Information Flow Control for Cooperation Support in Virtual Enterprises

Peter Bertok[1], Abdelkamel Tari[2], Saadia Kedjar[3]

[1]School of Computer Science and IT, RMIT University, Melbourne, Australia
[2]Laboratory of Applied Mathematics (LMA), University of Bejaia, Algeria
[3]Departement of Computing, University of Bejaia, Algeria

Abstract. Cooperation requires sharing data, but enforcing access restrictions across enterprises is a challenge. Different sites have different policies and use a variety of access control methods that are tailored to the individual enterprises' needs. Mapping one set of rules into another may require complex computations, possibly with a separate method for each pair of sites. This paper proposes an information flow control model for enforcing access restrictions across a virtual enterprise. Labels are assigned to data structures to ensure uniform treatment across the enterprise, and dynamic label checking provides flexibility during operation. A set of rules are presented to facilitate data manipulation so that they do not lead to information leak. The proposed solution particularly suits web-based environments and web services operations.

Keywords: virtual enterprise, access control, information flow control

1 Introduction

Preserving data security in a virtual enterprise is vital, as trust between the partners is limited. The loose coupling between sites facilitates cooperation, but also means that common mechanisms are kept at a minimum level and security policies of one site may be quite different from those at another site. Data needs to be exchanged and accessed at different sites for proper functioning of the virtual enterprise, but at the same time strict rules need to be enforced to prevent information leak to unauthorised entities. This paper presents a framework to share information between partners of a virtual enterprise while enforcing data security and privacy.

To facilitate integration with individual systems at virtual enterprise (VE) sites, we propose the use of role-based access control (RBAC) [9], as it is a proven method for VEs [8, 10]. By assigning access constraints to roles rather than to individuals, we can bridge the gap between different representations of actors by participants of the VE at different sites. Subjects are assigned roles before accessing objects, and the role of the subject determines the set of privileges a subject has on an object. Each object has a security label that describes its owners' access control policies with relation to reading and writing, and access restrictions are enforced by information flow control. Our model was designed primarily for web-based environments and web services.

The structure of the paper is as follows. First we provide a brief description of information flow control, then we describe the basic components of our model. It is followed by the details of the model and a brief overview of the implementation, before the paper is concluded.

2 Background

2.1 Information flow control (IFC)

Access control mechanisms are designed to control immediate access to objects, without considering implicit information flow. For example, if user A has no read access to an object but user B has, then user B can forward the content of the object to user A, and the result is information declassification. Similarly, if A is not allowed to write the object but user B is, A can pass information to B for writing.

Information flow control (IFC) addresses such problems, as unauthorised operations can immediately be detected and declassification can be avoided. An information flow is secure if it does not lead to unauthorised disclosure or destruction of information.

Information flow control has a fairly long history. A very general treatment of information flow security models was given by McLean [6] and Millen [7]; while Wittbold and Johnson [12] applied standard information theory concepts to concrete but simple examples. Gray [5] attempted to bridge the gap between general but abstract and concrete, specific but limited solutions, and proposed a general state-machine model. Effective implementations, however, were not presented.

More recently, Tari et al [11] proposed IFC for web services. The model proposed here extends that model and has a wider scope. We also improve applicability by introducing additional operations, such as write and controlled declassification.

3 The Proposed Method

3.1 Outline

Our IFC model is based on dynamic label checking [11]. Data structures have security labels attached to them to describe data sensitivity, and data can be accessed only via special modules that use the labels to enforce access restrictions. The modules can be part of the VE infrastructure.

We propose a set of operations on the security labels, and define accessibility of results when data items are manipulated. The operations on the labels are designed to maintain security of the data items, and can be associated with any operations on the actual data itself. Our focus is on read and write operations, and we also consider

controlled declassification under well-defined circumstances to help completion of operation sequences (transactions).

3.2 IFC Components

The model considers passive entities called objects that can be manipulated via different operations, and active entities called subjects who can perform those operations. Data items are objects, and subjects can be owners, readers and writers of those objects. Each object can have a number of owners, readers and writers. An owner of an object may trust some of the other owners of the same object. Each owner can nominate readers and writers of the object, as well as potential declassification recipients. Accordingly, we define the following *sets of subjects* for each object q.

Owner set Oq: all subjects that own this object. For example medical data owned by the patient and by doctors.

Effective Readers set ERq: the intersection of all owners' Reader sets, i.e. subjects who have been granted read access to the object by all owners $ERq = \bigcap_{oi \in Oq} Rq,oi$

Joint Reader set JRq: union of the Effective Reader set and all owners of the object, i.e. subjects who can read the object
$JRq = Oq \cup ERq$

Effective Writer set EWq: the intersection of all owners' Writer sets, i.e. subjects who have been granted write access to the object by all owners $EWq = \bigcap_{oi \in Oq} Wq,oi$

Joint Writer set: union of the Effective Writer set and all owners of the object, i.e. subjects who can write the object
$JWq = Oq \cup EWq$

Trusted Owner TOq: an owner of the object who is trusted by at least one other owner of the object

Effective Owner EOq: who is trusted by all other owners of the object.

