A Pseudonymity System for Grids

Joni Hahkala, Henri Mikkonen, Mika Silander, and John White
Helsinki Institute of Physics, Technology Programme
Email: {joni.hahkala, henri.mikkonen, john.white}@cern.ch, mika.silander@hip.fi

***Abstract*—Traditionally, Grid users have been identifiable and traceable beyond reasonable doubt by their digital certificates. However, Grids are used in an ever-increasing variety of contexts and thus, the number of usage scenarios has augmented accordingly. In bio-medicine and other health-related fields a need for anonymous access to Grid resources has been identified. Anonymous access to resources prevents the resource owners and other external parties from tracing the users and their actions. Such anonymity of resource usage in Grids is needed above all in commercial contexts, e.g. protecting the development process of a new medicine by anonymizing the accesses to medical research databases. In this paper we identify the requirements and define an architecture for pseudonymity system addressing these needs. Also the protocols used between the components are defined.**

***Index Terms*—Authentication, Authorization, Grid Security, Pseudonymity.**

I. Introduction

The Grid computing model envisages a heterogeneous fabric of computing resources that is provided to users in a transparent way. In this model, Grid users may run processes on computing resources and store and access data on storage resources that may not be owned by them or even their parent organization. The use of resources on any Grid infrastructure entails a balance between the owner's need to oversee and account for the resource usage and the user's privacy requirements.

From the Grid users' point of view, complete anonymity is desirable for maximum protection. This requirement can come from researchers in a field of competitive, commercial or basic research [1], [2]. These researchers may wish to work in secrecy and prevent their competitors from following their actions on a Grid. This would include being able to anonymize the credentials used for job submissions and the reading and writing of data. In general, this is not possible due to requirements that a Grid user should be traceable for accounting purposes and in the case of usage policy violation.

Hence, the anonymity problem is to find a compromise between the requirements of the Grid resource owner and users. The proposed solution to this problem is the concept of a lesser degree of anonymity, *pseudonymity*: the use of pseudonyms as identifiers [3]. A pseudonym is a unique anonymous identity given by a trusted third-party (service) to a Grid user. Only this trusted third party is able to re-establish this identity association later if necessary. In situations where resource owners detect misuse of their resources, the trusted third-party can act as an middle man to solve grievances or in serious cases can be requested to disclose the true identity of the suspected abuser, subject to the policies of the particular Grid infrastructure or the law.

The system presented in this paper is designed to hide the identity of the user invoking the operations on the Grid. If used properly and provided there is a sufficiently broad mix of operations and end users, the system will also prevent the correlation of operations and thus ensure the resource owners cannot identify users or research patterns by workflow tracking.

This paper extends our previous work *Requirements and Initial Design of a Grid Pseudonymity System* [4], which appeared in the Proceedings of the 2008 High Performance Computing & Simulation Conference (HPCS). Especially we describe the protocols for the communication between the components.

The rest of the paper is organised as follows: Chapter II looks into work generally related to pseudonymity. Chapter III offers a detailed study of the requirements. Chapter IV analyses the problem in light of the requirements and constraints. Chapter V discusses some architectural and functional issues. Chapter VI describes the solution to the problem and defines the protocol to be used in the interaction between the components. We summarize our findings in Chapter VII and propose future work in Chapter VIII.

II. Related work

Pseudonymity and pseudonym identifiers have already been covered by several specifications and software. Some of their definitions and interpretations are discussed in this section.

A. Terminology

Andreas Pfitzmann et *al.* [3] have defined a proposal for terminology in the area of anonymity and pseudonymity (among some others) which we use in this paper. They define that a pseudonym refers to an identifier of a subject other than one of the subject's real names, where the subject is the holder of the pseudonym. A subject itself is

This paper is based on "Requirements and Initial Design of a Grid Pseudonymity System," by J. Hahkala, H. Mikkonen, M. Silander, and J. White, which appeared in the Proceedings of the 2008 High Performance Computing & Simulation Conference (HPCS), Nicosia, Cyprus, June 2008.

This work was supported in part by the European Union grant EGEE-II INFSO-RI 031688 and by the Joint Committee of the Nordic Research Councils for Natural Sciences (NOS-N) through its Nordunet-3 programme project "Innovative services and tools for Nordugrid".

pseudonymous if a pseudonym is used as identifier instead of one of its real names.

For the linking between a pseudonym and its holder, they also define typical kinds of pseudonyms as follows:

- *Public pseudonym*: the linking may be publicly known from the very beginning.
- *Initially non-public pseudonym*: the linking may be known by certain parties, but is not public at least initially.
- *Initially unlinked pseudonym*: the linking is - at least initially - not known to anybody with the possible exception of the holder himself.

