Proceedings of DETC'04
ASME 2004 Design Engineering Technical Conference and
Computers and Information in Engineering Conference
September 28-October 3, 2004, Salt Lake City, Utah USA

DETC2004/CIE-57748

DISTRIBUTED DATA ACCESS CONTROL FOR LEAN PRODUCT INFORMATION SHARING IN COLLABORATIVE DESIGN

Yan Wang
Research Assistant Professor
NSF Center for *e*-Design
University of Pittsburgh
Pittsburgh, PA 15261
ywang@engr.pitt.edu

Pamela N. Ajoku
Ph.D. Candidate
NSF Center for *e*-Design
University of Pittsburgh
Pittsburgh, PA 15261
pne1@pitt.edu

Bart O. Nnaji
William Kepler Whiteford Professor
Director of NSF Center for *e*-Design
University of Pittsburgh
Pittsburgh, PA 15261
nnaji@engr.pitt.edu

ABSTRACT

Efficient, effective, and trustworthy collaboration in design is vital to maintain organizational competence. Conventional access control methods are too coarse in inter-organizational lean and secure data exchange. In this paper, a Scheduled Role-Based Distributed Data Access Control Model is described to support data security management in a distributed environment. The model allows for fine-grained data access control at both the server and client sides, thus securing different levels of design information dissemination for intellectual property protection purposes. Common interface for heterogeneous data is built based on XML.

Keywords: Access control, Intellectual Property Protection, Data security, XML encryption, Interoperability

1. INTRODUCTION

Global markets call for collaborative product development among designers, manufacturers, suppliers, vendors and others. As a result, product design and manufacturing processes are much more distributed now. The business pressures toward outsourcing allow much of the design work of complex products to be done across firms. Ford estimates that there are up to 800 links of supplier relations, and automotive companies are substantively relying on these suppliers to participate in vehicle design [1]. The Defense Advanced Research Projects Agency (DARPA) estimates that the supply chain accounts for more than 50% of weapon system and major subsystem production costs [2]. In such a geographically and temporally distributed environment, efficient, effective, and secure design collaboration should be assured to maintain product quality and organizational competency.

The information infrastructure that supports Internet-based product design should be established to assist cooperation among various design and engineering analysis systems. Collaborative product development includes substantive involvement of activities that contribute to the eventual realization of the product. This includes sharing data and information seamlessly across enterprise boundaries. There are new issues about design information modeling and communication in collaborative design. First, current CAD data formats were designed for standalone systems. All information about components and assemblies has to be available locally in order to be processed. Transferring CAD information among design collaborators requires a large amount of data to be moved around, which is inefficient under the limitation of the communication bandwidth. Second, corporations do not wish to expose complete design data to customers or suppliers because of information confidentiality considerations. A lean and secure communication among collaborators should be established, in which only necessary data is transmitted and exposed to receivers. Different views of data and data flow based on users' need-to-know criteria and their affiliated organization should be provided. It is important to have access controls on the data that has been sent to collaborators during the process of specification exchange. How much information to expose to the collaborators will depend on the trust levels between the collaboration participants. Some communication protocols (e.g. SSL) provide good end-to-end security by securing the communication channel at the packet level and providing in-transit document confidentiality using cryptographic technology. However, such data may be unsecure when it is at either end of the communication link. In product design, this includes proprietary product data. Organization's intellectual property need to be protected while in transit or at rest.

A secure collaborative design information infrastructure should support lean data processing. It should be compliant with industry standards of programming, communication, networking, system management, and interfaces between applications and system services. It should also have good compatibility and interoperability with current design and engineering systems. The activity-level security infrastructure for data exchange should be separated from the organization-level security infrastructure, since participants of inter-organizational data exchange and workflow may change dynamically during the life cycle of an activity. For example, an old organization involved in a project may be replaced, merged, or split. New outsourcing may be needed as a project proceeds. Therefore, a flexible access control model is needed to enable distributed design data access and sharing. Furthermore, a typical access control decision is made based on subjects (users or processes) and objects (files or resources). The conventional access control is too coarse in the situation of inter-organizational lean data exchange and data flow management, which tends to be multi-level and context-dependent. For example, an OEM has multi-tier suppliers, and a supplier participates in different design projects for multiple OEMs. A scalable access control model that supports many-to-many relations with multiple granularity requirements is necessary.

Trust, confidentiality, and integrity issues involved in sharing data are immense. Two important security services needed for product data are *confidentiality* (of product design data in storage or in transit) and *access control* (read, write, delete privileges). Only access control is discussed in this paper. Unlike conventional access control models for files and resources, we focus on the fine-grained access control of different data segments within one file in a distributed design environment. A Scheduled Role-Based Distributed Data Access Control (S-RBDDAC) model for distributed design data is presented. This model combines Role-Based and Cryptographic Access Control to form a new mechanism for flexible data security management in a collaborative environment.

2. BACKGROUND

Major access control methods include: (1) *Discretionary Access Control* (DAC) where access is based solely on the identity of the person trying to access the data resource; (2) *Mandatory Access Control* (MAC) where mechanisms assign security levels to data resources plus security clearances to the user ensuring that users only have access to data for which they have clearance, and (3) *Role-Based Access Control* (RBAC) where permissions are granted based on roles, which are properties of users' ids, sessions, contexts, etc. It should be kept in mind however, that no single access control mechanism would provide the greatest overall benefit to users in all circumstances. Tradeoffs have to be made regarding performance, compatibility, and ease of use for the quality of protection in a particular situation. In collaborative design environments, design data access is controlled by established policies, which interact with the network system architecture, which in turn interacts with the specified application design as shown in Figure 1.

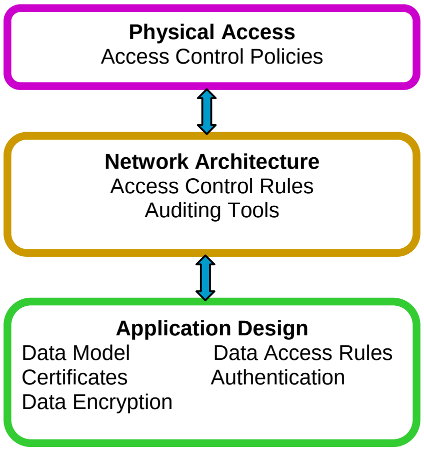


Figure 1. Interactions between policies, system architecture, and application design

Among different methods, RBAC rapidly emerged in the 1990s as a technology for managing and enforcing security in large-scale systems. The basic notion of RBAC is that permissions are associated with roles, which is a job function within the context of an organization with some associated semantics regarding the authority and responsibility conferred on the user assigned to the role [3]. RBAC models are prescribed as a generalized approach to access control and have been shown to be "policy-neutral" in the sense that by using role hierarchies and constraints, a wide range of security policies can be expressed [4].

As an improvement to RBAC, Georgiadis *et al.* [5] proposed a team-based access control using contexts (C-TMAC) approach, which is based on the integration of the RBAC [3] and the TMAC [6] approaches. C-TMAC consists of five sets of entities called users, roles, permissions, teams and contexts, as well as a collection of sessions. The contextual information may include among other things the time of access, the location from which access is requested, the location where the object to be accessed resides, transaction specific values that dictate special access policies etc, allowing TMAC model a much richer set of access policies.

