Supporting Virtual Organizations Using Attribute-Based Encryption in Named Data Networking

Craig A. Lee
Comp. Systems Research Department
The Aerospace Corporation
lee@aero.org

Zhiyi Zhang
UCLA
zhiyi@cs.ucla.edu

Yukai Tu
UCLA
ytu@cs.ucla.edu

Alex Afanasyev
Florida Int'l University
aa@cs.fiu.edu

Lixia Zhang
UCLA
lixia@cs.ucla.edu

***Abstract*—This paper investigates the use of Named Data Networks (NDNs) and Attribute-Based Encryption (ABE) to support federations of computing resources managed using the Virtual Organization (VO) concept. The NDN architecture focuses on fetching structurally named and secured pieces of application data, instead of pushing packets to host IP addresses. The VO concept allows management of federations across different administrative domains and enable secure collaborations. We show how hierarchicallly structured namespaces can be used to manage sets of named resources from different VO sites, and make them available to different VO members, based on their authorization attributes. For this initial investigation, we use a Two-Tier VO model and develop the associated VO data naming schema. We present an example, discuss outstanding issues, and identify future work.**

***Index Terms*—federation management, named data networks, attribute-based encryption**

I. Introduction

The need to securely manage on-demand collaborations across administrative domains is a fundamental requirement that is widespread across many application domains. We can cite supply chain management, international disaster response, and the Internet-of-Things (IoT) as a few examples. In each of these domains, multiple stakeholders need to securely share data, services, and resources to accomplish a common task or goal. While many collaboration needs can be satisfied by centralized approaches, e.g., Dropbox, SharePoint, Google Docs, where files are collected to a logically centralized place that others can access, there are many other situations where this is not possible nor desirable. These are situations where the stakeholders must *federate*, i.e., they selectively make a subset of their services or resources available to a set of external collaborators for a specific purpose or project.

We use the term *Virtual Organization (VO)* to denote any specific federation instance. A VO is a security and collaboration context wherein stakeholders can define, agree upon, and enforce resource discovery and access policies. The term virtual organization was first coined in the grid computing era [1] and has been used in RESTful service architectures [2] At the same time, the abstract VO concept can be used in any distributed computing paradigm. VOs can seem logically centralized, but could be highly distributed in implementation. In fact, there is a large design space for VOs depending on whether centralized or distributed methods are used to validate federated identities or enforce federated resource authorization policies based on a *VO attribute namespace* [3].

This paper investigates the use of VOs in *Named Data Networking (NDN)* architecture [4]. NDN represents a fundamental departure from today's IP communication model, shifting from forwarding packets to destination IP addresses to fetching named and secured pieces of application data (*Data* packets) based on consumer requests (*Interest* packets). Rather than relying on a secure channel or session, NDN directly secures all communicated data, so that all data can be properly authenticated and content decrypted by, and only by, authorized parties, independent of from where the data is retrieved: it could be from the original producer, in-network cache, managed storage, or peer datastore.

This property has profound implications for federations and VOs. Most security models, including VOs, are based on the notion of associating *attributes* with both users and resources, so that resource providers can enforce access policies prior to returning the requested data to the user based on the attributes the user possesses. Current implementation approaches typically use standard protocols, such as TSL, to secure the channel between two endpoints over which data is encrypted using a shared session key. Resource providers can use SAML and similar protocols to get a security assertion about the user using *attribute statements*. XACML protocols can also be used to manage multiple *Policy Decision Points* and *Policy Enforcement Points*.

An alternative approach is to use *Attribute-Based Encryption (ABE)* [5]. Rather than using a session key to encrypt all data between two endpoints during a session, a set of attributes can be used to manage the encryption and decryption process, such that only authorized users ever get to see the data in plaintext. While different ABE approaches will be discussed later, the key observation here is that a VO management system, including its attribute namespace, could

This work is partially supported by US National Science Foundation under award CNS-1719403.

be used to structure an ABE scheme that is integrated into an NDN namespace. That is to say, an arbitrary VO-based federation among stakeholders could be managed by using an NDN namespace to manage both the structure of the dynamically shared resources and the confidentiality of those shared resources by making attribute-based decryption keys available to only the appropriate users. This approach will leverage all of the inherent security properties that NDN has to support a flexible, attribute-based, on-demand collaboration mechanism, i.e., VO-based federations.

This paper reports on our initial efforts to integrate these three technologies. We begin with a short review of related work in the area of distributed security and federation approaches. Section III goes into a little more detail about VOs, NDNs, and ABE. In Section IV, we present our integrated NDN-ABE-VO design. This followed by an evaluation in Section V. We conclude with a summary and future work.

II. Federation Related Work

Federation, in general, and virtual organizations, in specific, entail a number of necessary capabilities that are inherent in distributed environments and must be addressed. These are examined at some length in [3], [6], [7], but can be summarized as

- Federated Identity Management,
- Federation Resource Discovery, and
- Federation Resource Access Control.

There has been on-going work in these and related areas of distributed systems security since the dawn of distributed systems. An early system for managing identity in distributed systems was Kerberos (circa late 1980s) [8]. Kerberos' use of a third-party *Authentication Server* to secure interactions between a *client* and a *principal* was a fundamental development in network security. The Security Assertion Markup Language (SAML) [9] from OASIS formalized this by enabling a Service Provider to securely get an XML-based identity assertion from an Identity Provider on behalf of a user. Shibboleth [10] was subsequently built on SAML by the Internet2 Middleware Initiative, and enabled Service Providers to resolve questions such as *"Where are you from?"*. The eXtensible Access Control Markup Language (XACML) [11], also from OASIS, essentially enables a Service Provider to securely get an access decision from a *Policy Decision Point* rather than having to integrate the access decision logic at every *Policy Enforcement Point*.