B. Shibboleth and SAML

Shibboleth [5] is Internet2's project to provide Single Sign-On (SSO) on the Web. The SSO is based on the federated user attribute distribution in a privacy-preserving manner. The current version (2.1) supports major portions of SAML 2.0 [6] standard, including the use of short-lived opaque name identifiers. A typical use case starts at a Service Provider (SP) which wants to know some attributes of the user. Instead of authenticating the user directly, the SP redirects the user to the Identity Provider (IDP) for authentication. Once authenticated and authorized, the IDP generates an opaque name identifier for the user and communicates it to the requesting SP. The name identifier is then utilized by the SP for obtaining user attributes from the IDP.

As the name identifier is opaque and short-lived, the SP cannot determine any additional user information apart from the attributes that are provided by the IDP. Thus the type of the (pseudonym) identifier is initially non-public. The attributes may include only the Virtual Organization (VO) membership information that is enough for the SP to authorize but not to individualize the user.

The integration of Shibboleth's SAML attribute framework and Grid security has been studied by the GridShib project [7]. The goals of the project include e.g. utilization of the Shibboleth attributes in the user authorization process, but also pseudonymous access for the Grid users [8]: Shibboleth's opaque name identifiers are used in the subject fields of the X.509 certificates which are issued to the users online after Shibboleth authentication. The users' Shibboleth attributes can be utilized with these pseudonym certificates too.

C. idemix

The Shibboleth approach is very IDP-centric as the IDP keeps track of its users' accesses to the SPs. Anonymous credential systems provides another approach. They allow the user's transactions to be carried out in a way that they cannot be linked to the same user [9]. A software called idemix (identity mix) [10] is an example of such a system.

The users establish pseudonyms with SPs that are used for creating credentials containing a set of attributes. Afterwards, the users present the credentials with desired sets of attributes to the same or other SPs by using zero-knowledge proofs. These proofs ensure the legitimate possession of the credentials but reveal no information about the true identity of the user employing them. The credentials can be used for obtaining new credentials, but only one master secret is related to all of them. Mechanisms for retrieving the identity of a user locally (one SP) or globally (all the SPs) exist, but they require the user's cooperation.

D. WS-* specifications

As an alternative to SAML, WS-* (Web Services) specifications also provide a model for federation between IDPs and SPs. From the set of specifications, WS-Federation defines a Pseudonym Service which maintains alternate identity information for its users [11]. The pseudonym identifiers are part of the security tokens that are used by resources' Security Token Services (STS) for authentication and authorization purposes. In addition to the Shibboleth-style short-lived (or one-time) opaque pseudonym identifiers, anything between them and constant clear-text identifiers are supported. As the communication with the service itself can occur via IDPs, the resources' STS:s, or directly with the resource or the requestor, numerous use cases are supported. A Pseudonym Service and the claims-based authorization model can be used to describe the set of attributes required to access a resource and the IDP can assert that a particular user possesses those attributes, without divulging their actual identity.

III. Requirements

In order for a pseudonymity system to function with current Grid middleware it should fulfill the following requirements.

Requirement 1. ***Confidentiality***
The pseudonymous identity must hide the true identity of the user.

The true identity must remain unknown to the service provider sites, their administrators and other legitimate Grid users as well as external parties. This implies encrypted communication is needed between pseudonym attestee and attester.

Requirement 2. ***Non-repudiation/Retrievability***
The true identity of a Grid user must be, a posteriori, unambiguously traceable via the pseudonymity system.

This functionality is mandatory in cases of misuse and may be imposed by regulatory or law enforcement issues. To this end, the pseudonymity attester must authenticate pseudonym requestors and maintain a record of the pseudonyms issued.

Requirement 3. ***Uniqueness and Short Life-time***
A pseudonym must be a unique, short-lived one-time identity in the Grids in which it is to be employed.

A pseudonym's Distinguished Name (DN) must not clash with the existing user DNs, nor with other pseudonyms as this would undermine the overall user authentication and violate the earlier requirement of retrievability. And ideally, only one Grid operation or set of operations should be performed under the protection of a pseudonym. For the next operations, a new pseudonym should be requested. This approach reduces the ability of outside observers to collect data for correlation attacks with the intent of discovering the true identity of the user. Pseudonymous credentials should be ephemeral to reduce the damage in cases of credential compromise. Due to ephemerity and large volume of issued credentials, the issuance itself should involve no manual intervention nor procedures.

Assuming there are several independent pseudonym attesters active in a Grid, each must be assigned an own unique name space. This name space prevents attesters from accidentally issuing pseudonyms with identical DNs.

Requirement 4. ***Identity Protection***
The pseudonymity attester must be the only party able to obtain the true identities of users.

The pseudonymity attester must adequately protect the records of issued credentials and the systems into which they are stored. Only authorized people are entitled to uncover the true user identities.

Requirement 5. ***Credential Source Compatibility***
The pseudonymity system should interoperate with different sources of Grid user credentials.