In addition, cryptography is valuable in access control of data independent of system implementations. *Cryptographic Access Control* (CAC) is the mechanism to control data access by key distribution, which is flexible to operate across multiple administrative domains and heterogeneous security policies. Harrington and Jensen [7] proposed a CAC model to ensure secrecy and authenticity of distributed file systems based on asymmetric cryptography and to ensure integrity and availability by using a log-structured file system.

These models and variations are indeed interesting starting points for further investigations of security models for next-generation collaborative applications. In order to adequately embrace the new framework, it is important to determine what the data access control requirements are and relate these requirements to a collaborative online/offline product design environment.

3. DATA ACCESS CONTROL REQUIREMENTS IN COLLABORATIVE ENVIRONMENTS

Access control is an essential part of any collaborative environment containing distributed data. Collaborative environments in which network resources are accessed require the protection of data from unauthorized users. Users or entities in the collaborative process must be identified by properly established access control mechanisms before access is granted or authorization is issued. This is the primary objective of access control in collaborative environments – the preservation and protection of confidential information, the integrity of data, and continued availability of information, systems, and resources. Authentication validates a subject's identity, while authorization determines what resources the subject is allowed to access. Authorization also involves the assignment of privileges such as read, write, modify etc.

In addition to the basic requirements for server-side access control, collaborative environments possess peculiar characteristics such as the dynamic nature of users and roles, wireless transactions etc. Access of critical network resources must be controlled and validated in order to protect proprietary data and corresponding intellectual property.

In a distributed collaboration environment, data sharing occurs within an enterprise as well as from enterprise to enterprise. Traditional access control needs a central management system containing subject-object relations and permissions such as an access control list (ACL) or capabilities. It is difficult to maintain scalability of the centralized monitor in a distributed application environment. Nevertheless, building distributed access control units raise new issues, such as how to update and keep the consistency of labels, how to deal with conflicts between heterogeneous security policies, and how to maintain confidentiality and integrity of access control data. Hence, new access control for collaboration should go beyond just labeling subjects or objects.

Also, access control mechanisms for distributed environment should be stable for the dynamic nature of networked collaboration. Collaboration parties may change during the life-cycle of an activity due to workflow scheduling and balancing, system and software upgrading, and corporation split, merge, and reconfiguration, etc. Access control models should be based on less volatile relations. Another level of abstraction such as roles, which enables indirect references, can provide more stable controls.

Design collaborations require flexible access control for data flow. Different functions and services in collaboration need different portions of data while the communication bandwidth has limits on the frequency of data transfers and updates. Different parties within one enterprise play different roles in collaboration. Data access control for a distributed environment should allow fine-grained information (dataset) dissemination and different views of data based on how organizations and individuals behave in a task and to support the least privilege security principle.

Furthermore, dataset level access control should consider the heterogeneity of data formats. The complexity of different formats, organizations, and structures of data is challenging in the implementation of the fine-grained access control model. Nevertheless, a general model should not make too many assumptions about data formats and storage media. Data could be stored as documents or in databases. A good model should be able to provide an interoperable solution for different data types and formats.

In this paper, we present an S-RBDDAC framework, which uses the RBAC as its core foundation. Utilizing a role-oriented design data access procedure when accessing data within a collaborative environment reduces the cost and complexity of security administration. Product design environment roles include designers, manufacturers, sales personnel, etc. within an enterprise, as well as suppliers, government agencies, retailers, and customers, etc. outside of the enterprise. An example of a product design role graph is shown in Figure 2. Permissions are based on what function a user performs in a particular project within a specified time period.

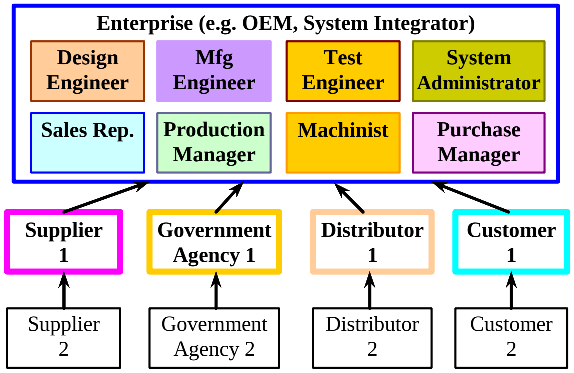


Figure 2. An example of product development role graph

4. DISTRIBUTED DESIGN DATA ACCESS CONTROL

4.1 S-RBDDAC Model

S-RBDDAC is an access control model for collaborative design data, not considering operations and processes. It intends to separate data level controls from system level controls. This model defines elements and their relations, as shown in Figure 3 and listed in Table 1.

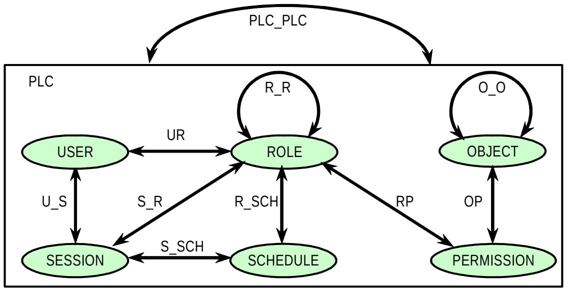


Figure 3. S-RBDDAC model

Table 1. Elements of S-RBDDAC

Elements	Definitions
Policy (*PLC*)	Operating rules that can be referred to as a means of maintaining order, control, and consistency for ease of management.

User (*U*)	A human being; However, the concept of a user can be extended to machines, networks, or intelligent autonomous agents.
Session (*S*)	An activity or work process.
Role (*R*)	A job function within the context of an organization with some associated semantics regarding the authority and responsibility conferred on the user assigned to the role.
Object (*O*)	Any data resource or data segment subject to access control.
Permission (*P*)	An approval to access to one or more protected objects.
Schedule (*SCH*)	A collection of access time intervals, locations, and collaboration states (UP & DOWN) showing the job schedule.

Within the model, the following relations are defined:

- $UR \subseteq U \times R$: a many-to-many mapping user-to-role assignment relation.
- U_S: $U \rightarrow 2^S$, a mapping of a user to the sessions created by the user.
- S_R: $S \rightarrow 2^R$, a mapping of a session to the involved roles.
- $RP \subseteq R \times P$: a many-to-many mapping permission-to-role assignment relation.
- $OP \subseteq O \times P$: a many-to-many mapping object-to-permission assignment relation.
- S_SCH: $S \rightarrow SCH$, a one-to-one mapping of a session to its schedule.
- R_SCH: $R \rightarrow 2^{SCH}$, a mapping of a role to its action schedules.
- $RR \subseteq R \times R$: a partially ordered role hierarchy.
- $OO \subseteq O \times O$: a partially ordered object hierarchy.
- $PLC\_PLC \subseteq PLC \times PLC$: a partially ordered policy hierarchy.

The user-to-role assignment allow the same user play different roles and a single role can have a team of users. The permission-to-role assignment allows a single permission to be applicable for multiple roles and one role can have multiple tasks. The object-to-permission assignment ensures that an object can be accessed with different levels of granularity.

The role hierarchy is mathematically a partial order defining a *seniority* relation between roles, whereby senior roles acquire the permissions of their juniors, and junior roles acquire the user membership of their seniors. The object hierarchy defines a *subset* relation between objects, whereby accessing subset objects at least needs permissions of accessing their supersets. The policy hierarchy is a policy *delegation* relation, in which a delegated policy should be stricter than its parent policies.