More recently, OpenID [12] uses the same three-actor model (users, relying parties, and identity providers), however a user can specify their preferred OpenID IdP. The OAuth protocol enables the delegation of authority whereby a Service Provider can access another Service Provider on behalf of the original client. To properly integrate this delegation of trust with federated identity management, OpenID Connect was developed [13].

While these tools provide critical support capabilities for managing identity and resource access in distributed environments, and are widely used, additional functionality is needed to support general federations. Managing the membership of a federation, and the trust relationship among the members, must be addressed. To this end, *WSFed* was defined that builds on the WS-Trust and WS-Security standards. Here a federation is a set of *security realms* where a service provider in one realm can make authorization decisions based on *claims* asserted by an IdP about a principal in another realm. Given that different federations may have different "business models" and governance requirements, the *Trustmarks* project [14] developed an infrastructure similar to PKI for exchanging signed documents, i.e., trustmarks, concerning these operational issues. They covered topics such as the meaning of an attribute namespace, password lifespans, and even legal agreements required for participation. In the course of the Trustmarks project, 643 trustmarks were defined and used in different federations.

There are also a number of "end-user" distributed systems that are tantamount to federations, or addressing a specific federated application domain. The *grid computing* concept offered institutions a way to make computing resources available to one another, typically for scientific, high-performance computing purposes. A key example is *Globus* and the Globus *Grid Security Infrastructure (GSI)* [15]. GSI uses X.509 PKI certificates issued by trusted Certificate Authorities. The *Interoperable Global Trust Federation* [16] was established to enable mutual trust among participating institutions, such that institutions could trust certificates signed by other institutional CAs. Single sign-on and delegation of trust could be accomplished by using *proxy certs* [17] derived from a user's original certificate. Over the years, Globus has evolved into Globus Online (or simply "Globus") that uses *Globus Auth* [18]. Globus Auth builds on OAuth 2 and OpenID Connect to provide an application-agnostic, authentication and authorization brokerages among client, identity providers, and resource providers. Resource providers must register their services with Globus Auth based on a uniquely identifiable DNS name.

More recently, the OpenStack Keystone project has been building out support for cloud-based federations, e.g., hybrid clouds. As the security service for all other services in the OpenStack suite of open source cloud services, Keystone supports the notion of *federating in* and *federating out* [19], i.e., trusting an external Keystone server as an identity provider, and trusting an external OpenStack service provider, respectively. This is enabled by the federated identity management and attribute aggregation [6].

EduRoam uses a form of federated identity management for the specific purpose of granting WiFi access at academic institutions worldwide [20]. When connecting to a local university WiFi network, a user's identity credentials are routed back to the user's home institution over a tree of *Radius* servers. If the credentials are valid, then the user gets local WiFi access.

InCommon is another federated environment for academic purposes [21]. Operated by Internet2 and based on Shibboleth, InCommon enables institutions to share web-based context by synchronizing federation metadata (typically as a nightly cron job).

These standards, tools, and systems represent the wide breadth of work that has been done on managing identity and resources in distributed systems and federations. At the same time, they are all based on traditional network communication models, e.g., IP, where security is focused on securing the communication channels between IP addresses and ports. The NDN concept was developed to address the shortcomings of this approach by basing all communication on namespaces and encrypting all data when created. Since the need to collaborate is fundamental, we can see some federation-related work in many NDN usage scenarios.

A simple example is that of authorizing your physician to access data from your personal medical devices or monitors, such as FitBits. This can be managed manually, on a case-by-case basis, by delegating the consumption credential management, in this case, to the physician [22]. A federation would provide the mechanisms to manage such delegations automatically across sets of users, institutions, and datasets.

A use case that is closer to a general federation is that of managing access to scientific data using NDN [23]. Here the goal is to make scientific datasets discoverable and accessible by scientists regardless of their geographic location. The owner of a named data catalog *Foo* maintains a fixed prefix, such as "`/Foo/catalog`". The catalog owner issues *publisher keys* whereby institutions can publish data under a prefix such as "`/Foo/catalog/institution_name`". Individual catalogs, however, can federate by running a synchronization protocol, based on *ChronoSync* [24]. While the ability of data publishers to publish to their local catalog is controlled (and potentially federated), data discovery and access is essentially unlimited. When discovering data, data consumers can include SQL query parameters as part of the original Interest request. The catalog will then run an SQL query using those parameters to identify just the desired data names. While this will limit the data names discovered, there is no notion of enforcing any discovery or access policy based on a consumer's attributes. Scientific projects and distributed collaborations, in general, may have members at different institutions around the world, and they may wish to share data with only specific partners for specific reasons. In this case, additional authorization machinery would need to be in place. To do this in a way that is flexible and responsive, rather than static and hard-coded, will require general federation management along the lines of the VO model.

III. The Technology Concepts

The Virtual Organization (VO) concept [1] was developed in the grid computing community over the last fifteen years [25], [26]. The core VO concept is based on a number of fundamental design principles:

- A VO is essentially a security and collaboration context that is not "owned" by any one organization, wherein participants, or *members*, can jointly define, agree upon, and enforce resource discovery and access policies.
- A VO has members that are assigned roles or attributes that essentially define their VO identity.
- Those roles or attributes are used to make specific authorization decisions for using resources within that VO.
- A VO has an administrator that grants or revokes membership and decides a member's role(s) or attribute(s).
- Sites can participate in a VO by contributing resources, i.e., making data and services accessible to VO members from other participating sites.
- Sites retain complete control over their own resources, i.e., access by other VO members can be unilaterally modified or revoked at any time.
- A VO should provide a Single Sign-On capability—once logged into the VO, a member should be able to easily discover and access all resources for which they are authorized, without additional login operations.

Beyond these basic design principles, there are a number of different design variations and implementation approaches within the overall, general federation management design space, as reviewed in [2].

