Federated Access Management for Collaborative Network Environments: Framework and Case Study

Carlos E. Rubio-Medrano, Ziming Zhao, Adam Doupé and Gail-Joon Ahn
The Laboratory of Secure Engineering for Future Computing (SEFCOM)
Arizona State University
Tempe, AZ, USA
{crubiome, zmzhao, doupe, gahn}@asu.edu

ABSTRACT

With the advent of various collaborative sharing mechanisms such as Grids, P2P and Clouds, organizations including private and public sectors have recognized the benefits of being involved in inter-organizational, multi-disciplinary, and collaborative projects that may require diverse resources to be shared among participants. In particular, an environment that often makes use of a group of high-performance network facilities would involve large-scale collaborative projects and tremendously seek a robust and flexible access control for allowing collaborators to leverage and consume resources, e.g., computing power and bandwidth. In this paper, we propose a federated access management scheme that leverages the notion of *attributes*. Our approach allows resource-sharing organizations to provide distributed *provisioning* (publication, location, communication, and evaluation) of both attributes and policies for federated access management purposes. Also, we provide a proof-of-concept implementation that leverages *distributed hash tables* (DHT) to traverse chains of attributes and effectively handle the federated access management requirements devised for inter-organizational resource sharing and collaborations.

1. INTRODUCTION

Traditionally, collaborative information sharing heavily relies on client-server or email-based systems. By recognizing the inherent deficiencies such as a central point of failure and scalability issues, several alternatives have been proposed to support collaborative sharing of resources, including Grid computing, Peer-to-Peer (P2P) networking [11] and Cloud computing [27]. Given all the diverse contexts of collaboration, achieving effective access control is a critical requirement. The sharing of sensitive information and resources is necessarily to be highly controlled by defining what is shared, who and under which conditions is allowed to share. In particular, users without pre-existing relationships may try to collaborate and request the information. It is required for a resource provider to be able to cope with a large number of collaborators and guarantee the information and resources be released only to trusted collaborators within the community. In addition, resources are constructed with various types and domain policies, and each collaborating party may enforce security policies in their systems with different degrees of assurance. Therefore, building systematic mechanisms for sharing resources across collaborative network environments is indeed an important challenge.

Furthermore, organizations including private and public sectors have recognized the benefits of being involved with inter-organizational, multi-disciplinary collaborative projects that may require diverse resources shared among participants, e.g., data, computation time, storage, etc. In particular, an environment that often makes use of a group of high-performance network facilities would involve large-scale collaborative projects and tremendously seek a robust and flexible access control for allowing collaborators to leverage and consume resources. For example, under the US Department of Energy (DoE), numerous research laboratories and scientists have collaborated and performed their experiments demanding specific network bandwidth and designated computing resources from each other. They even exchanged data and resources with other foreign researchers, which lead them to utilize high-performance network environments such as ESnet [29], GÉANT [7], and NORDUnet [21]. Despite the necessary administrative tasks such as resource scheduling and provisioning, there is a need to properly mediate the way such resources are to be safely shared in the context of collaborations. Because most of these providers depict their own in-house *authentication* and *authorization* services, a well-defined, inter-organizational and implementation-independent approach is needed. With this in mind, this paper presents our approach to address the aforementioned challenges by leveraging the concept of *attributes*: observable properties that are exhibited by access control entities, e.g., users and protected resources, that become relevant under a given security context [19], focused on DoE networks and their collaborators' networks. Using attributes as an underlying framework, we propose an approach based on the concept of a *federation* between participant organizations, allowing them to *provision*: specify, publish, locate, and communicate attributes for federated access management purposes in a distributed way, thus allowing for the specification and automated evaluation of both local (intra-domain) and federated (inter-domain) policies. With this in mind, this paper makes the following contributions:

- We formulate the main components involved in federated access management. We show how attributes in the *lo-*

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
SACMAT'15, June1–3, 2015, Vienna, Austria.
Copyright © 2015 ACM ISBN 978-1-4503-3556-0/15/06 ...$15.00.
DOI: http://dx.doi.org/10.1145/2752952.2752977.

cal context can be leveraged in a *federated* context such that access permissions for inter-organizational resource sharing can be properly granted.

- We also provide a well-defined description of attributes, which includes the use of data types, standardized names, and run-time values, so that participants can unambiguously use those to define inter-organizational attributes and policies for federated access management purposes.
- We propose an attribute generation approach by means of a set of so-called *attribute derivation rules* (AD-Rules). Moreover, we also introduce *attribute derivation graphs* (AD-Graphs) that allow to compose AD-rules.
- We provide an initial step toward automated attribute discovery based on *distributed hash tables* (DHT) [28], which allows for efficient discovery and retrieval of attributes within a federated and distributed context. In addition, we provide a *proof-of-concept* implementation of our attribute provisioning scheme, including an evaluation approach that shows the feasibility of our approach for real-life implementations.

This paper is organized as follows: we start by articulating problem statements and technical challenges with respect to federated access management in Section 2. Then, we describe our approach in Section 3 followed by the *proof-of-concept* implementation and evaluation results in Section 4, which shows the practicability of our approach for supporting real-world collaborations among the DoE-affiliated high-performance network facilities. We overview the related work in Section 5 and discuss some relevant topics related to our approach as well as matters for future work in Section 6. Finally, Section 7 provides concluding remarks.

2. BACKGROUND

As previously mentioned, DoE-affiliated high-performance network facilities have identified the need to provide automated means for resource sharing between different administrative (and security) domains. As an example, the Open Grid Forum [9] introduced a multi-organizational effort called the *network services interface* (NSI) [23] that is composed of a set of well-defined protocols that allow participants to collaborate on research endeavors by implementing inter-organizational *services* in an automated way. The protocol devised for a given NSI service is implemented by so-called *network service agents* (NSA) which are expected to support all service-related tasks within the context of a given administrative domain. Fig. 1 shows an example depicting a data transfer between two hosts that are located within the administrative boundaries of two different organizations and whose networking path involves the participation of a third network serving as a bridge. In this example, each participating network implements the protocol devised for the NSI *connection* service by means of a dedicated NSA. Following such a protocol, a connection request R is first serviced by the *local* NSA where R originates (NSA_1 in Fig. 1). On each network, the local NSA is in charge of reserving local ports and bandwidth to create a connection within its network boundaries. NSA_1 is also in charge of contacting the other NSAs involved in serving R (NSA_2 and NSA_3) so that they can make reservations within their inner networks. In addition, all involved NSAs must handle network connections

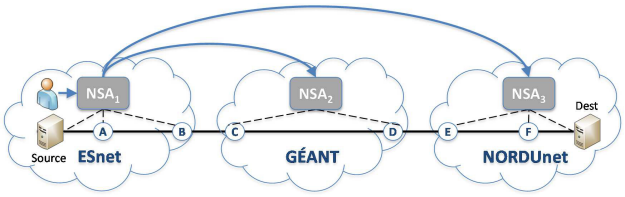

Figure 1: An NSI inter-domain data transfer: An end-user presents credentials to the software agent labeled as NSA_1, requesting for data stored in a host under the ESnet domain to be transferred to a host located under the NORDUnet domain, which is in turn managed by the agent known as NSA_3. The GÉANT domain (managed by NSA_2) serves as a bridge for the connection purpose.

between independent networks by physically interconnecting any relevant *service termination points* (STPs), which are abstract (high-level) representations of actual network ports and are labeled from A to F in Fig. 1. Once the connection path between the source and destination hosts is completed, the requested data transfer takes place.