This model eliminates unnecessary access from both functional and time considerations. It provides two-dimensional access controls such that access is granted only *when* absolutely necessary and collaboration is established only *where* needed.

User access of the data needs to be constrained by time. For example, a manufacturer within a specified team is not allowed to access relevant data until all *priori* time schedules have been met. This scheduled access control procedure provides a *lean* approach to dataset (*d*) access. Table 2 provides definitions of some more terms in the S-RBDDAC model.

Table 2. Privileges and states

Terms	Definitions
Dataset (*d*)	A set of data containing fine-grained information.
Instance (*I*)	A snapshot of a session.
Value-adding collaboration (υ)	Collaboration in which value is added during the session.
non-value-adding collaboration (*n*υ)	Collaboration in which value is not added during the session
UP State (UP)	The state in which the user is allowed access for expected value added collaboration.
DOWN State (DOWN)	The state in which the user is not allowed access for expected non-value added collaboration.
Negative State (NEG)	The state in which the user is strictly denied for access during the session.
Positive Privilege	The positive privilege contains the UP and DOWN states; This privilege exists for all collaborators who have some type of access to a particular resource
Negative Privilege	The negative privilege contains the Negative state; This privilege exists for all collaborators who do not have any type of access to a particular resource

In any given instance (*I*) of collaboration, only value-adding collaboration (υ) is needed during the process of a product development project. Non-value-adding collaboration (*n*υ), such as unnecessary data browsing and tampering, redundant service request, and unscheduled service initiation, interferes with other collaboration. Figure 4 depicts the relationships and transitions between the states as well as privileges. The project's owner creates sessions and configures users, roles, schedules, permissions, and objects within the sessions. The *schedule* element contains a set of time intervals and locations with associated collaboration status. According to the schedule, roles (and corresponding data access privileges) are classified as being in either the *UP* state (UP: the access is permitted) or *DOWN* state (DOWN: the access is denied) as the case may be for the corresponding session. Negative states are permitted where the users has no access at any time to a particular resource.

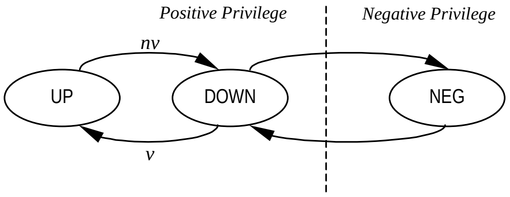


Figure 4. Transitions between states and privileges

A role access privilege is *UP* when the user's collaborative expertise is required for the current dataset. Such collaboration may be required at different stages of the collaborative effort within the product design and realization environment. Scheduled access also minimizes the possible clutter, which may exist in ordinary role-based models by incorporating the "as needed" clause. Depending on the context of the project, "as needed" may be defined as always, sometimes or never. Constraints can be added to the S-RBDDAC model, which are *separation of duty* (SOD) *relations* to enforce conflict of interest policies. These policies may be used to prevent users from exceeding a reasonable level of authority. The constraints can be added to any relations such as user-to-role assignments, role hierarchies, permission-to-role assignments, policy delegation, and SOD relations. For example, users from two competing suppliers cannot be assigned roles that work in one common session; a junior role cannot inherit two senior roles that have mutually exclusive permissions; the owner of a project may delegate its policy to subordinate collaborative partners in order to ensure compliance, maintain security and some level of control regarding the project; and a junior role inherits SOD constraints from its senior roles. SOD relations can be dynamic, which do not persist as schedules change. Figure 5 shows an example of access control policies for two collaborating corporations, each of which creates its own policy to protect its design data based on the defined roles, scheduled time intervals, as well as locations for each session.

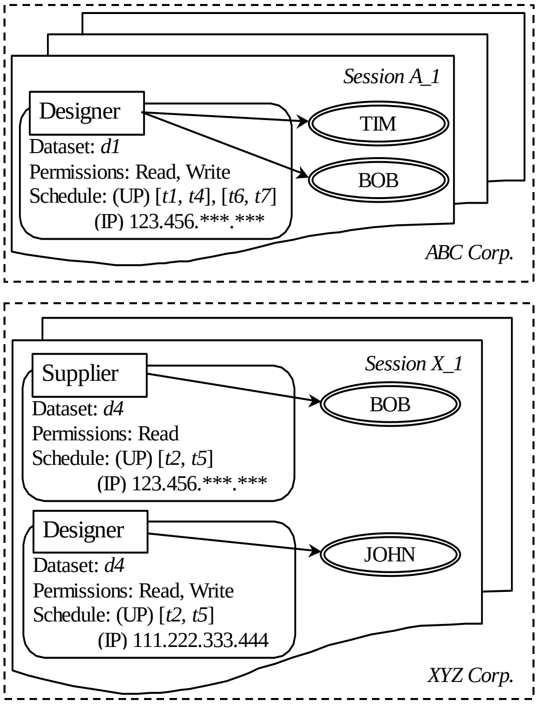
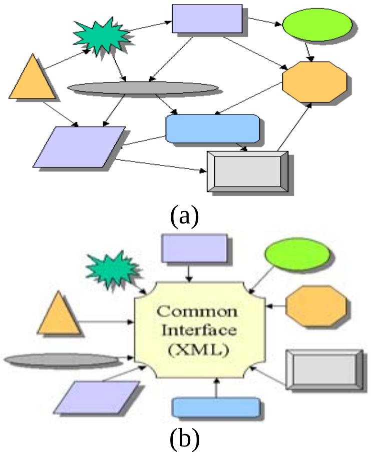


Figure 5. Two examples of access control policies

4.2 Lean Information Sharing through XML

Organizations find it difficult comparing and sharing data with other enterprise data sources due to varying data formats as depicted in Figure 6-a. To enable access control for different product data in various formats, an indirect approach has to be taken in a distributed data-sharing environment. Instead of directly sending original data, a common protocol such as an eXtensible Markup Language (XML) interface for different data formats may be established in advance. XML provides a common syntax for modeling data. It offers a user-defined and extensible format to represent data and information for different application areas. XML also allows for separation of content from format, enabling the processing and presentation of information. XML can handle arbitrary complex data structures and provides interoperability, system-independence, ease of transformation and data parsing. XML identifies structure in a document, and can host text, vector graphics, e-commerce transactions, mathematical equations and other kinds of structured information. Figure 6-b shows the integration of the different sources into one common format. Therefore, XML is a good open medium for information transferring and sharing. Dataset access control meta information can be easily embedded in XML structure.



(a)

(b)

Figure 6. Different data sources/formats and a common data interface

A design data scheme, Universal Linkage-Product Markup Language (UL-PML), is developed to model distributed design data (such as geometry, features, and constraints) in a networked collaborative design environment [8, 9]. Multidisciplinary design information can be captured, distributed, and linked with different levels of granularity and flexibility. UL-PML scheme captures geometric and non-geometric relations among entities in a virtual link style in PML so that references between entities can be made across the boundary of files and physical locations in a distributed design environment. This scheme allows design information to be integrated in a collaborative design environment. Besides static relations among design objects, dynamic relations/constraints are also incorporated.

The network-aware data model intends to improve design *information* interoperability based on general *data* interoperability. At the syntax level, the openness of UL model is guaranteed. Thus, semantics level interoperability is independent from syntax level interoperability.