Even though the user authentication is based on credentials, they may not necessarily come directly from the user's client software. The user's short- or long-term credentials can be stored in online credential repositories or be delegated to other Grid services acting on a user's behalf. For example, some portal usage scenarios involve the delegation of the user's proxy certificate [12] directly to the portal with no user intervention. Hence, the pseudonym system must support a broader set of use cases, not only those implied by direct user access.

Requirement 6. ***Information leakage prevention***
The pseudonymity system must actively counteract the leakage of information that allows the unique identification of a pseudonym user.

The operations and actions a pseudonym user performs and the set of additional personal attributes the user may have requested for inclusion into the pseudonym credentials, may provide enough information to uniquely identify the user. The pseudonymity system should therefore attempt to actively reduce and hide sources of such information. In the case of personal attributes from auxiliary authorization systems, the pseudonymity system should either prevent uniquely identifying pseudonyms to be issued, or, warn the user about the high probability of disclosure prior to using such a credential. Other covert sources of information, e.g. IP numbers of Grid job submission hosts, metadata in submitted files and Grid job description language attributes are harder to deal with and their removal or anonymization is ultimately up to the users themselves [1].

Requirement 7. ***Maintaining security***
The pseudonymity system must not provide ways to circumvent existing security.

The pseudonymity must not erode the security of the systems. There might be cases where enforcing a detailed policy would need the user's identity to be revealed and in these cases the policies can't be enforced at the time. Later, the authorized persons can do the enforcement of these policies and the corresponding actions if needed.

IV. Problem Analysis

Many of today's Grid middleware systems authenticate users with PKI certificates. Thus, in addition to the requirements presented in the previous section, we impose on ourselves an implementation constraint, that of compatibility: *The pseudonymity certification must be compatible with the certificate-based authentication and interoperate seamlessly with existing Grid middleware*. This also means the pseudonymity certification must work in concert with other commonly used auxiliary authorization systems such as Virtual Organization Membership Service (VOMS [13]) and Community Authorization Service (CAS [14]) without any significant changes to them.

Existing Grid user certificates and software can be employed with little effort to ensure that pseudonym requesters are unambiguously authenticated with cryptographically strong mechanisms. For the same reason, SSL/TLS channels can easily be set up to guarantee the confidentiality of communications. These fulfill the requirements 1 and 2 and comply with the above implementation constraint.

The pseudonym credential itself can be modeled as a standard X.509 [15] user certificate but having an anonymized DN. The set of resources available to a Grid user acting under a pseudonym will be more limited than if the user had employed their ordinary user certificate. This is due to the fact that authentication and authorization decisions must be based solely on auxiliary attributes provided by auxiliary authorization systems, the exact user identity being unavailable. Thus, users should be able to request some of their real identities' attributes to be included in the pseudonyms and this implies the pseudonymity system needs to interact directly with auxiliary authorization services. The VOMS auxiliary authorization service traditionally models the user attributes as Attribute Certificates [16] (AC). It is commonplace to include these into the extensions of X.509 certificates and this is a further motivation to use X.509 certificates as the format for pseudonym credentials. Nowadays the VOMS service also issues SAML assertions to be embedded into the X.509 extensions, but this is not yet supported by the Grid resources.

Certificate Authorities (CAs) may freely issue certificates unconstrained by any name space. In Grids however, the uniqueness of certificates is guaranteed by reserving specific name spaces for each CA. Only those certificates issued in conformance with the name space restriction are accepted as valid credentials in a Grid. The uniqueness of pseudonym credentials implied by requirement 3, can be ensured similarly by assigning unique name spaces to the pseudonym attesters.

Requirement 3 also states the pseudonym credentials need to be short-lived which implies a high volume of credentials to be issued. Hence they should be generated programmatically. The pseudonym system must therefore incorporate functionality similar to online certificate authority (online CA) services, e.g. EJBCA [17].

Requirement 4 has two implications: firstly, the pseudonym credential must not contain any information as to the identity of its requester, secondly, the internal security procedures and measures of the pseudonym attester must ensure the access to the records is strictly limited to authorized personnel. This requirement along with requirements 1 and 2 mean that the pseudonyms provided by the system can be described as initially non-public pseudonyms as defined by [3].

Requirement 5 describes the different types of sources from which pseudonym credential requests may originate. Pseudonym certificates requested by the user, either with the help of their long-term user certificates or a proxy certificate generated from the former, will leverage the existing X.509 authentication as is. In the course of using Grid resources, the user may delegate their rights to further components acting on their behalf such as the credential repository service, MyProxy [18]. These, in turn, may delegate the credentials further to Grid portals and hence, pseudonymity requests from portals need to be handled. In order to increase entropy and thus hinder statistical correlation attacks (req. 6), a portal should request new unique pseudonym credentials for each job launched. Portals will delegate the pseudonym user's rights further to the point where the job reaches a Computing Element (CE) [19]. The CE is responsible for collecting the resources defined by the job description and selects a computing node for the execution of the job. The collection is done with the permissions of a limited proxy. The CE may also decide to request new pseudonym credentials for each resource access needed for the staging of the job, thus, the pseudonymity system must accept requests from CEs as well.