In this collaborative setting within DoE-affiliated high-performance network facilities, we articulate the following the federated access management requirements that need to be accommodated:

1. Participating organizations should be allowed to define its own set of federated access management policies governing the way a given service, e.g., the aforementioned connection service, is provided in response to both local and external requests. As an example, ESnet may want to give priority over local resources to requests originated within its local domain.
2. Participating organizations may also agree on a set of inter-domain federated access management policies governing a subset of service interactions between them. As an example, ESnet and GÉANT may agree on a policy allowing for a collaborative project between both organizations to be guaranteed with high quality of service by reserving sufficient bandwidth for data transfers.
3. Participating organizations may implement their own in-house federated access management systems, which may in turn handle their own set of local credentials and possibly their own set of locally-relevant attributes. This may potentially result in problems such as attribute incompatibility, or different attributes being assigned to the same access control entity by different domains, e.g., users getting credentials issued by each service in response to their access request, possibly result in a large set of credentials to be handled. However, organizations may not favor a complete replacement of their current authentication and access control modules, as such an effort may involve considerable financial and organizational effort. As an example, ESnet may find it difficult to replace the current set of locally-issued credentials for the more than 40 research institutions currently being served by the network [29].
4. Every access control entity, e.g., end-users and protected resources, involved in serving a given access request is expected to provide a set of security-relevant properties, e.g., user credentials or resource descriptors, which may

have in turn been assigned either by its local security domain or by an external one, in such a way that proper policy evaluation based on such properties can take place. If a given entity fails to show those properties, even when they may have been legitimately assigned beforehand, the evaluation of a relevant access control policy may fail thus causing legitimate access to be denied as a result. In practice, such properties are commonly assumed to exist at policy evaluation time, either locally or remotely, e.g., stored in a dedicated centralized database. In addition, security-relevant properties may be in turn *derived* by processing other related properties. As an example, user credentials may be used to obtain the set of collaborative projects the user is involved in, without requiring the user to explicitly enumerate them, granting access only to the resources those projects are entitled to. However, existing infrastructures are not capable of seamlessly locating and transforming security properties in a distributed setting such as the one depicted in Fig. 1, which is composed of several independently-managed security domains.

5. Finally, existing federated approaches for security, e.g., OpenID [26] or Shibboleth [18], are focused on authentication: support for authorization is limited and is mostly left for third parties to implement from scratch, e.g., attribute and policy definition, discovery, and evaluation.

Consider the example of the three participant organizations on the data transfer requests depicted in Fig. 1. Each organization agrees on an inter-organizational policy P_1 that allows for data transfers between participants, e.g., from STPs A to F, if all of the following conditions are met: first, the requester is a member of a collaborative group labeled as G. Second, the size of the data to be transferred is less than or equal to 10 Tb. Third, the available bandwidth on each network is higher than or equal to 1 Gbit/s.

3. APPROACH

A well-defined approach for the specification and provisioning of both policies and security-relevant properties (attributes) is critical to enable inter-organizational resource collaboration—specifically an approach that goes beyond credential-sharing by including *heterogeneous* attributes obtained from different federated access management entities, which may have been assigned by different security domains. As depicted in a recent report by the *National Institute for Standards and Technology* (NIST) [19], proper provisioning mechanisms may become a crucial component for the successful development of new technologies and new infrastructures based on attributes. Inspired by recent successful approaches for federated authentication, we propose a federated and distributed solution for the specification, location, generation, and communication of both attributes and policies for federated access management purposes that is intended to support automated resource sharing and the establishment of collaborative projects among independent organizations, each possibly implementing their own security domain as well as their own dedicated federated access management infrastructure. A graphical depiction of our approach is shown in Fig. 2: a locally-defined attribute a_1 belonging to a given user is transformed into a series of federation-recognized attributes (a_2, a_3, a_4) that are in turn provided by other organizations engaged in a federation and may be used for access control decisions.

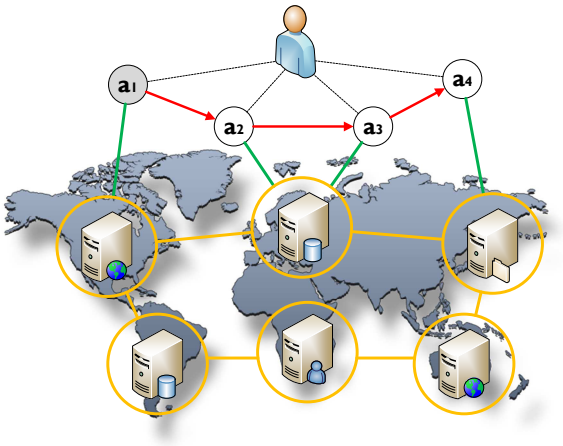

Figure 2: A federated access management framework: the local attribute a_1 is transformed into the federated attributes labeled as a_2, a_3 and a_4 by leveraging *attribute derivation rules* (AD-Rules) implemented by remote peers.

In order to participate in our proposed federation, participating organizations under DoE-affiliated high-performance network facilities must fulfill the following:

1. *Attribute identification*: Participating organizations are to identify security-relevant properties within their local domains that may serve as *local* attributes for federated access management purposes. As an example, in Fig. 1, ESnet should identify any relevant metadata belonging to the data to be transferred that can be used to obtain the properties that are relevant under policy P_1, e.g., its size in bytes.

2. *Attribute mapping*: Participants must *map* local attributes onto a set of publicly-known *federated* attributes to be used in the context of an inter-domain collaboration. Following our running example, a standard definition of an attribute depicting the size of a given chunk of data, e.g. a convention name, size unit, etc., would allow the specification and enforcement of policies across organizational domains. Because participant organizations may in turn have their own in-house definitions for local attributes, e.g., names, data ranges, etc., a consensual inter-organizational definition of federated attributes is needed. With this in mind, existing approaches based on ontological representations such as the one proposed by Paci, et al. [24], may be utilized to mitigate the existence of different attribute definition schemes, also known as *attribute heterogeneity*. Due to the nature of DoE-affiliated network facilities, we assume such a common knowledge base on attributes has been established beforehand.

3. *Attribute discovery*: Participants should allow organizational peers to leverage the federated attributes they provide by means of a discovery service. Following our running example, ESnet and GÉANT should be able to locate each other's attributes when constructing an inter-domain policy for shared connections.