UL model does not require that one data file contain all the information relevant to the designed product. Incorporating physical distribution and logical integration, it makes partial design information storage and retrieval easy to realize. This provides another level of granularity and increases the flexibility during design information query.

Design information can be stored modularly without compromising the integrity of the whole product. This reduces the requirement for computational time and storage space. Hence, it increases flexibility for scalable designer systems, and encourages reuse of designed components/sections.

The explicit linkage ensures product data's consistency in a distributed environment. Relations of design data elements and constraints are built in the UL model to create a distributed information framework, thus lean information sharing and exchange for collaborative design can be realized over the Internet. The relations among entities are not restricted within one data file. The relations of entities located in different files and domains can be created as well. Relations are linkages among information elements. A linkage model allows physically distributed entities to be linked, thus a logically integrated set of design information can be built. As illustrated in Figure 7, relations of entities (both static and dynamic) in different domains and physical locations can be created. One can easily refer entities in other data files, either at the same machine or other locations over the Internet.

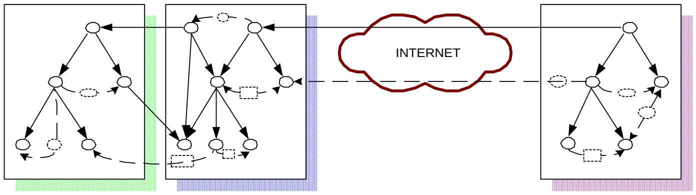
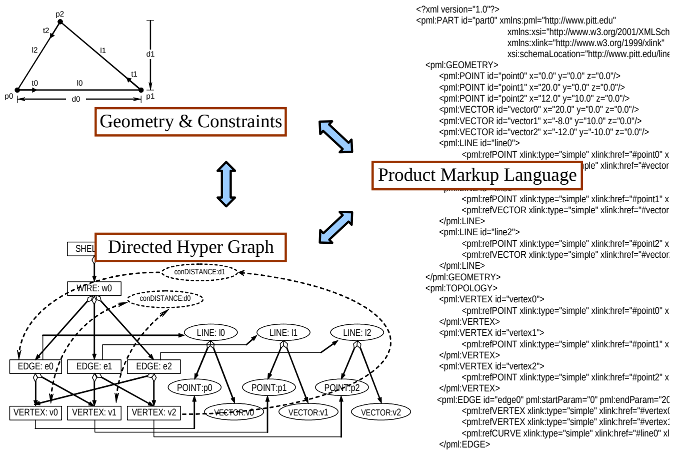


Figure 7. Universal linkage between files

The integrated geometric and non-geometric constraint representation in UL model incorporates more design knowledge into design data. The explicit capturing of multidisciplinary constraints, especially non-geometric constraints, enables a more complete information representation than current standard formats, which can provide a more comprehensive support for design intent representation at different design stages. Graphically, UL model can be represented by DHG. Textually, UL model is stored as PML. Figure 8 shows the model representation of the UL-PML scheme.



```
<?xml version="1.0"?>
<pml:PART id="part0" xmlns:pml="http://www.pitt.edu"
                    xmlns:xsi="http://www.w3.org/2001/XMLSch
                    xmlns:xlink="http://www.w3.org/1999/xlink"
                    xsi:schemaLocation="http://www.pitt.edu/line
  <pml:GEOMETRY>
     <pml:POINT id="point0" x="0.0" y="0.0" z="0.0"/>
     <pml:POINT id="point1" x="20.0" y="0.0" z="0.0"/>
     <pml:POINT id="point2" x="12.0" y="10.0" z="0.0"/>
     <pml:VECTOR id="vector0" x="20.0" y="0.0" z="0.0"/>
     <pml:VECTOR id="vector1" x="-8.0" y="10.0" z="0.0"/>
     <pml:VECTOR id="vector2" x="-12.0" y="-10.0" z="0.0"/>
     <pml:LINE id="line0">
          <pml:refPOINT xlink:type="simple" xlink:href="#point0" x
                                    ple" xlink:href="#vector

          <pml:refPOINT xlink:type="simple" xlink:href="#point1" x
          <pml:refVECTOR xlink:type="simple" xlink:href="#vector
     </pml:LINE>
     <pml:LINE id="line2">
          <pml:refPOINT xlink:type="simple" xlink:href="#point2" x
          <pml:refVECTOR xlink:type="simple" xlink:href="#vector
     </pml:LINE>
  </pml:GEOMETRY>
  <pml:TOPOLOGY>
     <pml:VERTEX id="vertex0">
          <pml:refPOINT xlink:type="simple" xlink:href="#point0" x
     </pml:VERTEX>
     <pml:VERTEX id="vertex1">
          <pml:refPOINT xlink:type="simple" xlink:href="#point1" x
     </pml:VERTEX>
     <pml:VERTEX id="vertex2">
          <pml:refPOINT xlink:type="simple" xlink:href="#point2" x
     </pml:VERTEX>
    <pml:EDGE id="edge0" pml:startParam="0" pml:endParam="20
          <pml:refVERTEX xlink:type="simple" xlink:href="#vertex(
          <pml:refVERTEX xlink:type="simple" xlink:href="#vertex1
          <pml:refCURVE xlink:type="simple" xlink:href="#line0" xl
     </pml:EDGE>
```

Figure 8. UL model representation and mapping

The typical design data has a hierarchical structure, as illustrated in Figure 9. This naturally fits into the XML tree structure. Detailed geometry and topology in a design can also be mapped to PML tree, which strictly follows the syntax of XML. The compliance to industrial computation and communication standard is the premise of computational interoperability at the syntax level.

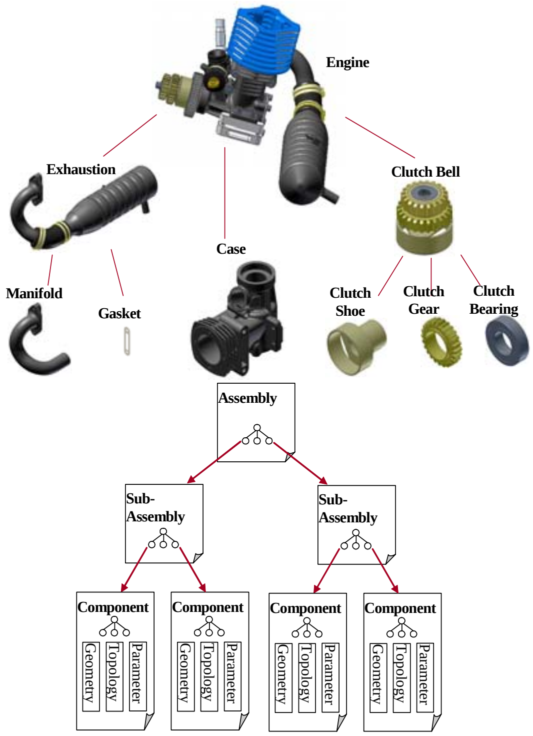


Figure 9. Hierarchical structure of design data

Nevertheless, there are open issues in applying XML to product data representation. First, the mapping between existing CAD data standards and the XML structure needs to be standardized [10]. The XML schema needs to be properly defined according to current needs as well as future extensions. Second, the XML syntax is not as succinct as other CAD data formats. The size of XML file is relatively large, and redundancy exists in tagged text. Third, the flexibility of XML syntax makes standardization difficult. Issues include early-binding vs. late-binding [11], child elements vs. attributes, etc.

