network.

1 This includes ECL gateways and indirectly non-ECL repositories and networks.

This document describes the security model for the federated search in the eduSource Communication Layer (ECL). The same model applies to other ECL Protocol functions such as ECL Submit and ECL Request services. This document does not cover an internal treatment of the security issues inside individual repositories.

Assumptions and Requirements

For the purpose of this document we will identify the following three entities as the main components the ECL infrastructure: ECL Client that sends a request on behalf of the user, ECL Provider representing the repository service, and ECL Registry holding information about existing ECL Providers on the ECL network. The ECL Client can search the ECL registry to discover ECL Providers and their related information. The ECL Client stores selected ECL Providers and any related information about them in its locally. ECL Client uses the information stored locally for further communication with ECL Providers.

The ECL Network adopted federated search an approach for searching across ECL Providers. The ECL search functionality is concerned with searching for the metadata records describing other resources. In this document we describe the design of the security mechanism with respect to searching for the metadata as it is the most complex of all ECL Protocol requests. All other ECL protocol requests use the same security mechanism. However, it should be explicitly noted that we are not concerned about securing access to the resources themselves where we assume the existing means provided by the content repositories will be used.

The main goal of the ECL security mechanisms design is to provide the means to support different levels of security (security profile) as required by a particular ECL Provider policy. Each ECL Provider can choose to implement different profile. The ECL Client will communicate with each of the providers using their specific security profile. The ECL aims to implement the following three security profiles for ECL Providers:

1. Anonymous access profile (no security) as provided by ‘plain’ ECL
2. Username token profile where user has to have an account with the ECL Provider. The ECL Provider is responsible for the authentication and authorization using its own mechanisms. This profile can be optionally implemented in signed and/or encrypted form.
3. Federated security profile (SAML-based profile) builds on the idea of the federation as understood by the Shibboleth working group [Shibboleth]. In particular, in the current design we assume that ECL Clients and providers are part of the same federation and have a shared vocabulary of roles and attributes. The required infrastructure supporting SAML-based profile is the same one that is developed and used by the LionShare P2P Security [LS-SEC]. This profile can be optionally implemented in signed and/or encrypted form.

As this document is focused on the design of the security from the perspective of integrating Secure ECL with LionShare P2P network we will further focus on the design of ECL federated

security profile.

The ultimate goals for the design of ECL security are to provide following features:

1. Security:
    1.a. ECL Provider has to be able to make authorization decisions to provide access to its services
    1.b. Content integrity of the ECL messages should be protected
    1.c. Content of the ECL messages can be encrypted if required
2. Privacy:
    2.a. ECL Client identity should not be exposed to the ECL Provider
3. Interoperability
    3.a. The ECL security infrastructure has to be compatible with the LionShare security infrastructure
    3.b. The ECL security infrastructure has to be compatible with the vanilla Shibboleth infrastructure
    3.c. The ECL security mechanism has to follow the existing specifications, such as WS-Security

The anonymous security profile does not provide any security and therefore does not meet any of the requirements specified above. The rest of this document discusses two other profiles.