4. *Federated access management administration*: Organizations should implement a proper administrative model for creating, updating, and removing both local as well as federated attributes and federated access management

policies that restrict access to protected resources within collaborative projects.

5. *Policy conflict resolution*: Finally, participants should be able to detect and resolve conflicts when constructing federated access management policies, e.g., contradictory or redundant rules, etc. As an example, let's assume the ESnet domain also provides a local policy P_2 that allows for intra-domain transfers to take place, e.g., from STPs A to B in Fig. 1, if the requesting end-user is a member of a certain local group and the data to be transferred has not been obtained from a particular server storing sensitive data located within the network scope. In such a setting, the inter-organizational policy P_1 depicted in Section 2 may be in conflict with P_2 if the data to be transferred comes from such a data-sensitive server, as P_1 may authorize the transfer but P_2 may deny it.

With respect to the evaluation of federated access management policies, participants are responsible for the following:

1. *Policy retrieval*: Upon receiving a given federated access management request R, participants should retrieve the set L containing local policies relevant to R. Following our running example, ESnet should retrieve the P_2 policy regarding data transfers originating in its local domain.

2. *Attribute provision*: Participants should provision any local and federated attributes as specified in the policies contained in L. To enable this provisioning, participants are to make their federated attributes available for other peers to provision upon request. In addition, participants should make (allowable) attribute transformations available to their peers. Following our running example, GÉANT transforms the credentials presented by an end-user in the ESnet domain into an attribute depicting membership to the federated collaborative project G that is required in the inter-organizational policy P_1.

3. *Policy dispatch*: Participants should dispatch policy evaluation requests for relevant federated policies I that are relevant to R. Conversely, participants should evaluate and provide results for any policy evaluation requests they receive as part of a request evaluation process initiated by a federated peer. Back to our running example, participant networks should retrieve all attributes relative to a connection request that happen to be under the scope of their local security domain and should dispatch both attribute and policy evaluation requests, e.g., P_1, to the other networks involved in the construction of the network path.

4. *Results aggregation*: Finally, the policy decisions for both sets L and I should be derived and combined to produce a final decision for the request R, which is to be communicated to the requesting entity, e.g., the end-user under the ESnet domain in Fig. 1.

3.1 Model Description

We start the discussion of our model for federated access management by defining the following components:

- *actors* are end-users (i.e. human agents) or subjects (i.e. computer processes) acting on behalf of users;
- *targets* are the protected resources within a security domain;

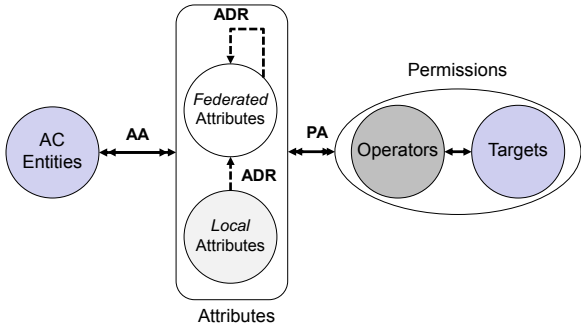

Figure 3: A model for federated access management: attributes are related to entities, e.g., end-users and protected resources, by means of the *attribute assignment* (AA) relation. Attributes (both *local* and *federated*) may be transformed into *federated* attributes by means of AD-Rules (ADR). Permissions are related to attributes by means of the *permission assignment* (PA) relation.

- *context* is the running (executing) environment, e.g. operating system, supporting platform, etc., where a given request is issued and/or served.

Fig. 3 shows a visual representation of our model: attributes are related to access control entities by means of the *attribute assignment* (AA) relation, allowing each entity to exhibit many different attributes and a single attribute to be potentially exhibited by more than one entity. *Federated* attributes are publicly-known attributes that may be relevant in the context of a given collaboration project. *Local* attributes are related to federated attributes through *attribute derivation rules* (AD-Rules), which are shown as directed arrows in a dotted line in Fig. 3. The precise definition of such AD-Rules, e.g., how local attributes are ultimately related to federated ones, is defined by peers within the context of a given collaboration. As we will discuss in Section 3.4, AD-Rules can be organized into a graph-like structure known as an *attribute derivation graph* (AD-Graph), which provides a representation of how attributes are related to permissions, which are in turn related to federated attributes by means of the *permission assignment* (PA) relation. Permissions are depicted as a combination of a protected source (target) and an operation that can be performed on it. A given attribute may be related to one or more permissions, and a given permission may be related to one or more attributes.

A description of our proposed approach is shown in Fig. 4. The basic components are actors (ACT), targets (TAR), and context (CON), which together construct the set E of access control entities. Moreover, we also consider the sets operations (OPER) and permissions (P). We define the sets names (N) and values (V), which are used for defining the sets of attributes (A) and federated attributes (F). The relationships between the elements of our model are described by defining the *attribute assignment* (AA) and *permission assignment* (PA) relations, as well as our proposed AD-Rules.

The definition of AD-Graphs is based on the concepts of graph theory and the definition of AD-Rules. The access control decision process is modeled by functions *provisionedAttributes*, *expectedAttributes*, *relatedPermissions*, and *checkAccess*. Function *provisionedAttributes* calculates the set of attributes that can be provisioned from a given AD-Graph

based on the local and federated attributes initially exhibited by a set of access control entities. Function *expectedAttributes* returns the set of attributes that are related to a given permission, by inspecting the PA relation. The mirror function *relatedPermissions* returns the set of all permissions that are associated in the PA relation with a given attribute. Finally, function *checkAccess* implements the authorization checking functionality by first calculating the set of attributes provisioned by the entities in a given access control request and comparing it with the set of attributes that are related to the requested permission, which is only granted if the set of provisioned attributes (obtained from the *provisionedAttributes* function) is a subset of the set of attributes related to such permission, which is obtained from the *expectedAttributes* function.

3.2 Attributes

We define attributes as an abstraction of *security-relevant* properties that are exhibited by access control entities, namely, actors, targets, policies, and any applicable context. Their physical nature, e.g., if the attribute represents a file's metadata or an end-user credential, and the way those attributes are collected from the access control entities remain dependent on each organizational domain.

As shown in Fig. 4, we define attributes to have the following three components: (1) a data *type*, which restricts the nature and the possible range of values defined for the attribute, (2) a *name*, which is later used for defining AD-Rules on them and is defined in the context of a given inter-organizational setting, and (3) a *value*, which is used when evaluating such AD-Rules. Examples of attributes include: <Double, *data.size*, 100.0>, <String, *data.source*, "*server.esnet*">, and <Date, *system.date*, "*10-10-2015*">.

3.3 Federated Attributes

Federated attributes are obtained by processing local attributes from access control entities under a given organizational domain. Such processing is to be modeled through the AD-Rules, thus allowing federated attributes to be related to access rights (permissions).