Requirement 6 is the most difficult to address since there are many sources that indirectly provide more information concerning the pseudonym user's identity. The pseudonym attester may however, guarantee that the additional personal attributes the user wishes to include in the pseudonym credential, e.g. role, group membership information, capabilities etc, will not uniquely identify the user. This requires modification of auxiliary authorization services since these must provide the pseudonymity system information about whether the requested user attribute combination is uniquely identifying or not.

According to the requirement 7, adding a pseudonymity system into the overall security infrastructure must not weaken security nor introduce new security holes. The ability of the pseudonymity system to circumvent purely identity based limitations such as blacklists is, at first sight, one such security hole. However, an abuser of pseudonyms will be detected equally and can be deprived the usage of pseudonyms. Other constraints on the user's credentials such as limitations in a proxy certificate must be preserved.

The Functional Requirements (FR#) and software components that are minimally needed to implement a working pseudonymity system are summarized below:

FR1 The pseudonymity system should authenticate all requests relying on existing Grid security mechanisms, i.e. SSL/TLS communication and X.509 certificates.

FR2 The communications in all interactions should be protected with authenticated and encrypted SSL/TLS channels.

FR3 Pseudonym credentials should be modeled as X.509 certificates.

FR4 Additional individual attributes (role, group etc) should be modeled as Attribute Certificates.

FR5 Pseudonym credential requests should be honoured to entities authenticating with user long-term X.509 certificates and proxy X.509 certificates.

FR6 The credential issuance of the pseudonymity system must not include manual operations, in other words, it should operate in the same manner as an online CA.

FR7 Auxiliary authorization services must offer functionality that allows the pseudonymity system to judge whether additional user attributes to be included in the pseudonym credential identify the user uniquely.

FR8 A pseudonym credential requested with a credential having rights limitations, must, if granted, return a pseudonym credential with identical limitations. A limited proxy is an example of such a credential.

FR9 The pseudonymity system must not create ways to circumvent the security of the system.

V. Discussion

In this section we discuss some architectural and functional issues of the pseudonymity system before describing the solution in the next section.

A. On the architecture of the pseudonymity system components

We outlay our solution alternatives using three independent components: the pseudonymity service, the online CA service and an Attribute Authority. Having this separation allows us to benefit from existing and well-tested Attribute Authority and online CA software. Also, we avoid reimplementing their functionality within the pseudonymity service. In this setting, the pseudonymity service acts as a Registration Authority by authenticating the users and validating their requests before forwarding the requests to the online CA. The pseudonymity service fulfills the traceability requirement 2 by maintaining records of the pseudonyms issued to the authenticated users. Having an automated online CA ensures the timely delivery of pseudonym credentials in accordance with FR6.

Due to the short-life time of pseudonymous credentials, we anticipate that certificate revocation functionality is not vital. However, such functionality can be added later non-intrusively if deemed necessary. This is similar to the fact that Certificate Revocation Lists (CRLs) are not used against individual proxy credentials, but against the underlying long-lived ones.

It is likely that a virtual organization offering Attribute Authority services will also provide a pseudonymity service for its user community. Therefore, even though this division into separate components apparently violates requirement 4, we consider this an insignificant relaxation of constraints.

B. On generating the pseudonymous identities

In principle, the pseudonymous identifiers could be generated by the user or the pseudonymity system. In the first scenario, the user would generate an opportunistic pseudonymous DN and embed it into the certificate request. However, only the pseudonymity system can ensure the uniqueness to fulfill requirement 3.

After this first stage the Grid user has a credential with an unique pseudonymous DN. This credential does not necessarily have the user's attributes attached as these are granted by their Attribute Authorities. Therefore, in order for the users to gain access to additional resources enabled by their attributes, the attributes have to be retrieved in a second stage from a trusted Attribute Authority. It has to have the knowledge of who possesses the pseudonymous identity.

C. On required modifications to Attribute Authority components

The internal data models of Attribute Authorities need to be extended to associate pseudonyms as aliases of real users. Upon reception of an attribute request related to a pseudonym, the Attribute Authority should return information on the degree of uniqueness of the user attributes as stated in FR7. This is not normal feature in Attribute Authority and thus implies slight changes to the ones supported. The threshold degree and what is done when this limit is reached need be configurable on a pseudonym service basis. Ultimately it is however the task of the VOs to attempt to ensure the user groups remain sufficiently large to prevent this from occurring. Also the Attribute Authority must have a way of cleaning up the expired pseudonyms in order that the alias list doesn't become unmaintainably large over time.