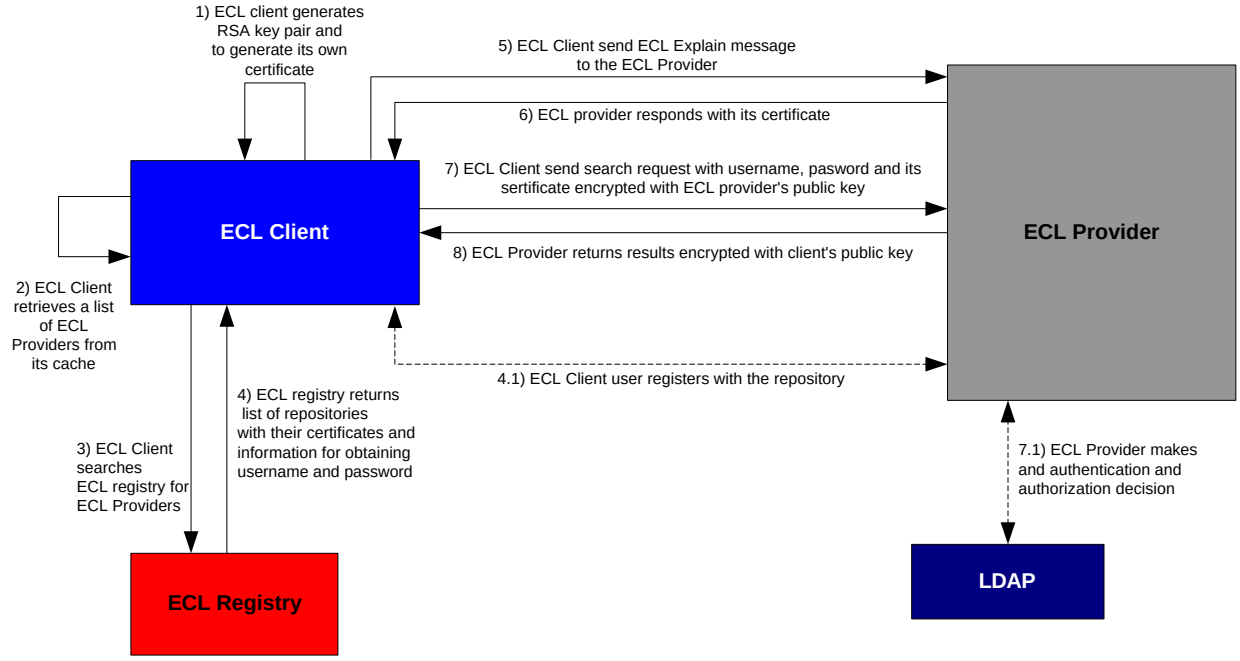
ECL Username Token Security Profile

The ECL Username Token Security Profile is used by the ECL Providers who maintain their own database of users and are fully responsible for authentication and authorization of their users. The user of ECL Client has to have an account with the ECL Provider to be able to use the services using username token security profile. ECL does not provide a direct support for registration of the users; instead it allows the ECL Provider to specify the URL where the user can register in the ECL Registry. When the user of ECL Client selects the ECL Provider with username token profile the ECL Client asks the user to provide username and password for this provider and provides information where the user can register. Once the user provides the credentials, those are cached in the ECL Client and used in all subsequent communication with the ECL Provider.

Alternative versions of this profile can provide signed and signed and encrypted communication. To encrypt messages the ECL Client uses public key from the ECL Providers network certificate. To encrypt messages from the ECL Provider the short-lived certificate generated by ECL Client is used.

As ECL Provider makes an authentication decision based on the user's name the ECL Username Token Profile does not provide privacy as specified above. All other requirements are met.

ECL Username Token Security Profile Diagram



The diagram above assumes that the ECL Provider has either registered with the ECL Registry or the ECL Client has obtained the information about the ECL Provider through other means, including the information that the ECL Provider supports federated security profile.

Steps 1 and 2 are performed during the ECL Client startup:

1) When the ECL Client starts up, it generates a RSA key pair and generates its own certificate.
2) ECL Client retrieves the list of the pre-selected ECL Providers from its cache[2]. This includes previously cached ECL-PNC and list of attributes required by the ECL Provider.

Steps 3 and 4 can be performed by the ECL Client anytime:

3) ECL Client searches ECL Registry for ECL Providers.
4) ECL Registry returns a list of ECL Providers together with their network certificates (ECL-PNC) and the information about supported profiles.
   4.1) When user selects ECL Provider supporting ECL Username Token Profile the user is instructed to enter the username and password for the selected ECL Provider. The user is also presented with a URL[3] where s/he can register to obtain the username and password for the ECL Provider.

[2] Pre-selected ECL Providers were selected by the end user. This could be accomplished either by searching the ECL registry in optional steps 4 and 5 or by directly supplying the ECL configuration file.
[3] The URL is part of the ECL Provider's registration record.

Steps 5 and 6 are repeated for each ECL Provider with expired ECL-PNC or non-existent ECL-PNC:

5) ECL Client sends an ECL Explain message to ECL Provider.
6) The ECL Provider replies with its network certificate (ECL-PNC). The ECL Client permanently caches the ECL-PNC.