A typical approach for managing all VO status information and the authorization attributes for users from different administrative domains is to use an external *VO Membership Service (VOMS) server* [27]. A similar approach is used by the Open Science Grid [28], where a user authenticates to a VOMS for a specific VO. The VOMS replies with a SAML assertion defining the user's authorizations. The user's client uses this information to build an X.509 proxy certificate, based on the user's primary certificate, and this is used for authorization at the VO-protected service. To date, support for VOs has been integrated into the lowest levels of system software [29] and VOs are used operationally [28], [30].

Additional VOMS can be built in a more modern RESTful service environment. KeyVOMS [2], for example, is a centralized, third-party VO management system based on using a re-purposed OpenStack Keystone v3 service. Keystone is the identity and security service for the OpenStack open-source cloud system. All other OpenStack services, e.g., Nova and Swift, rely on Keystone to maintain and validate user identities, project and group memberships, and granted identity attributes. The Keystone v3 server also supports the concept of a *domain*. Domains "own" different users and projects. RESTful service endpoints in the Keystone service catalog can also be associated with projects. By allowing different VO member sites to register service endpoints in the service catalog and associate them with specific domains and projects, Keystone v3 can be used as VO management system. When a user authenticates for a specific domain and project, they receive a *filtered* service catalog, based on the endpoints associated with that project. When the user makes a service request—regardless of who owns or operates the service—a *VO Policy Enforcement Point* can be used to validate the user's credentials and make an appropriate access decision.

In our previous work [31] we designed *peer-to-peer* KeyVOMS. Here the secure protocols built for Keystone to *federate in* and *federate out* would be extended to enable (a) the propagation of service endpoints among VO members such that replicated *VO shadow projects* could be maintained

at each site, and (b) a VO member's access request could be routed back to their "home institution" for validation. Such a peer-to-peer approach is inherently more scalable than a centralized third-party, and is also the "business model" that many federation may choose to adopt.

Securely implementing VOs in a RESTful service environment, with either a centralized or distributed approach, requires the issuance and validation of cryptographically signed tokens, and also the use of TLS and HMAC message signing to secure the exchange of information routed between service endpoints, i.e., IP addresses and ports. This means that VO management requires the management of discovery and use of service endpoints. In the face of the burgeoning number of online devices and the Internet of Things (IoT), the traditional and established method of routing and securing data based on IP addresses in a way that is orthogonal to application semantics introduces unnecessary complexity and risk.

The *Named Data Networking (NDN)* architecture [4] was developed to address this "semantic gap." NDN are based on the concept of named data at the network level, i.e., a hierarchical namespace [32]. Consumers request data by sending interest packets that include the name or name prefix of the desired data. Once data with the desired hierarchical name is found, a data packet is returned. This Interest-Data exchange is facilitated by an *NDN Forwarder*. An NDN Forwarder checks an Interest for locally cached data in its *Content Store*. If no match is found, the forwarder checks its *Pending Interest Table (PIT)*. If the same interest has already been recorded in the PIT, the forwarder aggregates the interests. Otherwise, the forwarder looks-up the Interest in the Forwarding Table using the longest prefix match, and propagates the Interest according to the forwarding strategy. Hence, the design of NDN applications hinges on the definition of the necessary data namespace.

Given this approach of using a data naming hierarchy to manage communication, how can security be provided and managed? A core tenet of NDN is that security is achieved not by securing channels or sessions between endpoints, but rather securing all data at the point of production [22]. More specifically, the architecture mandates that each Data packet is cryptographically signed, and when confidentiality required, content must be properly encrypted. *Named-based Access Control (NAC)* addresses the issues of (a) how to encrypt the data, and (b) how to securely distribute the decryption keys [33]. NAC is essentially a two-step process based on using separate *production* and *consumption credentials* provided by the Access Manager (e.g., data owner), and a *data content encryption key* that is provided by the Encryptor (e.g., data producer). The production credential is used to sign any data sent from the encryptor to the Decryptor (e.g., data consumer). First, the Data Producer uses the key-encryption key from the consumption credential to encrypt the Data Producer's content encryption key. The Data Consumer uses the key-decryption key from the consumption credential to decrypt the Data Producer's content encryption key. When the Data Producer sends the actual encrypted data, the Data Consumer

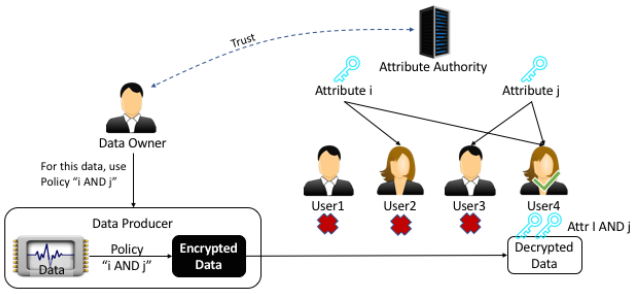

Fig. 1. Ciphertext-Policy Attribute-Based Encryption.

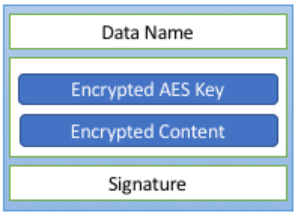

Fig. 2. The Ciphertext format.

can (a) verify that the Data Producer is authorized to produce the data, and (b) decrypt the content data. The distribution of these credentials and keys is managed using the naming hierarchy.

We note in this discussion of NAC that a Data Owner must maintain a key-value list recording the access rights for each Data Consumer. Also, the Data Consumer must maintain a different decryption key for each kind of data that it is authorized to decrypt and use. Clearly this may pose a scalability issue at some point. Another approach to this issue is to use *attributed-based encryption (ABE)* [5]. ABE is a type of public-key encryption where the keys used to encrypt and decrypt a ciphertext are generated from the desired access policy defined over a set of attributes.