VI. Solution

In this section we first give an overview to the pseudonymity system architecture that, in our understanding, fulfils the requirements and functional requirements described earlier. Pros, cons and differences of alternative architectures are discussed in our earlier work [4]. We also define the protocols needed for the interaction between the components.

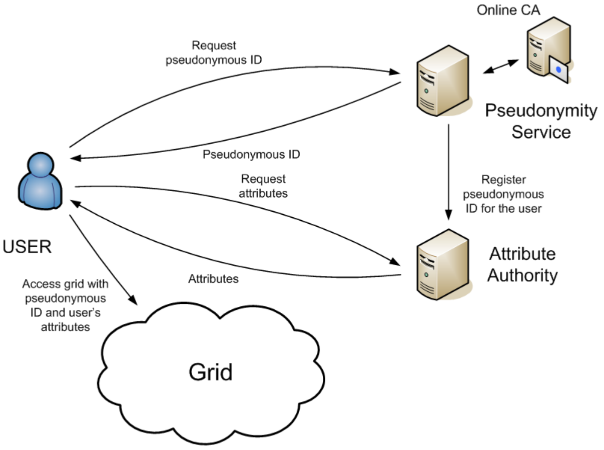

Fig. 1. Relationships between the components of the Pseudonymity System.

Fig. 1 describes the chosen architecture of the pseudonymity system. The user uses client software for communicating with pseudonymity service for obtaining pseudonymous Grid identity, modeled as standard X.509 certificate. The pseudonymity service exploits an online CA for the actual certificate issuance, acting itself as a Registration Authority that authenticates and authorizes users for the online CA.

As a part of of the process, the pseudonymity service registers the certificate to the Attribute Authority as an alias to the "real" user certificate that has already been registered to the VO beforehand. The user can then obtain the same attributes to his pseudonymous identity as to his real Grid identity. When finally accessing the Grid with the pseudonymous credentials, the attributes are used for authentication and authorization purposes by the Grid services.

The pseudonymity service has some similarities with Short-Lived Credential Service (SLCS) [20] as both of them are used for issuing X.509 credentials using a back-end online CA. For this reason, some parts of the existing SLCS specifications [21] are used as a basis for the protocol defined in the next subsections.

A. Protocol Overview

The message exchange for obtaining and initializing the pseudonymous credentials can be seen from Fig. 2. The sequence contains four phases:

- *Login phase* for authenticating the user to the pseudonymity service and providing the information needed for constructing a certificate request.
- *Certificate request phase* for communicating the certificate request first to the pseudonymity service which forwards it finally to the online CA for signing.

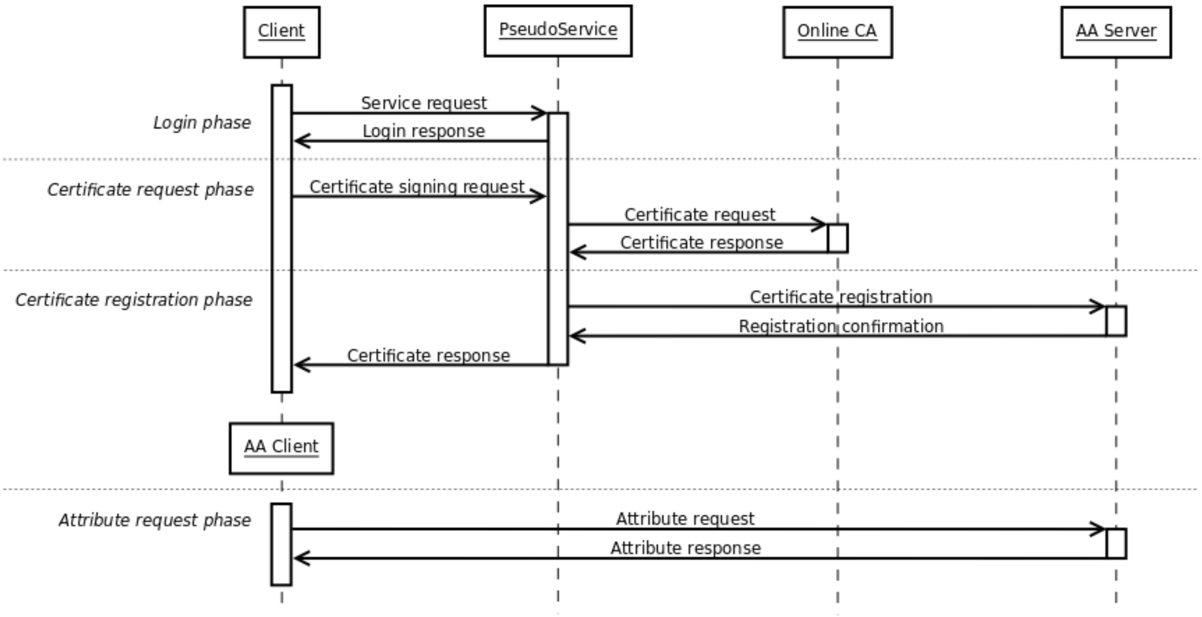

Fig. 2. Message sequence between the components of the Pseudonymity System.