The following steps are repeated for each search request send to the ECL Provider supporting ECL Username Token Profile:

7) ECL Client formulates the query, adds username, password and its certificate to the ECL message (using WSS4J username token profile), encrypts ECL search message with the public key extracted from ECL-PNC and sends the message to the ECL Provider.
8) The ECL Provider decrypts the message using its own private key used to generate ECL-PNC, extracts username and password and authenticates the user using its own mechanism (for example using LDAP). Then it performs the access control check based on its access policy, and if satisfactory, then performs the ECL search, obtains search results and formulates the ECL response message. The ECL Provider then extracts the public key from the user's certificate and encrypts the ECL message and sends it back to ECL Client. ECL Client receives encrypted search results, decrypts the results using its own private key and displays them.

ECL Federated Security Profile

The ECL federated security profile assumes that an ECL Client represent the end user affiliated with the trusted entity that is a part of the federation. In LionShare, the ECL Client is integrated with LionShare Peer giving LionShare user full access to the ECL Network.

The ECL federated security profile is designed to be fully compatible with the LionShare P2P network security infrastructure. For the full details consult LionShare Security WhitePaper [LSSEC].

The federated security profile depends on the 'trust federation' that exists between the ECL Client's Local Attribute Authority (LAA) and/or Certification Authority (CA) and ECL Provider (a repository). The ECL Provider trusts the LAA and Certification Authorities that are members of the "Trust Federation".

ECL search functionality is supported in the ECL Connector in the form of a federated search. However, for the purpose of the security design we will describe the approach with respect to the one ECL Provider only. When dealing with multiple providers some of the steps have to be performed for each ECL Provider independently.

A Sample Use Case for ECL Federated Security Profile

From the security perspective ECL Client is represented in the trust network by its ECL Client Network Certificate (ECL-CNC). The ECL-CNC is an alternative to the Shibboleth handle [Shibboleth]. The ECL Client will obtain its ECL-CNC from the Certificate Authority during the client initialization time and will use it during the whole session. The ECL-CNC will be stored in the memory only and it will expire after 8-10 hours or get discarded when the ECL Client is shut down.[4]