As an example, AD-Rules may provide functionality intended to validate a given local attribute by inspecting its value component and producing a proper federated attribute as a result. Thus, a validated federated attribute ensures that a given collaboration state remains secure.

As described in Section 3.1, permissions can be assigned to federated attributes, which then serve as a layer of association between local attributes and permissions defined in another organizational domain for collaborative purposes. Such a layer helps identify the local attributes that may be involved in granting a given inter-domain permission, as well as the set of constraints represented by AD-Rules that may be involved in such a process. Moreover, our approach allows for AD-Rules to take federated attributes as an input or may also take both local as well as federated ones as an input to produce federated attributes as a result, as depicted in Fig. 3, thus allowing for expressing richer inter-domain policies based on processing already existing federated attributes.

3.4 Attribute Derivation Rules and Graphs

As introduced in Section 3.1, *attribute derivation rules* (AD-Rules) are expected to provide a mapping between local attributes and federated attributes. For this purpose, AD-Rules are said to be *non-injective**, as two or more elements from an input set of attributes (domain) may be mapped to the same element in the output set (co-domain).

In addition, AD-Rules can be *chained* together to produce a graph-like structure showing how attributes can be provisioned. Such *attribute derivation graphs* (AD-Graphs) are *directed*, because AD-Rules represent unidirectional edges (due to their nature as functions). Moreover, AD-Graphs are also *weakly* connected, as there is no requirement for all nodes (attributes) to be connected to each other. Finally, AD-Graphs are also possibly *cyclic*, as a customized

- ACT, the set of actors.
- TAR, the set of targets.
- CON, the set of context instances.
- OPER, the set of operations.
- P $\subseteq$ TAR $\times$ OPER, the set of permissions.
- E = ACT $\cup$ TAR $\cup$ CON, the set of access control entities.
- N, the set of names.
- V, the set of values.
- T, the set of data types.
- A = { a | a = <*type*, *name*, *value*>where *type* $\in$ T, *name* $\in$ N, *value* $\in$ V, the set of attributes.
- F $\subseteq$ A, the set of federated attributes.
- AA $\subseteq$ A $\times$ E, the *attribute assignment* relation mapping attributes with a given access control entity.
- PA $\subseteq$ P $\times$ A, the *permission assignment* relation mapping permissions and attributes.
- ADR = { r | r: $2^A \rightarrow 2^F$}, the set of attribute derivation rules mapping sets of attributes to sets of federated attributes.
- ADG, the set of directed, weakly connected, and possibly cyclic attribute derivation graphs. A graph g =<NODES, ARCS>$\in$ ADG if NODES $\subseteq 2^A$ and ARCS $\subseteq$ ADR. We say (n_1, *arc*, n_2) $\in g$ if n_1, $n_2 \in$ NODES and *arc* $\in$ ARCS and $n_1 \subseteq$ *domain*(*arc*) and $n_2 \subseteq$ *codomain*(*arc*).
- *provisionedAttributes*: $2^E \times$ ADG $\rightarrow 2^A$, a function mapping a set of entities $E' \subseteq E$ with the set of attributes that the entities in E'can provision from a given ADG g. An attribute f is said to be provisioned by an entity $e \in E'$if there exists a set of attributes A′= { $a \mid a \in$ A, $e \in E'$, (a, e) $\in$ AA } $\subseteq$ A and a set of paths P = {$p \mid p = x_0, x_1, ...x_n$, n $\geq$ 0} in g such that $\forall$ p $\in$ P, $x_0 \in$ A′and $x_n = f$, and $\forall$ x_i, x_j in p, $1 \leq i < n$, $j = i + 1$, $\exists$ $r \in$ ADR such that $r(x_i) = x_j$.
- *expectedAttributes*: P $\rightarrow 2^A$, a function returning the set of attributes that are related to a given permission p. Formally, returns all $a \in$ A such that (p, a) $\in$ PA.
- REQ = {*req* = <*act*, *p* = <*tar*, *oper*>, *ctx* >| *act* $\in$ ACT, $p \in$ P, *ctx* $\in$ CON}, the set of access control requests, allowing an actor *act* to request for a permission p to be granted.
- *checkAccess*: REQ $\times$ ADG $\rightarrow$ {*true*, *false*}, a boolean function that checks if a given request *req* = (*act*, *p* = (*tar*, *oper*), *ctx*) $\in$ REQ should be granted or denied based on a given attribute derivation graph *adg*. Formally, the function returns *true* if *provisionedAttributes*({*act*, *tar*, *ctx*}, *adg*) $\subseteq$ *expectedAttributes*(p), and returns *false* otherwise.

Figure 4: A model description of our approach.

*A function f: A $\rightarrow$ B is said to be *injective* or *one-to-one*, $\forall$ a, $a' \in$ A, a $\neq$ a′$\Rightarrow$ $f(a) \neq f(a')$.

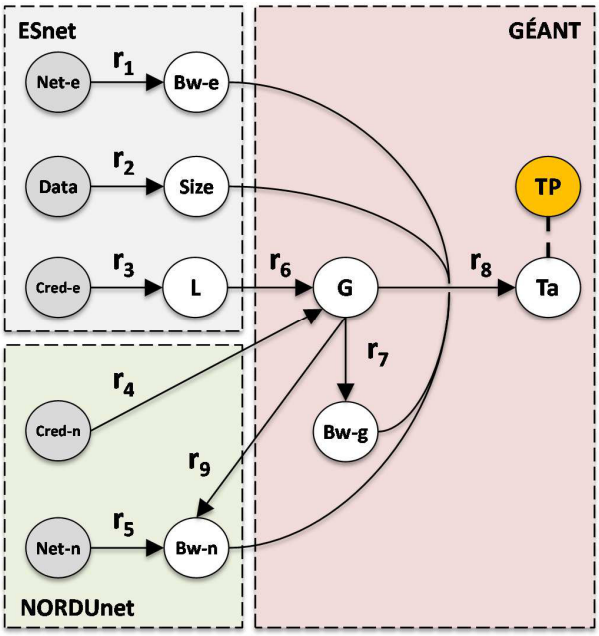

Figure 5: A distributed AD-Graph depicting policy P_1: local attributes (shown in grey) are transformed into federated ones (shown in white). As an example, the AD-Rule labeled as r_8 transforms attributes G (group membership), *Size* (data size) and *Bw-e*, *Bw-g*, *Bw-n* (bandwidth) into the federated attribute *Ta* that is related to the *TP* (data transfer) permission.

chaining of AD-Rules may end up introducing a cycle in the produced AD-Graph.

AD-Graphs may also support collaborative processing by allowing a division into proper subgraphs, each subgraph implemented in a different security domain: as mentioned in Section 2, each participating domain is in charge of defining its own permissions, local and federated attributes, as well as the AD-Rules and AD-Graphs to generate those. AD-Graphs can be modeled as a distributed graph: a given AD-Graph G defined for a federation F may be divided into a set of subgraphs G'_1, G'_2, ... G'_n, such that each G'_i is to be processed by a different domain in F.