- *Certificate registration phase* for registering the certificate to the Attribute Authority server and returning it back to the client.
- *Attribute request phase* for initializating the pseudonymous identity with a desired set of attributes from the Attribute Authority server.

The communication between the client and pseudonymity service consists of two sequential request/response steps. The first step is equivalent to the login phase as the second contains both the certificate request and registration phases. Together the steps form a two-step request/response protocol whose generalized schema definitions are listed in the following two subsections.

All the XML messages may optionally contain an extension element for protocol extensions established by predefined deployment-specific agreements between the client and the pseudonymity service. The contents of the element must be from the different XML schema namespace.

As the pseudonymous certificate is registered to the Attribute Authority in the third phase, existing protocols and client software implementations can be used in the attribute request phase. They just need to be directed to use the pseudonymous credentials in the process. This phase does not have to be executed directly after the three previous ones and can technically be repeated several times for different attributes. However, ideally only one set of Grid operations should be performed under one pseudonymous identity as described with requirement 3.

B. Protocol Schema: Certificate Information Step

The goal of the first step of the protocol is to authenticate the user to the pseudonymity service and provide the authorized users with the information needed for constructing a certificate request.

Request. After the authentication and authorization process to the pseudonymity service, no input is required from the client for the response generation.

Response. The pseudonymity service prepares a `<CertInfoResponse>` XML message for the response. The message has the following elements:

- `<Status>` (Required) - The status of the request processing (‘Success’ or an appropriate error message).
- `<SubjectDN>` (Optional) - The subject DN to be included in the certificate request.
- `<AuthorizationToken>` (Optional) - The authorization token that must be relayed back in the certificate request if it exists.
- `<ServiceEndpoint>` (Optional) - The URL for the pseudonymity service’s certificate service endpoint.
- `<CertificatePolicy>` (Optional) - Policy-related information like key strength, key algorithm and a list of critical and non-critical certificate extensions required in the certificate request.

The message is finally returned to the client as a response to the service request. It is used as a basis for the next step in this two-step protocol.

C. Protocol Schema: Certificate Request and Registration Step

The goal of the second step is to generate the certificate request and communicate it first to the pseudonymity service that, in turn, forwards it to the online CA for signing. The second step registers the certificate as an alias to the Attribute Authority. The pseudonymity service finishes the step by sending a response message back to the client.

Request. The client prepares `<CertRequest>` XML message using the response message from the first step as the source for required information. The message contains the following elements:

- `<CertificateSigningRequest>` (Required) - PKCS #10 [22] conformant certificate request encoded in the Privacy-Enhanced Mail (PEM) format [23].
- `<AttributeAuthority>` (Optional) - A pointer to the Attribute Authority for registering the certificate by the pseudonymity service.
- `<AuthorizationToken>` (Optional) - The authorization token must be relayed back if it was included in the <CertInfoResponse> message.

In addition to the subject DN and optional certificate extensions obtained from the `<CertInfoResponse>` message, a public key is a fundamental part of the Certificate Signing Request (CSR). Thus the client must generate a new public/private keypair and include the public key to the CSR. The private key must be kept safely by the client: the key should be encrypted if stored in the filesystem for example. Other mechanisms such as smartcards can also be utilized for generating the keypair and storing the private key.

The completed `<CertRequest>` message is sent by the client to the pseudonymity service's certificate service endpoint.

Response. The pseudonymity service first verifies the validity of the CSR and optional authorization token by comparing the contents to those issued in the first step of the protocol. That valid CSR is forwarded to the online CA. Depending on the protocol used between the pseudonymity service and the online CA, the CSR may be converted to a different format.

Successfully issued certificates are registered to the Attribute Authority specified in the request message. A default Attribute Authority can be used if the element was not present.

After the registration, the pseudonymity service prepares a `<CertResponse>` XML message with the following elements:

- `<Status>` (Required) - The status of the certificate request processing ('Success' or an appropriate error message).
- `<Registration>` (Optional) - The status of the certificate registration to the Attribute Authority ('Success' or an appropriate error message).
- `<Certificate>` (Optional) - The certificate encoded in the PEM format. Present if the status was successful, absent in the error cases.

The message is sent to the client as a response to the request. The client may store the certificate if it was included in the response message. This is the most typical case as, in addition to the private key, the certificate is used in the TLS/SSL handshake for example.