ABE can be implemented as *Key-Policy ABE (KP-ABE)* [34], where ciphertexts are simply labeled with a set of attributes. An authorized consumer's private key is associated with an *access tree structure* that defines which types of ciphertexts the key can decrypt. This access structure consists of nodes that are *gates* and leaves that are *attributes* that, in effect, define an *access policy*.

In *Ciphertext-Policy ABE (CP-ABE)* [35], however, this is reversed – a consumer's keys are associated with a set of attributes, while the encryptor can define the access tree structure associated with the encrypted data, essentially defining its access policy. This is illustrated in Figure 1. Here a *Data Owner* instructs a *Data Producer* to encrypt its data according to a specific policy defined over a set of attributes. In this example, the policy is simply *i AND j* which is used to define the access tree structure.

One drawback of ABE is that only a limited length of ciphertext can be handled. To address this issue, we will use the Advanced Encryption Standard (AES, symmetric) and Cipher Block Chaining (CBC) to encrypt and decrypt the plain text, and then use ABE (asymmetric) to encrypt and decrypt

the AES key. This will enable the NDN-VO system to handle any length of input. This arrangement of the resulting data packets is shown in Figure 2 where the AES key is encrypted using the attribute policy, and the content is encrypted by the AES key. (The exact data names will be discussed later.)

On the user's side of Figure 1, a separate *Attribute Authority* grants set of attributes to various users. In this example, only *User4* is granted both i and j. Hence, only *User4's* attributes allow them to "pass through" the ciphertext's access structure and decrypt the data. We note that there is an implicit trust relationship here, since the Data Owner must rely on the Attribute Authority to only issue attributes to genuinely authorized Consumers. However, CP-ABE allows a Data Producer to unilaterally define the access policy for each of its data products.

The challenge, then, is how to integrate ABE and NDN to support VOs. Specifically how to use these mechanisms to manage and enforce data discovery and access based on VO membership and granted attributes, possibly based on *trust schemas* [36]. This is the subject of the next section.

IV. The NDN-ABE-VO Integration

While there is a large design space and many implementation approaches for supporting federations, for this initial investigation we must choose one specific approach. Reviewing the major design requirements and approaches identified in [7], some VOMS implementations can be completely centralized. In this case, the single, centralized VOMS maintains its own *VO Admin*, *VO IdP*, and database of users and services. Users get a VO account assigned out-of-band, where the VO admin assigns any VO roles or authorizations. Some users, i.e., specific VO members, have authorization to register their local services for their VO, within the VOMs. This enables a VO member to discovery services for a specific VO, based on their VO membership, roles, and authorizations.

At the other extreme, the VOMS is completely decentralized. In this case, VO member sites have *VO Site Admins* that must peer to one another, in some fashion, to address all VO management requirements. Here, VO Site Admins can grant/revoke VO membership and roles to their local users. The VO Site Admin must decide with local services to "advertise", i.e., make discoverable, to other VO member sites by communicating with them, perhaps by gossiping, automatic replication, or other peer-to-peer methods. More problematic, though, is (a) how a new VO member site is admitted, and (b) how a new site learns what the VO structure and roles are, and how they are intended to be used, i.e., what their semantics are. The typical way to address (b) is to simply rely on out-of-band knowledge. Addressing (a), however, means that some number of VO Site Admins have the authorization to grant/revoke VO membership to other sites. This has a number of ramifications: (1) Which VO Site Admins get this privilege? (2) How many are there compared to the total number of VO sites? (3) How are the trust relationships managed among the sites that have, and don't have, VO admission privileges? While these are very significant issues, they are unfortunately out of the scope for this paper and must be left as future work.

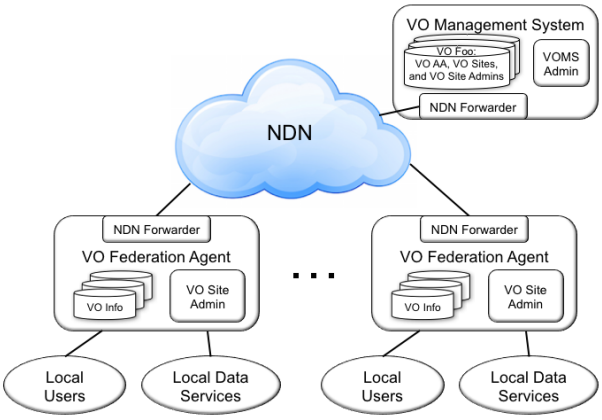

Fig. 3. NDN-VO Two-Tier Model Overview.

A. A Two-Tier VOMS Model

To avoid the added complexity of a completely distributed approach, we will adopt a *Two-Tier* approach, as shown in Figure 3. The VOMS will be logically centralized, but key capabilities will be delegated to the local *VO Site Admins*. The VOMS will maintain its own VOMS Admin, IdP, and database, but only for VO member sites and their VO Site Admins. The VOMS will also maintain the *VO definition*, i.e., the VO project structure and attributes, which will be served by a *VO Attribute Authority (AA)*. (See the next subsection.)

Each site that wishes to participate in a VO will do so through a *VO Federation Agent* that is managed by the VO Site Admin. The VO Site Admin will have authorization to: grant/revoke VO membership and roles to their local users, make local service discoverable by other VO member sites, and to define and enforce local access policy. These federation agents communicate with one another, and the VOMS, through an NDN Forwarder. We note that the VOMS and the agents can all manage multiple, distinct VOs.

This Two-Tier deployment model strikes a balance between implementability, usability, and adoptability. While a VO Site Admin must observe the VO's structure, roles, and semantics (i.e., "play well with others"), the VO Site Admin retains ultimate control over which roles their local users play, and how their local services are used. Having just site membership managed by a logically centralized VOMS avoids the issues of a completely decentralized VOMS. This Two-Tier deployment model is also probably the most natural "business model" that most sites/organizations will be most comfortable with when engaging in a federation.