As an example, the AD-Graph in Fig. 5 implements the inter-organizational policy P_1 described in Section 2 as follows: the ESnet local attribute *Cred-e*, which depicts a locally-issued credential, is transformed by the AD-Rule labeled as r_3 into the federated attribute L that features membership to a local group within ESnet. L is subsequently processed by the AD-Rule r_6 in the GÉANT domain, producing the federated attribute G, which in turn depicts membership to an inter-organizational collaborative group. Later, G, along with attributes *Size*, *Bw-e*, *Bw-g*, and *Bw-n* are taken as input for the AD-Rule labeled as r_8, producing the *Ta* attribute as a result. This attribute features an access token related to the *TP* permission authorizing the data transferring process shown in Fig. 1. Such a permission is included in Fig. 5 for the illustrative purposes.

Leveraging the previous definitions, the problem of resolving an access request to a shared resource within a federation can be first modeled as a path traversal problem within a distributed graph: determining if there exists a path between a set of starting nodes (local attributes) and a given ending node (federated attribute). Then, determining if such a federated attribute grants the requested permission over the desired resource. A general procedure for resolving an access request, derived from the model shown in Fig. 3: given a federation F, a permission P, and a set of input attributes I the procedure starts by obtaining the set of *expected* federated attributes granting P, e.g., by parsing local and federated access management policies. If the *required* set is found to be a subset of I—that is, I contains the attributes required for P, access is granted. Otherwise, the procedure extracts a set of paths from an AD-Graph in F, each of these paths starting with an attribute in the set I and ending with an attribute in the *required* set. Then, each path is traversed by executing each of the included AD-Rules. If new federated attributes are generated, and such attributes happen to include the attributes in *required*, access is granted and the procedure terminates. Otherwise, access is denied.

3.5 Attribute Provisioning

In our approach, attribute provisioning is crucial to handle federated access management requests in the context of inter-organizational resource sharing. Such a process includes allowing for participating organizations to know about the AD-Rules that are implemented by other organizations and are involved in a given AD-Graph G. Concretely, participants need up-to-date information about G so that they can extract correct paths within G that can produce the desired federated attributes. With this in mind, attribute provisioning can therefore be divided into two process: path discovery and path traversal.

The *path discovery* process allows for each organization to distribute information about its locally-implemented AD-Rules to the federated access management federation, so that they can potentially maintain a representation of G for path calculation. However, there are several practical challenges: first, each organization needs to be notified when changes to G occur, e.g. adding or removing a given AD-Rule, which may create a large set of communication messages between participants. Second, there is an added maintenance cost, e.g. processing time, that participating organizations must incur for handling and maintaining an up-to-date G. Finally, storage efficiency may become an issue when a large G must be locally maintained. An alternative approach would be creating a central database storing G, along with a set of replicas for enhanced availability. However, such a scheme may suffer from service bottlenecks and consistency issues when communicating updates to the replicas. In addition, a centralized server may become the subject of a *denial of service* (DoS) attack, which could certainly limit the availability of the overall attribute provisioning scheme, thus potentially preventing participating organizations from serving federated access management requests. With this in mind, there is a need for a distributed approach that allows for participating organizations to release information about the AD-Rules they implement in such a way that the administration burden, e.g., number of communication messages, is significantly reduced. In addition, such an approach should also prevent organizations from having to store a complete AD-Graph locally for path discovery purposes and should provide support against attacks targeting a single point of failure. We present an implementation tailored for meeting such goals in Section 4.

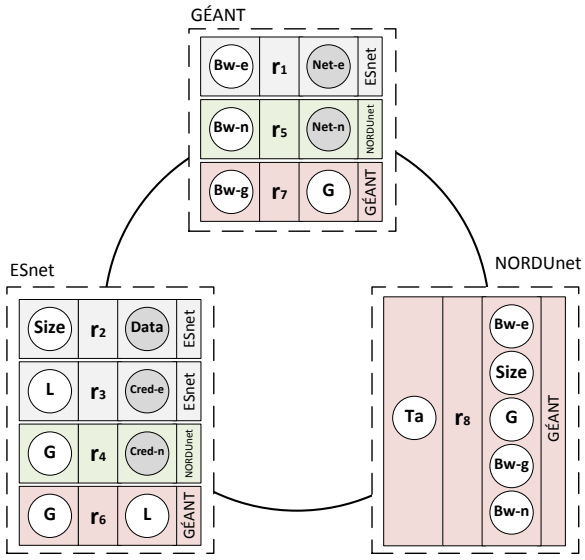

Figure 6: An illustrative DHT *ring* depicting the AD-Graph of Fig. 5: federated peers store *entries* containing information about the AD-Rules implemented by other peers in the context of federated access management.

Following the model described in Fig. 3, the path traversal process allows participating organizations to invoke the AD-Rules included in a given path p in G that may ultimately produce a given federated attribute. Invocation of such AD-Rules should be done by following a sequence starting from the first AD-Rule in p up to the last one. Each time an AD-Rule is executed, the produced set of attributes is added to a set of input attributes for the next AD-Rule in the sequence. In addition, the invocation of an AD-Rule r enables to locate the federated domain implementing r, the set of input attributes, as well as the set of produced attributes. A request for the invocation of r should include the set of attributes that serve as its input. Finally, the attributes produced by r, if any, should be then communicated back to the requesting organization.

4. IMPLEMENTATION AND EVALUATION

In this section, we describe our proof-of-concept implementation and evaluation results. We elaborate how we accommodate the concerns described in Section 3.5. Also, we discuss how the path discovery process was implemented with the concept of *distributed hash tables* (DHT) [28]. In addition, we discuss our implementation on the path traversal process which is based on a client-server architecture for the remote invocation of our proposed AD-Rules.

4.1 Path Discovery

Fig. 6 illustrates the path discovery process based on our running example. We allow for participants in a federation F to join a DHT *ring* to publish and retrieve information about the AD-Rules that may produce federated attributes. This process may be in turn decomposed into two inner components, namely, AD-Rule publishing and AD-Rule retrieval.

The procedure for publishing an AD-Rule is conducted as follows: each domain is in charge of inserting an *entry* into the DHT for each AD-Rule they implement for a given AD-Graph under the context of F. Such an entry should include information about the input attributes (either local or federated ones), the name of the AD-Rule, and the set of federated attributes to be produced as a result. Moreover, some information on how to *execute* such AD-Rule should be also provided, e.g., a universal resource locator (URL). As an example, the ESnet domain will publish an entry into the DHT containing information about the AD-Rule r_1, including the local input parameter *Net*, which conceptually depicts information about the current state of the local network, and the federated attribute *Bw*, which provides a standard representation of the current bandwidth capacity. In addition, such an entry should contain a valid URL for other federated peers invoking the AD-Rule r_1 remotely. Following the insertion procedure for DHTs [28], such an entry may end up being stored for future location at a different federated peer, following a hashing scheme based on the standardized naming convention for federated attributes introduced in Section 3.2. In Fig. 6, the entry for the AD-Rule labeled as r_1 (published by ESnet) ends up being stored by the DHT node under the scope of the GÉANT domain. Conversely, AD-Rules may be retired from a given AD-Graph by removing their corresponding entries from a given DHT ring. Recall such procedure may not necessarily remove the production of federated attributes in the context of an AD-Graph, as such attributes may be produced by another AD-Rule in the DHT ring, e.g., removing the entry for the AD-Rule r_6 does not prevent an attribute G from being produced by the AD-Rule labeled as r_4.