D. On Protocol Bindings and Profiles

The request-response messages need to be transferred between the components over the network. Several communication protocols and frameworks already exist for this purpose. For example, the SLCS specifications that were used as a basis for our protocol use an embedded HTTP client in the command-line client tool. Standard HTTP GET and POST methods are used for sending the request to the service and the response is obtained from the body of the corresponding HTTP response. This approach has also been proven to interoperate with wide variety of devices (e.g. mobile phones) that support the mobile edition of the Java framework [24]. Another example approach would be making the pseudonymity service a Web Service and embedding the protocol inside SOAP messages.

The requirements for the user authentication and authorization mechanisms are also relatively sparse: at the very least the pseudonymity service only needs the information to "individualize" the user to the identity that is already registered to the Attribute Authority. This can be obtained from the certificate used in the TLS/SSL client authentication process or from the attributes provided by a Shibboleth IDP for example.

VII. Conclusions

This paper describes the requirements, proposes an architecture and the required protocol for a general pseudonymity system that provides pseudonymous access to the Grid. The system allows the users to employ their attributes to access the Grid while hiding their true identity. The pseudonyms are initially non-public as the relationships between the true and pseudonymous identities can be revealed by authorized people e.g. in the cases of misuse.

The repercussions of hiding the user's identity are hard to determine without getting real world experience with the prototype. A prototype will also shed light on the magnitude of the information leakage problem (req. 6). Currently, to reduce the risk of leaks both the community of users employing pseudonyms and the mix of actions and operations they perform in a Grid need to be large. Ideally, every action on the Grid would use a different pseudonym and use it only once. This makes it difficult to correlate the different actions of any single user. On the other hand, even different one-time pseudonym identifiers may be correlated if they are used from the same IP address and this address is not used by any other pseudonym user. The pseudonymity system may thus only partially counteract the leakage problem. Ultimately, the users themselves are required to actively reduce such risks.

The large groups needed for preventing the correlation of actions to a single user also pose a problem. For example using pseudonyms for file access means that the access has to be based solely on the groups and attributes of the user. If the groups are large, it means that there are many people that have access to the files, thus there is less privacy of the files. Also there is less compartmentalization of users and thus in case of a compromise of a user there is a bigger potential for damage. In the end the group size is a balancing act between the users' need for identity hiding and resource security.

Also, some legislative concerns must be addressed. For example, in some countries the law expects site administrators to know the real identities of the users. The pseudonymity system records the link between the real user identity and the pseudonym but this may not fulfill the requirements of some regulations. Unless some governmental identity escrow is available, this effectively bans the usage of pseudonyms within these jurisdictions.

For large-scale deployments the certification policy and the pseudonym user's authentication sequence has to be approved by a Grid Policy Management Authority (GridPMA). A probable conflict with most site policies is the generation of long-term pseudonym credentials for anonymizing long running jobs, as a long-term credential implies a higher security risk than an ephemeral one. A compromised long-term credential has a longer window of opportunity for misuse.

VIII. Future Work

The near term work is to implement prototypes of the pseudonymity service and the client software compatible with the protocol defined in this paper. It will allow us to gain practical experience of the system and the identified problem areas, especially the information leakage problem.

Another important goal to pursue is to ensure the mix of operations and pseudonymous users is sufficiently broad to prevent correlation attacks. Also, the connection source tracking needs to be investigated. To this end, Grid portals are ideal: using pseudonyms through a portal effectively prevents IP addresses from being used for tracking users provided that also job results are accessed and stored through the portal using a pseudonym. The benefits and drawbacks of including portals into the overall architecture will be explored.

Acknowledgment

The authors wish to hereby thank Olle Mulmo, Ákos Frohner and David Groep for fruitful discussions, ideas and feedback for this work.

References

[1] EGEE Design Team, "EGEE Middleware Architecture And Planning (Release 1)," Enabling Grids for E-science in Europe, Tech. Rep., Aug. 2004. [Online]. Available: https://edms.cern.ch/document/476451/1.0/architecture.pdf

[2] O. Mulmo, "Global Security Architecture for Web and Legacy Services," Enabling Grids for E-science in Europe, Tech. Rep., Sep 2005. [Online]. Available: https://edms.cern.ch/document/1715618922/1.2

[3] A. Pfitzmann and M. Hansen, "Anonymity, Unlinkability, Undetectability, Unobservability, Pseudonymity, and Identity Management - A Consolidated Proposal for Terminology v0.31," Feb. 2008. [Online]. Available: http://dud.inf.tu-dresden.de/Anon_Terminology.shtml

[4] J. Hahkala, H. Mikkonen, M. Silander, and J. White, "Requirements and Initial Design of a Grid Pseudonymity System," in *Proceedings of the 2008 High Performance Computing & Simulation Conference (HPCS)*, Nicosia, Cyprus., June 2008.