In previous RESTful implementations, the interactions between users and services was managed using a *VO Site Information Object Model*. This object model was comprised of four object types: users, roles, hierarchical projects and data services. While roles could be granted to users by the *vo_site_admin*, the ability of users to discover and use services was managed using a *project tree*, as illustrated in Figure 4.

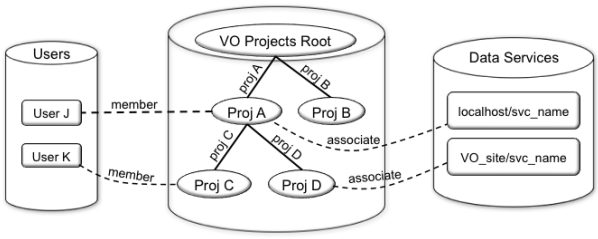

Fig. 4. The Project Tree in the VO Site Information Object Model.

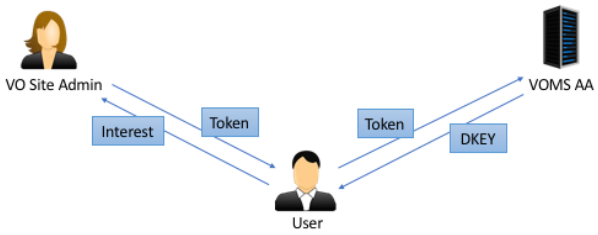

Fig. 5. Issuing attribute decryption keys.

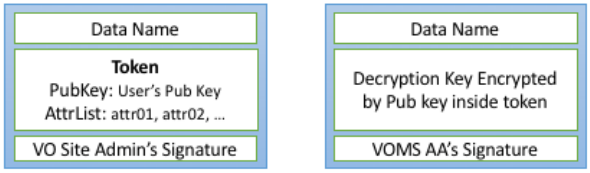

Fig. 6. Message formats.

End-user membership in a VO is granted by granting membership in a specific project within the VO. The data services available to a VO member are determined by which service names are associated with the member's VO project. How a given VO member can use the service would be determined by their roles and the access policy being enforced by the service owner. We note that each VO can also have its own set of roles or attributes that can be granted to VO members. Users and data services at a given site could be associated with more than one VO. Roles and projects, however, are unique to a given VO and could be considered the *VO definition*.

To support these same semantics in an NDN and ABE, users can simply be granted a *project* attribute that will be part of the name space and access policy enforced by Data Owners. While the notion of hierarchical projects can be useful, for now, we will consider just a flat set of projects. Issues of hierarchy and inheritance will be left for future work.

The intent of this model is that data services in a VO could be local or remote, relative to a consumer of those services. Hence, data service names associated with a VO must be replicated and kept consistent among all of the participating sites. By doing so, data services become discoverable by VO members regardless of who the data producer is. The challenge now is to define how the VO name, project path names, and roles can be used to create a consistent, navigable, named data space.

B. Integration with CP-ABE

To utilize CP-ABE in the Two-Tier VOMS model, we must make some fundamental observations. In a symmetric federation between multiple sites, there could be Data Owners, Data Producers and Data Consumers at each site. Without loss of generality, we can say a VO Site Admin is the Data Owner for all data produced by the local data services, i.e., Data Producers. Likewise, Data Consumers are the local users. In the Two-Tier VO model, the VOMS maintains the definition of a VO – including the possible role attributes – while the actual assignment of attributes to users is delegated to the local VO Site Admin. However, since users are issued attributes by an Attribute Authority in the CP-ABE model, we will have the local VO Site Admin *issue a token* to local users, who can then retrieve the actual attribute from the VOMS that functions as the *VO Attribute Authority* for each VO.

Consider Figure 5. After sending an Interest message to the *vo_site_admin*, a token is returned with the format shown in Figure 6 (left). The token includes the User's public key, the User's list of VO attributes, and is signed by the *vo_site_admin*. (The exact data names will be introduced later.) After sending this token to the VOMS, the VOMS AA returns the decryption key with the format show in Figure 6 (right). This is signed by the VOMS AA.

C. Defining the NDN Name Space

Having defined the VO model we wish to use, and how CP-ABE can be integrated into it, the challenge now is how to cast the functional behavior and semantics of this model into an NDN name space, in a manner consistent with [33]. To do this, we will go through a "day in the life" of an NDN-VO. That is to say, we go through the operations and interactions that would be common among a VOMS and VO site members. This will identify all functional semantics that must be supported in the name space. Rather than running all of these hierarchical names in with the text, they are collected in Figure 7 and indexed by the paragraph letter (a-f) in which they are discussed.

a. Site Admin decides to join a VO.
We assume that the Site Admin knows out-of-band the name of the VO it wishes to join, and the name of the VOMS managing that VO. We also assume that the VOMS has an out-of-band method for vetting the identity of the Site Admin requesting VO membership. The VOMS is the "Owner" of the VO membership, i.e., which sites are members. The VOMS also maintains a list of one or more VO Site Admins for each VO member site. Granting VO membership to a site entails responding with the VO's attribute name space, and using a parallel Consumer Credential Namespace (pCCN) [22], the VOMS authorizes the new VO member to read (consume) VO Service Data packets by issuing a *Key Decryption Key (KDK)*.

b. The newly admitted site sends an Interest for all services in VO "<vo_name>".
Other sites respond by sending data packets with service names. The new site can decrypt these data packets using the KDK provided by the VOMS.

a.) "/root_prefix/VOMS/<vo_name>/<vo_site_name>/ADMIT"
b.) "/root_prefix/VO/<vo_name>/VO_SERVICES"
c.) "/root_prefix/VO/<vo_name>/<data_set_name>"
d.) "/root_prefix/VO/<vo_name>/<AA_token>"
e1.) "/root_prefix/VO/<vo_name>/<vo_site_name>/<vo_svc_name>/<vo_req_site>/<vo_user>"
e2.) "/root_prefix/VO/<vo_name>/<wild_card>/<vo_svc_name>/<vo_req_site>/<vo_user>"