The retrieval procedure for entries containing information about AD-Rules is to be conducted as follows: a participating domain D interested in producing a given attribute A may retrieve the set S of entries corresponding to A in the DHT ring, e.g., by hashing the A's identifier. Then, by inspecting the information about AD-Rules contained in S, D must determine if there exists a local or federated attribute under its local domain that can be used as an input parameter to an AD-Rule to produce A. If so, information from the corresponding entry in the set S is retrieved and the AD-Rule is invoked. However, if no suitable entry is found, e.g. all input attributes to the entries in S are out of scope or cannot be locally produced, D may attempt to explore the DHT ring once again for entries producing the attributes taken as an input to the entries in S, thus potentially producing a set P of graph paths in an AD-Graph stored in the DHT. Such a process may be repeated up to the point when no more entries can be obtained from the DHT or a cycle in the AD-Graph stored in the DHT is detected, e.g., when an iteration retrieves entries that were previously retrieved in the past, or a path can be traversed. A path in P is traversed, e.g., by calling the sequence of AD-Rules contained in it, only if it starts with an attribute under the scope of D and ends with the desired attribute A. Considering our running example, an entity under the ESnet domain may provision an attribute *Ta* depicted in Fig. 5 as follows: the DHT featured in Fig. 6 retrieves the entry for the AD-Rule labeled as r_8 from the ring node implemented by NORDUnet. As the input parameters of r_8 are all federated attributes, ESnet inspects the DHT ring once again for determining proper AD-Rules provisioning those attributes. Then, entries generating *Bw-e* (r_1), *Bw-g* (r_7), *Bw-n* (r_5 and r_9), *Size* (r_2) and G(r_4, r_6), are returned. For the federated attribute *Bw-e*, ESnet can provide the local attribute *Net-e* required for r_1, thus creating a *traversable* path within the distributed AD-Graph. In addition, for the federated attribute *Size*,

Table 1: Performance (ms) for policy P_1.

Processing Time	ALT	ATT	OPT
10	411	352	83
50	484	921	1,405
100	531	1,613	2,144
100	492	14,214	14,706
2,500	470	35,201	35,671
5,000	498	85,254	85,652

ESnet can also provide the required local attribute *Data* required for r_2, thus creating a path as well. In the case of G, the entry belonging to r_4 may be discarded as its input attribute (*Cred-n*) is local only to NORDUnet. However, in the case of the entry for r_6, ESnet may inspect the DHT ring once again for an entry producing the input attribute L. Next, the entry for r_3 is returned taking *Cred-e* as an input. Since *Cred-e* is local to ESnet, another traversable path is constructed. With respect to an attribute *Bw-n*, the AD-Rules labeled as r_5 can be also discarded as its input attribute (*Net-n*) is local to NORDUnet. However, r_9 can be used as it takes the federated attribute G as an input, and a path producing G has been already obtained. Similarly, an attribute *Bw-g* can be obtained from r_7 as such an AD-Rule takes G as an input. The setting depicted in Fig. 5 and Fig. 6 allows for the AD-Rules labeled as r_7 and r_9 to disclose network-related information, e.g., bandwidth, only when membership to an inter-organizational project (as depicted by the G attribute) can be shown.

4.2 Path Traversal

Our implementation supports the process of path traversal by allowing for each participant domain D to implement a software agent that is capable of handling requests for the invocation of the AD-Rules that are under the scope of D. Information on locating such agent and invoking the implemented AD-Rules should be consistent with the entries published in the DHT ring described in Section 4.1, e.g., ESnet may provide a TCP/IP agent that implements the AD-Rule labeled as r_1 in Fig. 5 and Fig. 6. For a given path P composed of n entries obtained from a DHT ring, the traversal procedure would include requesting for the execution of each entry starting from the entry at the first position and collecting the attributes produced by the AD-Rule being invoked (if any). The process continues as soon as new attributes are produced on every AD-Rule invocation and finishes either when a given AD-Rule depicted by an entry in the path is not able to produce any attributes or the final entry (at position n - 1) has been executed and the final attributes have been produced as a result.

4.3 Experimental Results

We have implemented the DHT functionality discussed before by leveraging the Open Chord 1.0 API [15]: an open source implementation of the Chord DHT [28] that allows for remote peers to implement a DHT ring by communicating with each other over TCP/IP sockets. In addition, our proposed AD-Rules, as discussed in Section 3.4, were implemented by leveraging a client-server architecture over TCP/IP sockets with the standard `java.net` package.

In the first experiment, we examined the inter-organizational policy P_1 shown in Fig. 5 and Fig. 6. Such an AD-Graph is stored in a DHT ring composed of three nodes and each of them simulates three participating organizations in our running example. In addition, each of these nodes has been augmented with a server module implementing each of the AD-Rules included in the aforementioned AD-Graph. As an example, the ESnet domain was simulated by DHT node as well as a server module implementing the r_1, r_2 and r_3 AD-Rules. In addition, the *processing* time of each AD-Rule included was simulated by introducing a code to halt the execution for a certain period of time. In our experiments, we measured the *average location time* (ALT) for constructing a given path P within the AD-Graph implementing P_1 policy. Also, we measured the *average traversing time* (ATT) for P to return a federated attribute as a result. Finally, we calculated the *overall provisioning time* (OPT) by consolidating both ALT and ATT. Table 1 shows our experimental results when attempting to provision the federated attribute *Ta* simulating an entity in the ESnet domain holding the local attributes *Cred-e*, *Net-e* and *Data* as shown in our example. Since the *length* (number of DHT entries) of the paths under the experiments remains the same, e.g. the same number of involved attributes and AD-Rules, variation in the OPT for each experiment is mostly due to the preconfigured execution time of the AD-Rules included in such paths, whereas the ALT involved in constructing those paths remains manageable.

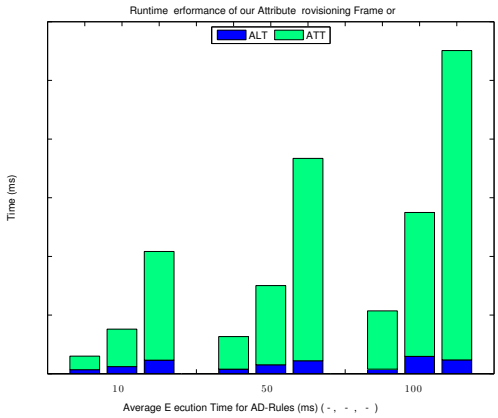

Figure 7: Experimental results for our implementation.