4.3 Fine-Grained Modeling and Control for Shared Dataset

In a collaborative design environment, access control at the data level needs to provide confidentiality and flexibility. Cryptography is good for dataset level control. A dataset could be sent to multiple collaborators who have different privileges to access the data subsets. It is also possible that a subset of the dataset that the collaborator received would be sent to a third party with supply chain relationships.

The access privilege is granted to each role through key distribution and policy delegation. The number of keys a collaborator owns is corresponding to the permission he or she has been granted. The policy applied for first-tier collaborators can be delegated to other tier collaborators.

To ensure performance, the XML dataset itself is encrypted with symmetric (secret) keys. Granting access permissions is the process of key distribution. Key distribution is a major issue for distributed secure systems and has been described as a practical problem for parties who wish to set up an encryption system. Key distribution schemes include the Diffe-Hellman Secret-Key Exchange Protocol, which does not provide authenticity and may be suitable for only passive attacks. Nevertheless, there are immense advantages of separating the authentication and encryption functions. These include improved performance, facilitation of analysis and modularity in design and programming. Secret keys may also be distributed based on the asymmetric key cryptosystem (Public key certificate and authority, X.509 certificate architecture).

The key should be established before communication can begin. However, as the number of keys increases, key management becomes highly problematic because n(n-1)/2 keys are needed for n different users. The key distribution scheme should secure against known key attacks. If a particular session key is compromised, it should not affect the usage of other session keys. Based on the PML structure, different key sets may be distributed to users at different security levels as shown in Figure 10. Key *A* serves as an outer key, which unlocks the outer region of the dataset. Key *A* & *B* unlock a deeper subset of the dataset, and so on. Policy delegation for users may be used to enhance key distribution. This policy will determine who uses the keys and when these keys come into effect.

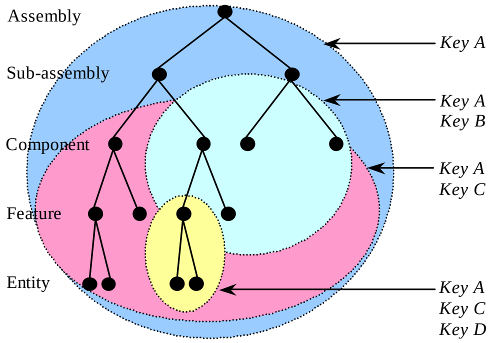


Figure 10. PML encryption with different key sets

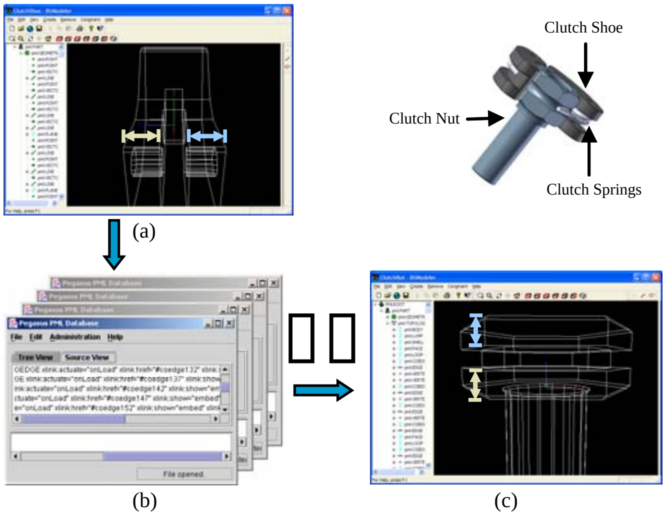
5. DATA SHARING SCENARIOS

The S-RBDDAC model needs two categories of mechanisms for implementation. One is at the system administration level and the other is at the dataset level. Each project owner defines and implements its own access control policy based on its interests. No centralized policy enforcement exists for data access control. At the system level, accesses to memory, disks, database, and other data media need to be controlled. Privileges can be granted through mechanisms of locks, synchronization, and file read/write protections based on access control matrices. It is essential for achieving network security. There is no absolute and perfect security. However, we can achieve computational security if the cost of breaking the cipher is more than the value of the information it is protecting and the time required to break the encryption exceeds the useful lifetime of the protected information. Encryption-decryption key management can have direct controls on disseminated data. Different roles in each session are given different sets of keys to access allowed data sets. The role hierarchy and object hierarchy provide the mechanisms of permission delegation. The project owner defines these high-level access control policies for each project that is overseen.

5.1 Selective Data Exchange

In real world design collaboration, different types of data need to be shared, including geometry, specification, mesh model, simulation, image, as well as documents containing text, graphs, formulas, etc. The design and manufacture of a product is a chained workflow of multiple processes. Open product development systems have to deal with heterogeneous environments and different data formats.

UL-PML scheme provides a data modeling infrastructure for lean product information exchange. Only necessary and relevant data is transmitted based on PML model's fairly loose structure. In the example of Figure 11, two groups who design clutch shoes and clutch nuts need to share some data to make sure that the contacting surfaces of the two parts geometrically match each other. Links between faces in two components can be built. The geometry and topology information of the contacting faces in one can be fetched from the other to maintain the consistency. In this linkage relation, the clutch shoe (Figure 11-a) is at the server site. Once the data is published in a library (Figure 11-b), it is available for reference to meet the surface match requirement. Instead of transferring the whole data file, only relevant faces as well as the corresponding geometry are transferred to the client site through data sharing agents.



```
OEDGE xlink:actuate="onLoad" xlink:href="#coedge132" xlink:
GE xlink:actuate="onLoad" xlink:href="#coedge137" xlink:show
ink:actuate="onLoad" xlink:href="#coedge142" xlink:show="em
ctuate="onLoad" xlink:href="#coedge147" xlink:show="embed"
e="onLoad" xlink:href="#coedge152" xlink:show="embed" xlink
```

Figure 11. Lean information exchange in UL-PML scheme

An example of fine-grained information sharing system with data level access control is shown in Figure 12. The structures and sizes of data involved in the whole product development cycle could vary significantly. This puts a formidable challenge on data access control and management. Thus, access control on a common medium for different domains becomes a feasible solution. The XML-based cross-domain information model becomes a bridge between different data types used in various product development areas. It is important to be able to transform information completely into the desired domain dependent forms.