Fig. 7. A VO Data Naming Schema.

c. VO Service and Attribute information is kept consistent.
VO service and attribute information is kept consistent by using ChronoSync [24]. This is done by periodically sending out a *sync interest* that includes the *data set name* to be synchronized. Hence, whenever a VO Site Admin adds or deletes a service to a VO, this information is eventually consistent at all other sites.

d. Site authorizes local users to interact with a VO with a specific role.
The VO Site Admin grants a local user VO membership by issuing a token whereby the user can retrieve the appropriate attribute keys from the VOMS AA. This enables the user to decrypt only those data packets encrypted with the corresponding attribute-based policy.

e. VO member issues Interest on specific service names.
Here a VO member issues an interest request on data for which they have the decryption keys (e1). The use of a wild card to request data from all sites providing a given type of data would be useful (e2). We note there could certainly be scalability concerns using wildcards.

f. User Revocation
Local VO Site Admin can change/revoke local user's VO membership and authorization attributes. Since a user's VO traffic must go through the local VO Federation Agent, the VO Site Admin can simply prevent the user from issuing any interests, and not allow the user to use the KDKs to decrypt any data packets.

g. Site Revocation.
At some point in time, a site may wish to leave a VO, or the VOMS Admin may need to revoke the site's VO membership. There are at least two ways this could be addressed. The attribute decryption keys could be issued with a limited period of validity [35]. Hence, unless the keys are periodically re-issued, they will expire and a site's users would not be able to decrypt any data. A second approach is to change all data decryption policies to include an attribute *not_Site_A*, where *Site A* is the site whose membership is being revoked. By issuing this attribute to all VO member sites *except Site A*, all users at Site A can be prevented from decrypting any data.

V. An Example

Figure 8 illustrates how a VO name space can be used to manage sharing between two institutions. The VO Management System and the VOMS Admin are managing two VOs – one named *CS* and another named *MATH*. *CS* has the attributes *Stu, Prof, PostDoc*, and also the *networking* project attribute. *MATH* also has the attributes *Stu, Prof, PostDoc*, and the *n-theory* project attribute. Both VOs have admitted *UCLA* and *MIT* as organizational VO members. Both *UCLA* and*MIT* produce data for the *networking* and *n-theory* VO projects, but define and enforce different access policies.

The *UCLA* VO Site Admin has granted VO attributes to two users. *UserA* is a *Prof* in the *networking* VO project, while *UserB* is a *PostDoc* in both the *networking* and *n-theory* projects. The *MIT* VO Site Admin has granted VO attributes to one user. *UserC* is a *Prof* in both the *networking* and *n-theory* projects.

Both *UCLA* and *MIT* are data producers for the *networking* and *n-theory* projects in the *CS* and *MATH* VOs. For each of these data sources, the data owners define the policies that control access, based on the available VO attributes. Using the operations described in the previous section, each site can appropriately discover the names of other data sources and get attribute tokens allocated to the appropriate VO members. Each VO member can then access the data they are authorized to use, regardless of where the data is actually produced. The data owners can make their data available to a select group of VO users, according to policy. The data owners retain ultimate control over their resources. At any time, data owners can unilaterally change the access policy for their data resources.

To summarize, the Virtual Organization abstraction gives us a security and collaboration mechanism, whereby VO member sites can define and enforce joint access policies for specific data resources they wish to share with other VO members. We have demonstrated that this VO mechanism can be implemented using Named Data Networks and Attribute Based Encryption, thereby realizing the communication and security benefits these technologies enable.

VI. Discussion and Future Work

As an initial effort investigating how to support the virtual organization approach to managing federations using NDNs and ABEs, we have made a number of assumptions and simplifications for expediency. First, we have not discussed or evaluated in detail the local interactions between users and their local VO Site Admin or NDN Forwarder. We have also not addressed any issues of semantic interoperability. This is a fundamental challenge of distributed systems that is outside of the scope of this paper.

A major assumption in this paper was the Two-Tier federation model. This was a good initial choice, based on the arguments given above. Maintaining the consistency of VO information across different sites is central requirement.