In the second experiment, we measured the response time in provisioning attributes over various AD-Graphs. On each experiment, we produced an AD-Graph depicting a varying number of paths (*branches*) and each of them includes the different number of composing nodes (*links*). In addition, we simulated the execution time of each AD-Rule involved in the produced AD-Graph by using a configurable parameter. We maintained the DHT and server configuration as described earlier. On each experiment instance, we attempted to provision the attribute produced by the DHT entry located at the last node of each path in the simulated AD-Graph. As an example, for a path composed of l nodes, we issue a request for the attribute produced by the DHT entry located at position l-1, assuming that we can include the attribute in the request as the input for the entry depicted in position 0 of the path. Fig. 7 shows our results when constructing AD-Graphs of size (b-l) where b stands for the number of branches and l stands for the number of links on each AD-Graph, e.g., the first three-column set shows the evaluation results when setting up an execution

time of 10 ms for AD-Rules and constructing AD-Graphs of size (5-5), (10-10) and (20-20) respectively.

As described before, we obtained both the ALT and the ATT on each experiment, which are used to calculate the OPT. In the first experiment, most of the overall provisioning time is spent on the path traversal, which is mostly influenced by both the execution time of each AD-Rule in a given AD-Graph, as well as the length of the path. Similarly the ALT observed in the second experiment, while it was also affected by the length of the path, remains just as a small fraction of the OPT, mostly due to the nature of distributed network settings based on DHTs.

5. RELATED WORK

The problem of providing security guarantees in inter-organizational settings has been largely addressed in literature. In particular, several *federated identity* [4] approaches have been introduced to allow partnering organizations to reuse locally-issued credentials when accessing resources located under the scope of an external security domain. As an example, OpenID [26] and Shibboleth [18] have recently gained acceptance in both industry and academia respectively for user-credential sharing. Our approach builds on this idea by allowing participants to exchange federated *attributes*, thus potentially allowing for such attributes to serve as *tokens* granting access to shared resources, in an approach also inspired by Kerberos [20], OAuth [14] and more recently, Facebook Login [8], which strives to allow third-party applications to leverage the user credentials defined for the popular social network to access application-dependent resources.

Moreover, our AD-Rules are inspired by the idea depicted in the credential-discovery protocol proposed by the RT Framework [17], which allows for credentials issued by independent domains to be located and leveraged for federated access management purposes. Similar to the RAMARS Framework [12], our AD-Rules are depicted in a graph-like structure that allows for user-defined attributes to be transformed into a set of widely-recognized credentials. However, the RAMARS framework assumes each security domain implementing the transformation functions may be *partially* trusted by modeling trust in the range [0,1]. In our approach, we assume all federated peers *fully* trust each other for the implementation of the federation goals as discussed in Section 3 and the model presented in Fig. 4, due to the nature of DoE-affiliated high-performance network facilities.

In addition, recent approaches leveraging federated identity for sharing resources include the work of Broeder et al. [2] and Ananthakrishnan et al. [1]. Moreover, Klingenstein [16] and Chadwick and Inman [5] incorporate the concept of end-user attributes with the federated identity. Our approach includes attributes originated from different access control entities rather than considering attributes and credentials from end-users.

In the context of attribute-based models, Zhang et al. [30] introduced their *attribute-based access control matrix*, which extends classical theory in the field of access control to accommodate attributes as well as the notion of *security state*. Moreover, Priebe et al. [25] presented an approach leveraging the concepts of *ontologies* and the *semantic web* in order to formalize the notion of attributes. An approach close to ours was introduced by Covington and Sastry [6], who presented a *contextual attribute access control* (CABAC) model which was realized in mobile applications. However, our approach goes a step further by describing the way such attributes are mapped to access rights (permissions) by means of AD-Rules and AD-Graphs. Recently, a notable approach was proposed by Jin et al. [13], whose approach formalizes a series of attribute-based model families. However, our approach introduces a notion of security token and AD-Rules to capture the mapping between attributes and corresponding access rights.

6. DISCUSSION AND FUTURE WORK

Attribute Provisioning. As shown in Section 4, efficient provisioning of federated attributes is crucial for processing federated access management policies in order to resolve policies in a timely manner. The attribute provisioning scheme presented in Section 3.5 supports this goal by reducing the number of communication messages between participating domains to determine if a given AD-Graph depicts a path between a pair of attributes. Each participant organization should decide the number of times it will attempt to retrieve new entries from a DHT ring when constructing a given path. As an example, an organization may set a limit of three explorations of the DHT ring while trying to find a set of input attributes for AD-Rules that fall under the scope of its local domain. Setting a low limit of explorations might prevent participants from discovering a potential path in the AD-Graph, however a large limit may increase attribute provisioning time, thus possibly affecting the overall processing time of a given federated access management policy. In addition, due to the fact DHTs require participants to locally store only a subset of all the entries included in a given ring, our scheme allows participants to store only a subset of AD-Rules entries, thus potentially relieving them from storing information related to the complete AD-Graph. In this way, the process of adding and removing AD-Rules is significantly simplified, thus providing a means for modifying a given AD-Graph to better meet the specific goals devised for collaborations, e.g., adding new AD-Rules to handle user credentials from a new participating domain.

Trust Model. Our current approach assumes all participants in our federation *fully* trust each other for the implementation of both the AD-Rules as well as the model defined in Fig. 4. This strong assumption requires that participants faithfully produce federated attributes by providing verified and accurate AD-Rules and communicating those in a timely manner. However, such an assumption may not always hold in practice. As an example, the incorrect implementation of a given AD-Rule may potentially compromise the overall security of a federated environment. Future work may focus on incorporating a trust model among participants and a risk analysis framework such that incidents can be detected and proper countermeasures can be deployed as a result.

Privacy. Following the *fully-trusted* assumption just described, a basic privacy model may be implemented on top of our approach by allowing for sensitive information contained in locally-defined attributes not to be revealed to other organizational peers when producing federated attributes. For instance, in Fig. 5, sensitive information in attribute *Cred*, e.g., a user's full name, may be replaced by a *pseudonym* in the L attribute produced by the AD-Rule labeled as r_3. An alternative approach may allow for end-users to hide sensitive attributes at request time by incorporating techniques such as the *privacy-preserving attribute-based credentials* (PABC) proposed by Camenisch et al. [3].

Policy Language and Conflict Resolution. Efficient discovery and retrieval of policies (as shown in Section 3) may benefit from the use of a standard policy language, in a similar technique to the one used by the XACML *role-based access control* (RBAC) Profile [22]. Moreover, a comprehensive policy specification framework is critical to detect and resolve conflicts that may arise between federated and local policies, or the intersection of the two, e.g., contradictory rules, following an approach similar to the one proposed by Hu et al. [10].