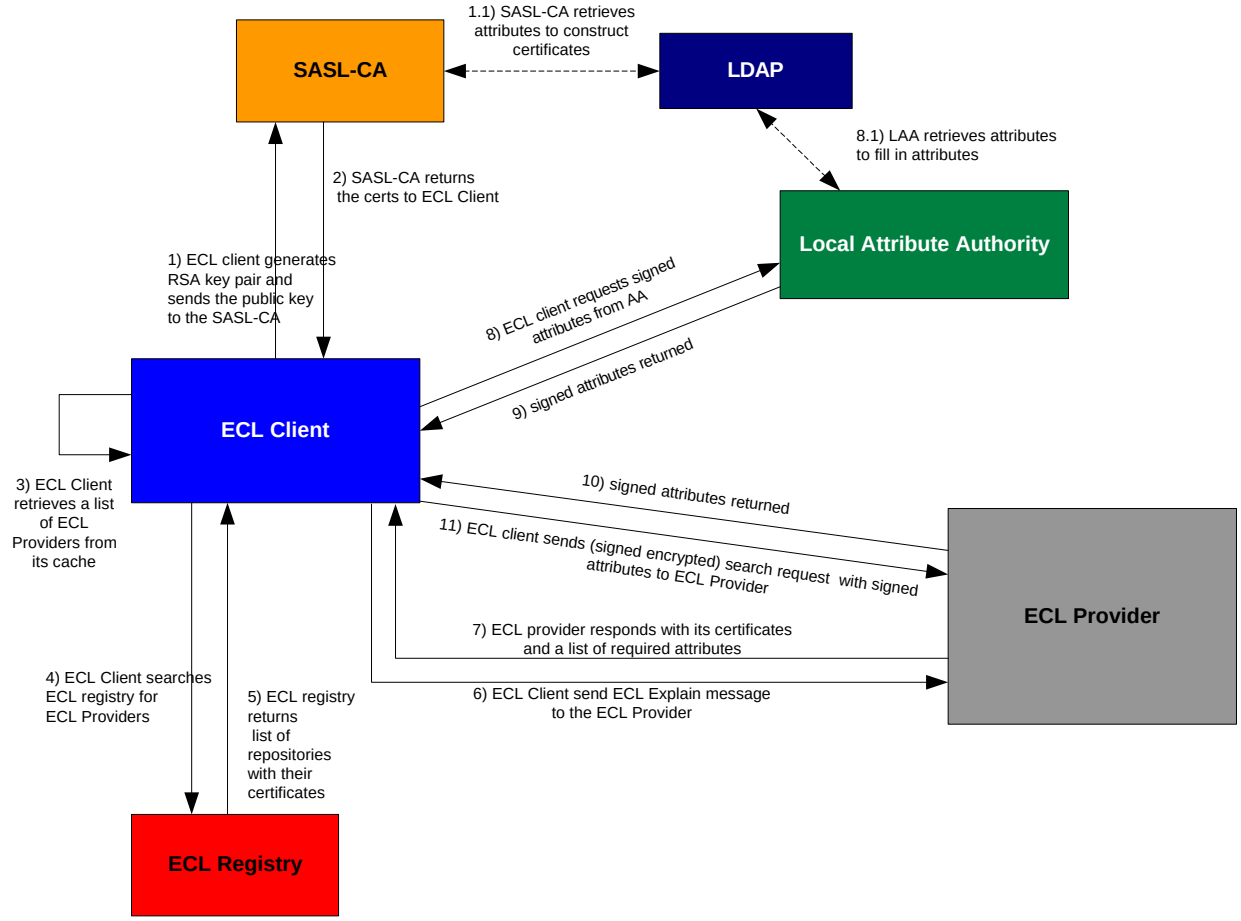
ECL Provider (a repository) will obtain the ECL Provider Network Certificate (ECL-PNC) from the highly trusted Certificate Authority. The ECL-PNC is a permanent certificate with a long expiry time. The certificate will be typically available through the ECL Registry and will be downloaded to the client at the time ECL Client user selects the ECL Provider from the registry. Another option how ECL Client can obtain ECL-PNC is by sending ECL Explain message to the ECL Provider. The ECL Client will cache ECL-PNC for each ECL Provider permanently until ECL-PNC expires. When ECL-PNC expires the ECL Client needs to obtain new ECL-PNC from the ECL Provider by sending another ECL Explain message.

Each ECL Provider is unique in the way how they control access to their resources. With that respect, each ECL Provider can require different set of attributes[5] to grant the access for person authenticated in a particular role. The ECL Client will obtain a list of attributes required by the ECL Provider by sending a special ECL Explain message to ECL Provider during the ECL Client startup. After obtaining the list of attributes the ECL Client asks Local Attribute Authority to fill and sign the SAML attributes and ECL Client will cache signed attributes for each pre-selected ECL Provider. The

The ECL Connector will use WSS4J SAML assertion profile to include security layer into the ECL Protocol. The ECL Search message will include the search query, ECL Client network certificate (ECL-CNC), signed attributes, and attribute authority certificate. The message can be encrypted with the public key from the targeted ECL Provider repository's network certificate (ECL-PNC). The repository would decrypt the search query, validate the certificate, perform authorization based on supplied attributes and process the search request. When returning the search results, repository would use the public key from the ECL-CNC to encrypt the search results and sign the results.

[4] These assumptions are consistent with the LionShare security design. In case of the use of ECL Client in the LionShare Peer client the ECL Client Network Certificate is identical to the LionShare client certificate.
[5] The attributes required by the ECL Provider have to be selected from the set of attributes agreed upon within the trust federation.

ECL Federated Security Profile Diagram



The diagram above assumes that the ECL Client is part of the federation and is associated with the SASL-CA and LAA. It is also assumed that the ECL Provider has either registered with the ECL Registry or the ECL Client has obtained the information about the ECL Provider through other means, including the information that the ECL Provider supports federated security profile.