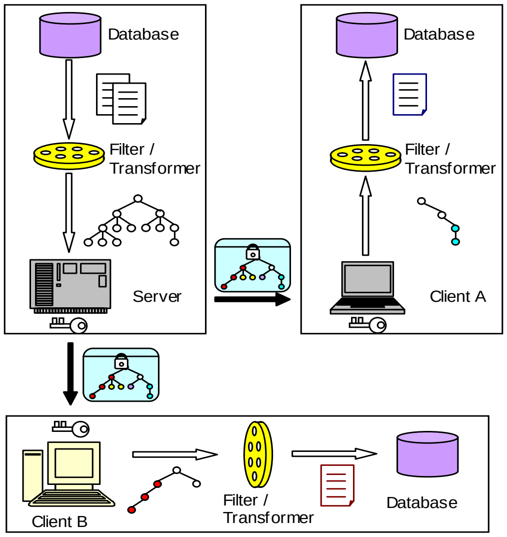
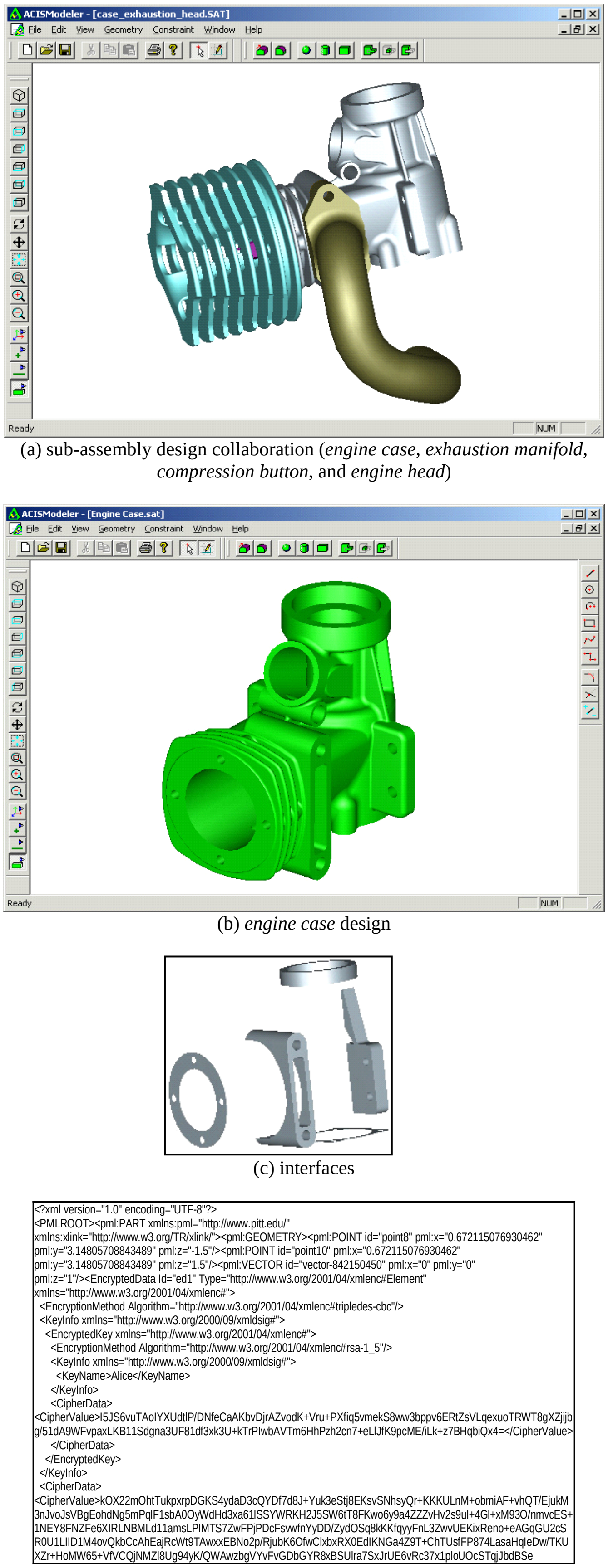


Figure 12. Selective information flow based on XML data model

5.2 Geometric Data Sharing

Based on the XML scheme, relevant information is extracted from original types at the data owner's site and transformed into XML format. This XML dataset then can be transmitted to different client sites securely by encryption mechanisms. Once the data arrives at a client site, it can be transformed into the original or a different format and can be processed locally. The application of the S-RBDDAC Model to PML will provide restrictions on what portion of a PML document a client is allowed to see and statement on when such access is permitted, along with the restriction on the dataset. Such restrictions are achieved through XML data encryption. XML encryption provides end-to-end security for structured data transfer (such as XML data). However, XML encryption can also support certain non-XML data such as binary data. The World Wide Web Consortium (W3C) launched the XML encryption-working group in 2001 [12]. This group defines how to encrypt XML documents. Similarly, PML encryption provides nodal confidentiality for product design data through the elimination of seemingly unnecessary nodes for any given instance or session. Different XML encryption modes exist. The entire PML file may be encrypted. A portion of the PML file may be encrypted, such as an element of the file or only the contents of the element.


ACISModeler - [case_exhaustion_head.SAT]
File Edit View Geometry Constraint Window Help
Ready
NUM
ACISModeler - [Engine Case.sat]
File Edit View Geometry Constraint Window Help
Ready
NUM

(a) sub-assembly design collaboration (*engine case*, *exhaustion manifold*, *compression button*, and *engine head*)

(b) *engine case* design

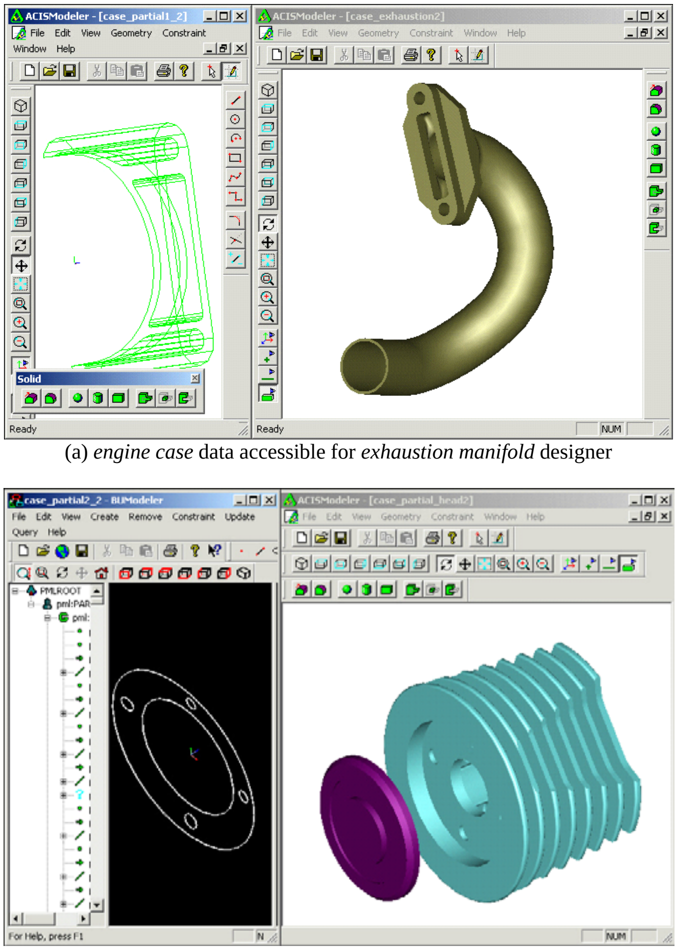
(c) interfaces

```
<?xml version="1.0" encoding="UTF-8"?>
<PMLROOT><pml:PART xmlns:pml="http://www.pitt.edu/"
xmlns:xlink="http://www.w3.org/TR/xlink/"><pml:GEOMETRY><pml:POINT id="point8" pml:x="0.672115076930462"
pml:y="3.14805708843489" pml:z="-1.5"/><pml:POINT id="point10" pml:x="0.672115076930462"
pml:y="3.14805708843489" pml:z="1.5"/><pml:VECTOR id="vector-842150450" pml:x="0" pml:y="0"
pml:z="1"/><EncryptedData Id="ed1" Type="http://www.w3.org/2001/04/xmlenc#Element"
xmlns="http://www.w3.org/2001/04/xmlenc#">
 <EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#tripledes-cbc"/>
 <KeyInfo xmlns="http://www.w3.org/2000/09/xmldsig#">
  <EncryptedKey xmlns="http://www.w3.org/2001/04/xmlenc#">
   <EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-1_5"/>
   <KeyInfo xmlns="http://www.w3.org/2000/09/xmldsig#">
    <KeyName>Alice</KeyName>
   </KeyInfo>
   <CipherData>
<CipherValue>I5JS6vuTAoIYXUdtlP/DNfeCaAKbvDjrAZvodK+Vru+PXfiq5vmekS8ww3bppv6ERtZsVLqexuoTRWT8gXZjijb
g/51dA9WFvpaxLKB11Sdgna3UF81df3xk3U+kTrPIwbAVTm6HhPzh2cn7+eLlJfK9pcME/iLk+z7BHqbiQx4=</CipherValue>
   </CipherData>
  </EncryptedKey>
 </KeyInfo>
 <CipherData>
<CipherValue>kOX22mOhtTukpxrpDGKS4ydaD3cQYDf7d8J+Yuk3eStj8EKsvSNhsyQr+KKKULnM+obmiAF+vhQT/EjukM
3nJvoJsVBgEohdNg5mPqlF1sbA0OyWdHd3xa61lSSYWRKH2J5SW6tT8FKwo6y9a4ZZZvHv2s9ul+4Gl+xM93O/nmvcES+
1NEY8FNZFe6XIRLNBMLd11amsLPIMTS7ZwFPjPDcFswfnYyDD/ZydOSq8kKKfqyyFnL3ZwvUEKixReno+eAGqGU2cS
R0U1LlID1M4ovQkbCcAhEajRcWt9TAwxxEBNo2p/RjubK6OfwClxbxRX0EdIKNGa4Z9T+ChTUsfFP874LasaHqIeDw/TKU
XZr+HoMW65+VfVCQjNMZl8Ug94yK/QWAwzbgVYvFvGDbGYR8xBSUlra7SxJrUE6vRc37x1ploUOcSTqjJbdBSe
```