[5] Internet2, "Shibboleth Project Web Site," (Referenced on 30.10.2008). [Online]. Available: http://shibboleth.internet2.edu

[6] J. Hughes and E. Maler, "Security Assertion Markup Language (SAML) 2.0 Technical Overview, Working Draft 04," Apr. 2005, http://www.oasis-open.org/committees/security/.

[7] The Globus Alliance, "GridShib Project Web Site," (Referenced on 30.10.2008). [Online]. Available: http://gridshib.globus.org

[8] V. Welch, T. Barton, K. Keahey, and F. Siebenlist, "Attributes, Anonymity, and Access: Shibboleth and Globus Integration to Facilitate Grid Collaboration," in *4th Annual PKI R&D Workshop: "Multiple Paths to Trust"*, NIST Gaithersburg MD, USA, Apr. 2005.

[9] J. Camenisch and A. Lysyanskaya, "An Efficient System for Non-transferable Anonymous Credentials with Optional Anonymity Revocation," in *EUROCRYPT '01: Proceedings of the International Conference on the Theory and Application of Cryptographic Techniques*. London, UK: Springer-Verlag, 2001, pp. 93–118.

[10] J. Camenisch and E. V. Herreweghen, "Design and implementation of the idemix anonymous credential system," in *CCS '02: Proceedings of the 9th ACM conference on Computer and communications security*. New York, NY, USA: ACM Press, 2002, pp. 21–30.

[11] BEA Systems, BMC Software, CA, IBM Corporation, Layer 7 Technologies, Microsoft Corporation, Novell, and VeriSign, "Web Services Federation Language (WS-Federation), Version 1.1," Dec. 2006. [Online]. Available: http://www.ibm.com/developerworks/library/specification/ws-fed/

[12] S. Tuecke, V. Welch, D. Engert, L. Pearlman, and M. Thompson, "Internet X.509 Public Key Infrastructure (PKI) Proxy Certificate Profile," RFC 3820, IETF, June 2004. [Online]. Available: http://www.ietf.org/rfc/rfc3820.txt

[13] R. Alfieri, R. Cecchini, V. Ciaschini, L. dell'Agnello, Á. Frohner, A. Gianoli, K. Lőrentey, and F. Pataro, "VOMS, an Authorization System for Virtual Organizations," in *1st European Across Grids Conference*, Santiago de Compostela, Spain, Feb. 2003.

[14] L. Pearlman, V. Welch, I. Foster, C. Kesselman, and S. Tuecke, "A Community Authorization Service for Group Collaboration," in *POLICY '02: Proceedings of the 3rd International Workshop on Policies for Distributed Systems and Networks (POLICY'02)*. Washington, DC, USA: IEEE Computer Society, June 2002, pp. 50–59.

[15] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile," RFC 5280, IETF, May 2008. [Online]. Available: http://www.ietf.org/rfc/rfc5280.txt

[16] S. Farrell and R. Housley, "An Internet Attribute Certificate Profile for Authorization," RFC 3281, IETF, Apr. 2002. [Online]. Available: http://www.ietf.org/rfc/rfc3281.txt

[17] EJBCA, "The J2EE Certificate Authority Web Page," (Referenced on 30.10.2008). [Online]. Available: http://ejbca.sourceforge.net/

[18] J. Basney, M. Humphrey, and V. Welch, "The MyProxy online credential repository," *Software: Practice and Experience*, vol. 35, no. 9, pp. 801–816, July 2005.

[19] P. A. et al., "CREAM: A Simple, Grid-accessible, Job Management System for Local Computational Resources," in *Proceedings of the 2006 Computing in High Energy and Nuclear Physics Conference (CHEP06), Mumbai, India*, Feb. 2006.

[20] SWITCH, "Short Lived Credential Service (SLCS) Web Page," (Referenced on 30.10.2008). [Online]. Available: http://www.switch.ch/grid/slcs/

[21] EGEE-II, "MJRA 1.4: Shibboleth Interoperability Through a Short-Lived Credential Service (SLCS) v0.96," Nov. 2006. [Online]. Available: https://edms.cern.ch/document/770102/1

[22] M. Nystrom and B. Kaliski, "PKCS #10: Certification Request Syntax Specification Version 1.7," RFC 2986, IETF, Nov. 2000. [Online]. Available: http://www.ietf.org/rfc/rfc2986.txt

[23] B. Kaliski, "Privacy Enhancement for Internet Electronic Mail: Part IV: Key Certification and Related Services," RFC 1424, IETF, Feb. 1993. [Online]. Available: http://www.ietf.org/rfc/rfc1424.txt

[24] M. Pitkänen and H. Mikkonen, "Initalizing Mobile User's Identity From Federated Security Infrastructure," in *Proceedings of the Second International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM '08)*, Valencia, Spain., Sept.–Oct. 2008, pp. 390–394.