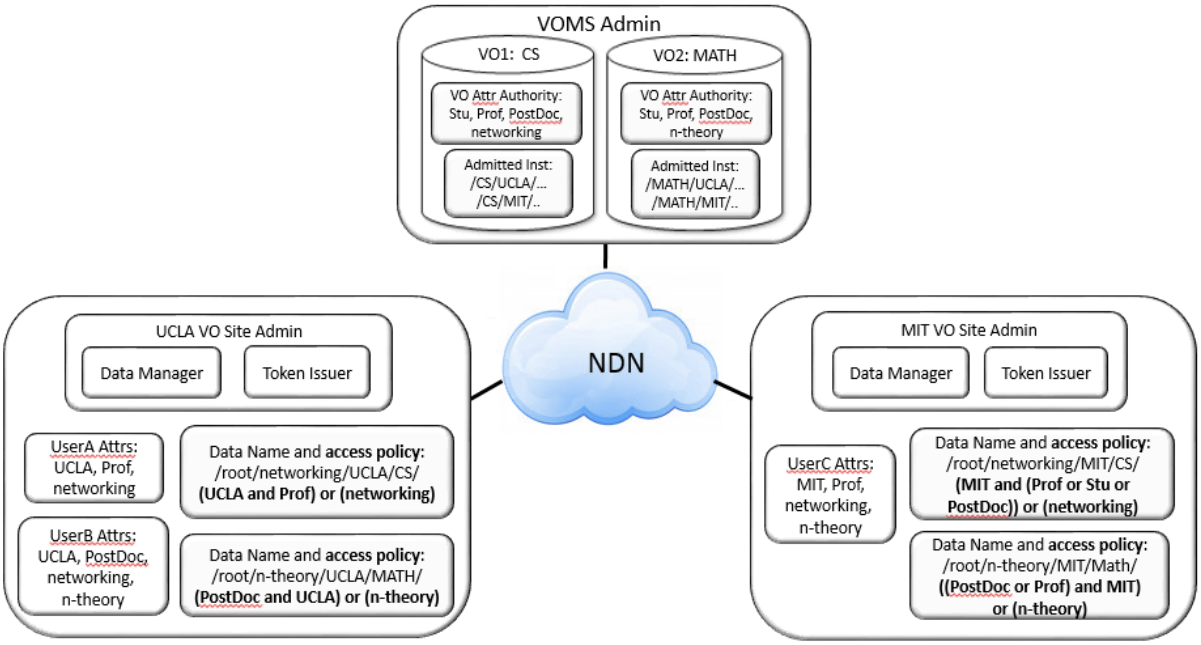

Fig. 8. An NDN-VO Example.

The scalability of maintaining this consistency needs to be evaluated.

The use of a centralized VO Management System would, at some point, also become a bottleneck. A distributed VOMS design would have greater scalability, however, information would have to be replicated across multiple VOMS, with consistency being maintained through a ChronoSymc method. Similarly, the performance impacts for the dissemination of site names and data names should be evaluated.

Perhaps more importantly, a distributed VOMS design raises the issue of having multiple VO Admins. Aside from managing the dissemination of site names, should multiple VO Admins have VO member admission authority? If so, this implies a major trust relationship among VO Admins. If not, this implies a *three-tier* model where only the "root" VO Admin can admit new members, while all the others merely facilitate the scalability of the VO.

Decentralization can also be applied to ABE. Lewko and Waters [37] describe how *Multi-Authority ABE* systems can be defined where any participant can become an authority. After a set of common reference parameters is established, no further coordination is necessary. In their approach, a hash function over a user's global identity and the common reference parameters is used to associate key components together, whereby collusion can be prevented. Such decentralized approaches will be necessary, once VOs are deployed and operated beyond the capacity of a centralized authority.

We have also not properly addressed the issue of *data name discovery policies*. In some VO application domains, Data Owners may wish to limit the discovery of their available data names to a subset of VO Data Consumers, based on some policy. While we have used ABE to ensure that only authorized users can decrypt data packets, we have made an implicit assumption that all data names are available to all VO members, or at least to all local VO Site Admins. In keeping with the Two-Tier model, the VOMS Admin could grant different data discovery authorizations to different VO member sites that would be enforced by the other VO Site Admins.

We also observe that the VO concept essentially constitutes a *trust schema*. Yu, et al. [22], discuss the development of trust schemas in NDNs. A trust schema is defined by a set of *trust rules* that determine which keys must be used for data packets that are produced and consumed within an application. A given set of trust rules are linked with one or more *trust anchors*. This model enables the signing and verification process to be more automated. Applying this approach to VOs may be quite advantageous.

Ultimately, more complete implementations are needed whereby end-to-end testing and evaluation can be done for all of these issues. Complete use cases and demonstration scenarios need to be developed as part of these evaluations. While NDN-supported VOs should benefit from the many useful properties of NDNs, comparisons should be made with other approaches for supporting VOs and federation management. While making direct comparisons may be difficult, comparing the pros and cons of other approaches, such as RESTful service architectures, should be done.

VII. Conclusion

We have made a first effort at understanding how to support virtual organizations using attribute-based encryption in Named Data Networking. As the preceding discussion documented, we made a number of simplifications and assumptions to enable this effort. This gave us at least a point of departure for examining and evaluating the design options. In the process we have identified a number of issues that deserve further examination.

The first and foremost issue is how to motivate the broader community to fully develop the exciting new solution sketched out in this paper. The ability to support on-demand collaborations is a fundamental need across many, many application domains, for many different segments of academia, industry and government. There needs to be an emergent, dominant, best practice for collaboration and federation management that can ultimately be standardized and widely deployed. Ideally such a capability would just "disappear" into the background of commonly used tools that people expect to be available. It is this ultimate goal that should motivate further work in this area.

REFERENCES

[1] I. Foster, C. Kesselman, and S. Tuecke, "The Anatomy of the Grid: Enabling Scalable Virtual Organizations," *Int. J. High Perform. Comput. Appl.*, vol. 15, no. 3, pp. 200–222, 2001, dOI=10.1177/109434200101500302.

[2] C. Lee, N. Desai, and A. Brethorst, "A Keystone-Based Virtual Organization Management System," in *CloudCom*, December 2014.

[3] C. Lee, "A Design Space Review for General Federation Management Using Keystone," in *First IEEE Cloud Federation Management Workshop, UCC 2014*, December 2014.

[4] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, kc claffy, P. Crowley, C. Papadopoulos, L. Wang, and B. Zhang, "Named Data Networking," *ACM Computer Communication Reviews*, Jun. 2014. [Online]. Available: http://dx.doi.org/10.1145/2656877.2656887

[5] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Proceedings of the 24th Annual International Conference on Theory and Applications of Cryptographic Techniques*, ser. EUROCRYPT'05. Berlin, Heidelberg: Springer-Verlag, 2005, pp. 457–473. [Online]. Available: http://dx.doi.org/10.1007/11426639_27

[6] D. Chadwick, K. Siu, C. Lee, Y. Fouillat, and D. Germonville, "Adding Federated Identity Management to OpenStack," *J. of Grid Computing*, December 2013, published on-line: http://rd.springer.com/article/10.1007/s10723-013-9283-2.

[7] C. Lee, "Cloud Federation Management and Beyond: Requirements, Relevant Standards, and Gaps," *IEEE Cloud Computing*, vol. 3, no. 1, pp. 42–49, Jan-Feb 2016.