(d) shared interfaces in encrypted PML

Figure 13. Selective geometry sharing in encrypted PML

Geometric information needs to be shared selectively without compromising intellectual property. As shown in Figure 13, the design of an engine sub-assembly is distributed among three companies. While the engine case is designed at Company A, its geometric interfaces with other components need to be shared with the other companies to ensure proper assembly relation. The interface datasets thus can be selected and represented in PML. The access control to these datasets is defined in Company A's policy. According to the policy, keys are distributed to Company B and C who design the exhaustion manifold and the engine head respectively. They have access to the minimal datasets of the interfaces related to their design and have different views of the shared dataset, as shown in Figure 14.


ACISModeler - [case_partial1_2]
File Edit View Geometry Constraint Window Help
Solid
Ready
ACISModeler - [case_exhaustion2]
File Edit View Geometry Constraint Window Help
Ready
NUM
case_partial2_2 - BUModeler
File Edit View Create Remove Constraint Update Query Help
PMLROOT
pml:PAR
pml:
For Help, press F1
ACISModeler - [case_partial_head2]
File Edit View Geometry Constraint Window Help
NUM

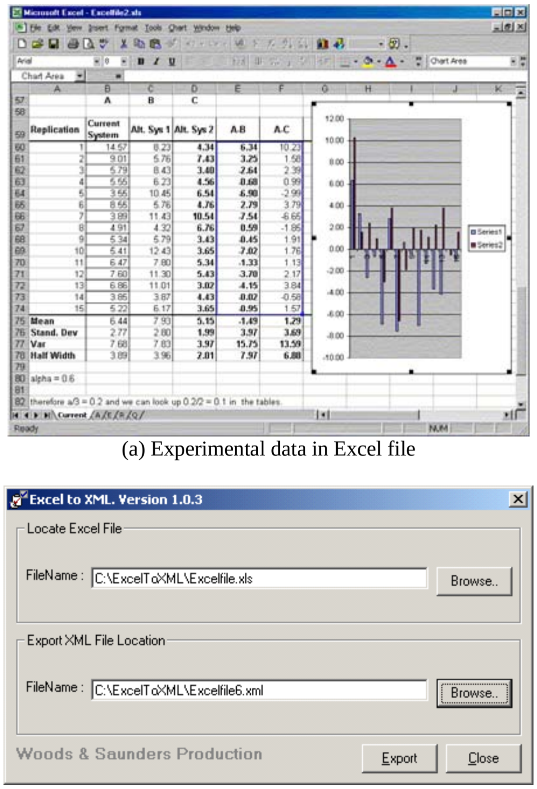
(a) *engine case* data accessible for *exhaustion manifold* designer

(b) *engine case* data accessible for *engine head* designer

Figure 14. Different views of shared data provided for different roles

5.3 Non-geometric design data sharing

Another example is sharing non-geometric data. A set of experimental data for a design (in Figure 15-a) is originally in Microsoft Excel file format. It can be converted into XML format (Figure 15-b), then transmitted in encrypted format (Figure 15-c). Thus, encrypted XML provides a generic and secure data structure for heterogeneous models in collaborative design. Data interoperability and openness enhance the overall information infrastructure of collaboration environment.


Microsoft Excel - Excelfile2.xls
File Edit View Insert Format Tools Chart Window Help
Arial
Chart Area
12.00
10.00
8.00
6.00
4.00
2.00
0.00
-2.00
-4.00
-6.00
-8.00
-10.00
Series1
Series2
Current / A / E / R / Q /
Ready
NUM
Excel to XML. Version 1.0.3
Browse..
Woods & Saunders Production
Export
Close

	A	B	C	D	E	F
57		A	B	C		
58						
59	Replication	Current System	Alt. Sys 1	Alt. Sys 2	A-B	A-C
60	1	14.57	8.23	4.34	6.34	10.23
61	2	9.01	5.76	7.43	3.25	1.58
62	3	5.79	8.43	3.40	2.64	2.39
63	4	5.55	6.23	4.56	-0.68	0.99
64	5	3.55	10.45	6.54	-6.90	-2.99
65	6	8.55	5.76	4.76	2.79	3.79
66	7	3.89	11.43	10.54	-7.54	-6.65
67	8	4.91	4.32	6.76	0.59	-1.85
68	9	5.34	5.79	3.43	-0.45	1.91
69	10	5.41	12.43	3.65	-7.02	1.76
70	11	6.47	7.80	5.34	-1.33	1.13
71	12	7.60	11.30	5.43	-3.70	2.17
72	13	6.86	11.01	3.02	-4.15	3.84
73	14	3.85	3.87	4.43	-0.02	-0.58
74	15	5.22	6.17	3.65	-0.95	1.57
75	Mean	6.44	7.93	5.15	-1.49	1.29
76	Stand. Dev	2.77	2.80	1.99	3.97	3.69
77	Var	7.68	7.83	3.97	15.75	13.59
78	Half Width	3.89	3.96	2.01	7.97	6.88
79						
80	alpha = 0.6					
81						
82	therefore a/3 = 0.2 and we can look up 0.2/2 = 0.1 in the tables					