Integration with NSI. Finally, we plan to work on integrating our approach with the NSI effort presented in Section 2, in such a way that the collaborative efforts devised by participant organizations can be better met by securily leveraging DoE-affiliated high-performance facilities.

7. CONCLUDING REMARKS

In this paper, we have explored the problem of implementing well-defined, consistent, and inter-organizational access management for collaborative resource sharing. In our proposed approach and experiments, we also showed that participants could engage in a federation under a well-defined set of responsibilities, including the use of standardized attribute definitions, attribute provisioning, and distributed policy evaluation. We believe our approach may also be applicable to any other collaborative settings beyond high-performance network environments, e.g. collaborative projects in the health-care domain would certainly benefit for automated approaches that allow for information to be safely shared between independently-run organizations, possibly improving the patient experience and encouraging the development of groundbreaking advancements.

8. ACKNOWLEDGEMENTS

We would like to thank the anonymous reviewers for their valuable comments that helped improve the presentation of this paper. This work was partially supported by the grant from the United States Department of Energy (DE-SC0004308). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding agency.

9. REFERENCES

[1] R. Ananthakrishnan, J. Bryan, K. Chard, I. Foster, T. Howe, M. Lidman, and S. Tuecke. Globus nexus: An identity, profile, and group management platform for science gateways. In *Proceedings of 2013 IEEE International Conference on Cluster Computing (CLUSTER)*, pages 1–3, Sept 2013.

[2] D. Broeder, R. Wartel, B. Jones, P. Kershaw, D. Kelsey, S. Lüders, A. Lyall, T. Nyrönen, and H. J. Weyer. Federated identity management for research collaborations. Technical report, CERN, 2012.

[3] J. Camenisch, A. Lehmann, G. Neven, and A. Rial. Privacy-preserving auditing for attribute-based credentials. In *Proceedings of European Symposium on Research in Computer Security (ESORICS)*, pages 109–127, 2014.

[4] D. W. Chadwick. Federated identity management. In *Foundations of Security Analysis and Design V*, pages 96–120. Springer, 2009.

[5] David W Chadwick and George Inman. Attribute aggregation in federated identity management. *IEEE Computer*, 42(5):33–40, 2009.

[6] M. J. Covington and M. R. Sastry. A contextual attribute-based access control model. In *Proceedings of the 2006 International Conference on the Move to Meaningful Internet Systems (OTM)*, pages 1996–2006. Springer, 2006.

[7] Europe's National Research and Education Networks (NRENs). Geánt Project Home, 2015. `http://www.geant.net/`.

[8] Facebook Inc. Facebook Login, 2015. `https://www.facebook.com/about/login/`.

[9] Open Grid Forum. An Open Global Forum for Advanced Distributed Computing, 2015. `https://www.ogf.org/`.

[10] H. Hu, Gail-J. Ahn, and K. Kulkarni. Detecting and resolving firewall policy anomalies. *IEEE Transactions on Dependable and Secure Computing*, 9(3):318–331, 2012.

[11] Jing J. and Gail-J. Ahn. Role-based access management for ad-hoc collaborative sharing. In *Proceedings of 11th Symposium on Access Control Models and Technologies (SACMAT)*, pages 200–209. ACM, 2006.

[12] Jing Jin and Gail-Joon Ahn. Authorization framework for resource sharing in grid environments. *Grid and Distributed Computing*, 63:148–155, 2009.

[13] X. Jin, R. Krishnan, and R. Sandhu. A unified attribute-based access control model covering dac, mac and rbac. In *Proceedings of the 26th Annual IFIP WG 11.3 conference on Data and Applications Security and Privacy (DBSec)*, pages 41–55. Springer, 2012.

[14] M. Jones and D. Hardt. The oauth 2.0 authorization framework: Bearer token usage. Technical report, RFC 6750, October, 2012.

[15] Kaffille, Sven and Loesing, Karsten. Open Chord, 2015. `http://sourceforge.net/projects/open-chord/`.

[16] N. Klingenstein. Attribute aggregation and federated identity. In *Proceedings of the 2007 International Symposium on Applications and the Internet Workshops (SAINT)*, pages 26–26, Jan 2007.

[17] Ninghui Li, J.C. Mitchell, and W.H. Winsborough. Design of a role-based trust-management framework. In *Proceedings of the 2002 IEEE Symposium on Security and Privacy*, pages 114–130, 2002.

[18] R. L. Morgan, S. Cantor, S. Carmody, W. Hoehn, and K. Klingenstein. Federated Security: The Shibboleth Approach. *EDUCAUSE Quarterly*, 27(4):12–17, 2004.

[19] National Institute of Standards and Technology. Guide to Attribute Based access Control (ABAC) Definition and Considerations, 2013. NIST Special Publication 800-162 Draft.

[20] B.C. Neuman and T. Ts'o. Kerberos: an authentication service for computer networks. *Communications Magazine, IEEE*, 32(9):33–38, Sept 1994.

[21] Nordic Council of Ministers. Nordic Infrastructure for Research & Education (NORDUnet), 2015. `https://www.nordu.net/`.

[22] OASIS. XACML v3.0 Core and Hierarchical Role Based Access Control (RBAC) Profile Version 1.0, 2014. `http://docs.oasis-open.org/xacml/3.0/xacml-3.0-rbac-v1-spec-cd-03-en.html`.

[23] Open Grid Forum. Network Services Interface (NSI), 2015. `https://redmine.ogf.org/projects/nsi-wg`.

[24] F. Paci, R. Ferrini, A. Musci, K. Steuer, and E. Bertino. An interoperable approach to multifactor identity verification. *IEEE Computer*, 42(5):50–57, May 2009.

[25] T. Priebe, W. Dobmeier, and N. Kamprath. Supporting attribute-based access control with ontologies. In *Proceedings of the First International Conference on Availability, Reliability and Security (ARES)*, pages 465–472, Washington, DC, USA, 2006. IEEE.

[26] D. Recordon and D. Reed. Openid 2.0: A platform for user-centric identity management. In *Proceedings of the Second ACM Workshop on Digital Identity Management*, DIM '06, pages 11–16, New York, NY, USA, 2006. ACM.

[27] M. S. Singhalm, S. Chandrasekhar, Ge Tingjian, R. Sandhu, R. Krishnan, Gail-J. Ahn, and E. Bertino. Collaboration in multi-cloud applications: Framework and security issues. *IEEE Computer*, 2013.

[28] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. In *Proceedings of the 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, pages 149–160, New York, NY, USA, 2001. ACM.

[29] US Department of Energy. Energy Sciences Network (ESnet), 2015. `http://www.es.net/`.

[30] X. Zhang, Y. Li, and D. Nalla. An attribute-based access matrix model. In *Proceedings of the 2005 ACM symposium on applied computing (SAC)*, pages 359–363, New York, NY, USA, 2005. ACM.