[8] "Kerberos: The Network Authentication Protocol," http://web.mit.edu/kerberos.

[9] OASIS, "Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML) V2.0," http://docs.oasis-open.org/security/saml/v2.0, March 2005.

[10] "The Shibboleth Consortium," http://shibboleth.net.

[11] E. Rissanen, (ed.), "eXtensible Access Control Markup Language (XACML) Version 3.0," http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.pdf, January 22 2013.

[12] "OpenID Authentication 2.0 Final," http://openid.net/specs/openid-authentication-2_0.html, Dec 5, 2007.

[13] N. Sakimura *et al.*, "OpenID Connect Standard 1.0 - draft 13," http://openid.net/specs/openid-connect-standard-1_0.html, Aug. 16, 2012.

[14] Georgia Technology Research Institute, "GTRI NSTIC Trustmark Pilot," https://trustmark.gtri.gatech.edu.

[15] T. Barton and others, "Identity Federation and Attribute-based Authorization through the Globus Toolkit, Shibboleth, GridShib, and MyProxy," in *5th Annual PKI R&D Workshop*, April 4-6 2006, NIST, Gaithersburg MD. Available from http://toolkit.globus.org/toolkit/presentations/gridshib-pki06-final.pdf.

[16] "The Interoperable Global Trust Federation," http://www.igtf.net.

[17] S. Tuecke *et al.*, "Internet X.509 Public Key Infrastructure (PKI) Proxy Certificate Profile, IETF RFC 3820," http://www.ietf.org/rfc/rfc3820.txt, June 2004.

[18] S. Tuecke, R. Ananthakrishnan, K. Chard, M. Lidman, B. McCollam, S. Rosen, and I. Foster, "Globus auth: A research identity and access management platform," in *2016 IEEE 12th International Conference on e-Science (e-Science)*, Oct 2016, pp. 203–212.

[19] OpenStack Keystone Team, "Configuring Keystone for Federation," https://docs.openstack.org/developer/keystone/federation/configure_federation.html.

[20] L. Florio and K. Wierenga, "Eduroam: providing mobility for roaming users," in *Proceedings of the EUNIS 2005 Conference*, 2005.

[21] InCommon, http://incommon.org.

[22] Y. Yu, A. Afanasyev, and L. Zhang, "Name-Based Access Control," Named Data Networking Consortium, Tech. Rep. Tech. Report NDN-0034, 2015, http://named-data.net/techreports.html.

[23] C. Fan, S. Shannigrahi, S. DiBenedetto, C. Olschanowsky, C. Papadopoulos, and H. Newman, "Managing scientific data with named data networking," in *Proceedings of the Fifth International Workshop on Network-Aware Data Management*, ser. NDM '15. New York, NY, USA: ACM, 2015, pp. 1:1–1:7. [Online]. Available: http://doi.acm.org/10.1145/2832099.2832100

[24] Z. Zhu and A. Afanasyev, "Let's ChronoSync: Decentralized dataset state synchronization in Named Data Networking," *2013 21st IEEE International Conference on Network Protocols (ICNP)*, pp. 1–10, 2013.

[25] B. Nasser *et al.*, "Access control model for interorganizational grid virtual organizations," in *Proceedings of the 2005 OTM Confederated international conference on On the Move to Meaningful Internet Systems (OTM'05)*, 2005, pp. 537–551, dOI=10.1007/11575863_73.

[26] J. Cummings *et al.*, "Beyond Being There: A Blueprint for Advancing the Design, Development, and Evaluation of Virtual Organizations," http://www.ci.uchicago.edu/events/VirtOrg2008/VO_report.pdf, 2008, Final report on NSF workshop *Building Effective Virtual Organizations*.

[27] The Enabling Grids for E-SciencE Team, "EGEE User Guide VOMS Core Services," http://egee.cesnet.cz/en/voce/voms-guide.pdf, 2005.

[28] "Open Science Grid, Virtual Organization Summary," http://myosg.grid.iu.edu/vosummary?all_vos=on&active=on&active_value=1&datasource=summary.

[29] M. Coppola *et al.*, "Virtual Organization Support within a Grid-Wide Operating System," *IEEE Internet Computing*, vol. 12, no. 2, pp. 20–28, March 2008, dOI=10.1109/MIC.2008.47.

[30] "The European Grid Infrastructure," http://operations-portal.egi.eu/vo.

[31] C. Lee, E. Dimpfl, and S. Cathers, "A Keystone-based General Federation Agent," *Fifth IEEE International Workshop on Cloud Computing Interclouds, Multiclouds, Federations, and Interoperability (Intercloud 2016)*, pp. 160–165, 2016.

[32] W. Shang *et al.*, "Named Data Networking of Things (Invited Paper)," *IEEE First International Conference on Internet-of-Things Design and Implementation*, April 2016.

[33] Z. Zhang, Y. Yu, S. Ramani, A. Afanasyev, and L. Zhang, "NAC: Automating Access Control via Named Data," in *MILCOM 2018*, 2018, to appear.

[34] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proceedings of the 13th ACM Conference on Computer and Communications Security*, ser. CCS '06. New York, NY, USA: ACM, 2006, pp. 89–98. [Online]. Available: http://doi.acm.org/10.1145/1180405.1180418

[35] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proceedings of the 2007 IEEE Symposium on Security and Privacy*, ser. SP '07. Washington, DC, USA: IEEE Computer Society, 2007, pp. 321–334. [Online]. Available: http://dx.doi.org/10.1109/SP.2007.11

[36] Y. Yu, A. Afanasyev, D. Clark, V. Jacobson, L. Zhang *et al.*, "Schematizing trust in named data networking," in *Proceedings of the 2nd International Conference on Information-Centric Networking*. ACM, 2015, pp. 177–186.

[37] A. Lewko and B. Waters, *Decentralizing Attribute-Based Encryption*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 568–588. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-20465-4_31