(a) Experimental data in Excel file

Locate Excel File

FileName : C:\ExcelToXML\Excelfile.xls

Export XML File Location

FileName : C:\ExcelToXML\Excelfile6.xml

(b) Convert Excel to XML

```
<?xml version="1.0" encoding="UTF-8"?>
<xml xmlns:dt="uuid:C2F41010-65B3-11d1-A29F-00AA00C14882" xmlns:rs="urn:schemas-microsoft-com:rowset"
xmlns:s="uuid:BDC6E3F0-6DA3-11d1-A2A3-00AA00C14882" xmlns:z="#RowsetSchema">
<s:Schema id="RowsetSchema"><s:ElementType content="eltOnly" name="row"><s:AttributeType name="F1"
rs:nullable="true" rs:number="1" rs:writeunknown="true"><s:datatype dt:maxLength="8" dt:type="float"
rs:fixedlength="true" rs:precision="15"/></s:AttributeType><s:AttributeType name="F2" rs:nullable="true"
rs:number="2" rs:writeunknown="true"><s:datatype dt:maxLength="8" dt:type="float" rs:fixedlength="true"
rs:precision="15"/></s:AttributeType><s:AttributeType name="F3" rs:nullable="true" rs:number="3"
rs:writeunknown="true"><s:datatype dt:maxLength="8" dt:type="float" rs:fixedlength="true"
rs:precision="15"/></s:AttributeType><s:AttributeType name="F4" rs:nullable="true" rs:number="4"
rs:writeunknown="true"><s:datatype dt:maxLength="8" dt:type="float" rs:fixedlength="true"
rs:precision="15"/></s:AttributeType><s:AttributeType name="F5" rs:nullable="true" rs:number="5"
rs:writeunknown="true"><s:datatype dt:maxLength="8" dt:type="float" rs:fixedlength="true"
rs:precision="15"/></s:AttributeType><s:AttributeType name="F6" rs:nullable="true" rs:number="6"
rs:writeunknown="true"><s:datatype dt:maxLength="8" dt:type="float" rs:fixedlength="true"
rs:precision="15"/></s:AttributeType><s:extends type="rs:rowbase"/></s:ElementType></s:Schema>
<rs:data>
<EncryptedData Id="ed1" Type="http://www.w3.org/2001/04/xmlenc#Element"
xmlns="http://www.w3.org/2001/04/xmlenc#">
  <EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#tripledes-cbc"/>
  <KeyInfo xmlns="http://www.w3.org/2000/09/xmldsig#">
    <EncryptedKey xmlns="http://www.w3.org/2001/04/xmlenc#">
      <EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-1_5"/>
      <KeyInfo xmlns="http://www.w3.org/2000/09/xmldsig#">
        <KeyName>Alice</KeyName>
      </KeyInfo>
      <CipherData>
<CipherValue>Y0BSXA8zLyC/MEncA+G8xsRLuqJWEKF5vYBi+UgxaqKErQB7+qyhf/FDHT9aUKZUkOUYJWdS3XAf2
g+4Hh/NGpzjTT0hSJrj428iW/4YK6vv3d3VhloUel7we2XSXLaKUImsZpgVv38q88xsPJ0NYvgQgeN6MrWAI+romkVM2
BM=</CipherValue>
      </CipherData>
    </EncryptedKey>
  </KeyInfo>
  <CipherData>
<CipherValue>X22JkGbqcchKKzv+leAaBMD24PzlHEyagAwJtS3zkDGEpRpfJCS3rvmJdnnKhe/GeSXckSdXsogCeXN
C8Be49GMX9g1nAvDRxHUeC1Rr0b7EN8AJ/FD0BPTFLCxg+k0COXyUd4CD1t2enuDhZizVRCfBw5tutFQPjUlD0yQR
```

(c) Encrypted XML for data exchange

Figure 15. Secure non-geometric data exchange by XML encryption

6. SUMMARY

This paper presents an S-RBDDAC model for data security management in a collaborative design environment. This model combines RBAC and Cryptography methods for fine-grained data access control such that lean and secure data exchange and sharing are supported. Based on the functional roles and schedules, relatively stable and easy access control for federated environments can be created. The S-RBDDAC model is based on the requirement of intellectual property protection while sharing data in collaborative environments. The uniqueness of this model includes the consideration of time, scheduling, and value-adding activity with roles, the policy delegation relation in a distributed context, and fine-grained access control at dataset level. Heterogeneous data is shared through XML common interface, which provides a neutral solution to enhance data interoperability. These factors increase the flexibility of the model and promote an open and interoperable information infrastructure.

REFERENCES

[1] The white paper of U.S. National Science Foundation workshop on e-product design and realization for mechanically engineered products, October 19-20, 2000, Pittsburgh, PA, http://www.e-designcenter.info

[2] H.V.D. Parunak, "Distributed Collaborative Design (DisCollab): An ATP Opportunity" http://www.mel.nist.gov/msid/groups/edt/ATP/white-paper (Whitepaper of NIST-ATP Workshop: Tools and Technologies for Distributed and Collaborative Design), August 1997

[3] D.F. Ferraiolo, R. Sandhu, S. Gavrila, D.R. Kuhn, and R. Chandramouli, "Proposed NIST Standard for Role-Based Access Control", *ACM Transactions on Information and System Security*, Vol.4, No.3, pp.224-274, August 2001.

[4] S. Osborn, R. Sandhu, and Q. Munawer, "Configuring Role-Based Access Control to enforce Mandatory and Discretionary Access Control Policies", *ACM Transactions on Information and System Security*, Vol.3, No.2, pp.85-106, May 2000.

[5] C.K. Georgiadis, I. Marvridis, G. Pangalos, and R.K. Thomas, "Flexible Team-Based Access Control Using Contexts", *Proceedings of the Sixth ACM Symposium on Access Control Models and Technologies, May 3-4, 2001, Chantilly, Virginia, USA*, pp.21-27

[6] R. K. Thomas, "Team-Based Access Control (TMAC): A Primitive for Applying Role-Based Access Controls in Collaborative Environments", *Proceedings of the Second ACM workshop on Role-based Access Control, Fairfax, VA, USA, November 6-7, 1997*, pp.13-19

[7] A. Harrington and C. Jensen, "Cryptographic Access Control in a Distributed File System", *Proceedings of the Eighth ACM Symposium on Access Control Models and Technologies, June 2-3, 2003, Como, Italy*, pp.158-165

[8] Y. Wang and B.O. Nnaji, "Functionality-Based Modular Design for Mechanical Product Customization Over the Internet", *Journal of Design and Manufacturing Automation*, Vol. 1, No.1-2, pp.107-121, October 2001.

[9] Y. Wang and B.O. Nnaji "UL-PML: Constraint-Enabled Distributed Product Data Model", *International Journal of Production Research*, in press, 2004

[10] J. Lubell and S. Frechett, "XML Representation of STEP Schemas and Data", *ASME Journal of Computer and Information Science in Engineering*, Vol.2, No.1 (2002), pp.69-71

[11] J. Lubell, "From Model to Markup", *Proceedings of XML Conference, Baltimore, MD, December 8-13, 2002*, http://www.idealliance.org/papers/xml02/dx_xml02/index/title/f8d35958a0b772897676d5c4dc.html

[12] W3C XML Encryption, http://www.w3.org/TR/xmlenc-core