Steps 1-3 are executed during the ECL Client startup (some of them being optional):

1) When the ECL Client[6] starts up, it generates a RSA key pair, and sends the public key to the ECL SASL-CA authority.
   1.1) SASL-CA retrieves attributes from the organization registry (for example LDAP) to construct certificates.
2) SASL authority authenticates the ECL Client and sends return the client's network certificate (ECL-CNC)[7]. The ECL-CNC contains the handle for the user and the role s/he was authenticated in.

[6] Any ECL Client that implements ECL connector enables application to communicate via ECL.

[7] The RSA public key that ECL Client sent is used to generate the ECL Client Network Certificate (ECL-CNC). The ECL-CNC only has the identity of the network it belongs (for e.g. Splash or LionShare or EdNA etc). The individual identity of the client is not disclosed in the Network Certificate.

3) ECL Client retrieves the list of the pre-selected ECL Providers from its cache[8]. This includes previously cached ECL-PNC and list of attributes required by the ECL Provider.

Steps 4 and 5 can be performed by the ECL Client anytime:

4) ECL Client searches ECL Registry for ECL Providers.
5) ECL Registry returns a list of ECL Providers together with their network certificates ECL-PNC.

Steps 6 and 7 are repeated for each ECL Provider supporting ECL Federated Security Profile if either a list of required attributes is missing or ECL-PNC expired:

6) ECL Client sends an ECL Explain message to ECL Provider requesting a list of required attributes and ECL-PNC if ECL-PNC expired or is non-existent:
7) The ECL Provider replies with a list of attributes and its network certificate (ECL-PNC). The ECL Client permanently caches the ECL-PNC and the list of required attributes.

Steps 8 and 9 are repeated for each pre-selected ECL Provider supporting ECL Federated Security Profile:

8) ECL Client request Local Attribute Authority9 to fill and sign the required attributes for each ECL Provider.
   8.1) Local Attribute Authority retrieves values for the attributes contacts organizational registry (such as LDAP).
9) The Local Attribute Authority for ECL Client fills requested attribute values, formats them in SAML assertion format, and signs them with the Local Attribute Authority Certificate (LAAC). Local Attribute Authority sends the signed attributes and LAAC to the ECL Client which keeps the signed attributes and LAAC in the memory.

The following steps are repeated for each search request send to the ECL Provider supporting ECL Federated Security Profile:

10) ECL Client formulates the query, adds requested signed attributes and LAAC to the ECL message, adds its own network certificate ECL-CNC, encrypts ECL search message with the public key extracted from ECL-PNC and sends the message to the ECL Provider.
11) The ECL Provider decrypts the message using its own private key used to generate ECL-PNC, extracts LAAC, ECL-CNC and signed attributes and validates them. Then it performs the access control check against the signed attributes and its access policy, and if satisfactory, then performs the ECL search, obtains search results and formulates the ECL response message. The ECL Provider then extracts the public key from the ECL-

[8] Pre-selected ECL Providers were selected by the end user. This could be accomplished either by searching the ECL registry in optional steps 4 and 5 or by directly supplying the ECL configuration file.
9 Local Attribute Authority (LAA) forms a federated network in ECL. When, LAA boots it authenticates with ECL SASL-CA and gets its LAA certificate. Repositories trust this LAA certificate and allow searching. Also every time ECL Client sends a request for attributes, LAA authenticates PeerECL.

PNC and encrypts the ECL message and sends it back to ECL Client. ECL Client receives encrypted search results, decrypts them using its own private key and displays them.

References

[ECL] EduSource Communication Layer, http://www.edusplash.net/technical/ecl/index.html

[IMSDRI] IMS Digital Repositories, http://www.imsglobal.org/digitalrepositories/index.cfm

[LSSEC] LionShare Security White Paper, URL???

[DOCWS] Reap the benfits of document style web services, http://www-106.ibm.com/developerworks/webservices/library/ws-docstyle.html

[Shibboleth] Internet2 Shibboleth Project, http://shibboleth.internet2.edu/