Effective declassification set for reading $EDRq$: subjects for whom reading declassification can occur $EDRq = \bigcap_{oi \in EOq} DRq,oi$

Effective declassification set for writing $EDWq$: subjects for whom writing declassification can occur $EDWq = \bigcap_{oi \in EOq} DWq,oi$

We assign *security labels* to objects. These labels contain access control policies of all owners of the object, and they are used to control how the information contained in this object can be disseminated and modified. A label is a set of components where each component represents an owner's policy on the object.

$$L(q) = \{K_1, K_2, \ldots, K_i, \ldots, K_c\}$$

A component has six elements:

$$K = \{K_o, TO_{q,Ko}, R_{q,Ko}, W_{q,Ko}, DR_{q,Ko}, DW_{q,Ko}\}$$

where:

- K_o is an owner of object q, i.e. $K_o \in O_q$
- $TO_{q,Ko}$ is the trusted owner set defined by K_o
- $R_{q,Ko}$ is the reader set defined by K_o
- $W_{q,Ko}$ is the writer set defined by K_o
- $DR_{q,Ko}$ is declassification policy set for reading, such as
  $DR_{q,\,Ko} = \{\delta_1, \delta_2, \ldots, \delta_p\}$.
  Each component δ contains two parts:
  $\delta_{init} \in R_{q,o}$ the subject to whom the declassification will appear.
  $\delta_{inter} \subseteq R - R_{q,o}$ the subset of intermediate subjects (roles) that carry out the declassification, such that $\{\delta_{init}\} \cap \delta_{inter} = \varnothing$
- $DW_{q,Ko}$ is declassification policies set for writing, such as
  $DW_{q,\,o} = \{\delta_1, \delta_2, \ldots, \delta_p\}$.
  Each component δ contains two parts :
  $\delta_{init} \in W_{q,o}$ the role to whom the declassification will appear.
  $\delta_{inter} \subseteq R - W_{q,o}$ the subset of intermediate roles that carry out the declassification, such that $\{\delta_{init}\} \cap \delta_{inter} = \varnothing$

3.3 IFC Rules

A set of rules manage access to objects in the system and to their labels, and maintain data confidentiality and integrity. Permissions-to-roles assignments are represented by the function MapPR, its arguments being the permissions and the object.

Table 1. *Data access permissions.* Data can be accessed only by subjects who are authorized to perform the requested operation.

Rule	Interpretation
Rule 1 MapPR (read, q) = $\{r / r \in JR_q\}$	A role can read the object q if and only if the role belongs to the joint reader set of q
Rule 2 MapPR (write, q) = $\{r / r \in JW_q\}$	A role can read object q if and only if the role belongs to the joint writer set of q
Rule 3 MapPR (delete, q) = $\{r / r \in O_q \wedge \lvert O_q \rvert = 1\}$	A role can delete object q if and only if it is the single owner of q

Table 2. *Label access permissions.* Labels can be manipulated only by authorized subjects, who maintain the integrity of the labels and thereby the integrity of the whole system.

Rules	Interpretation
Rule 4 MapPR ({read, write, add, delete}, K. $R_{q,Ko}$) = $\{K_o\}$	Only the owner of a component in a label can read, write, add and delete an element of the reader set of that component.

Rule 5 MapPR ({read, write, add, delete}, K. $W_{q,Ko}$) = {K_o}	Only the owner of a component in a label can read, write, add and delete an element of the writer set of that component.
Rule 6 MapPR ({read, write, add, delete }, K. $TO_{q,Ko}$) = {K_o}	Only the owner of a component in a label can read, write, add and delete an element of the trusted owner set of that component.
Rule 7 MapPR ({read, write, add, delete }, K. $DR_{q,Ko}$) = {K_o}	Only the owner of a component in a label can read, write, add and delete an element of the read declassification set of that component.
Rule 8 MapPR ({read, write, add, delete }, K. $DW_{q,Ko}$) = {K_o}	Only the owner of a component in a label can read, write, add and delete an element of the write declassification set of that component.
Rule 9 MapPR (add, K.O_q) = {r / r ∈ EO_q} MapPR (delete, K. O_q.o) = {r / r ∈ EO_q ∧ o ∉ EO_q}	Only the effective owners of an object can extend the set of the owners of the object and can delete an owner. The owner to be deleted must not be an effective owner.

Table 3. *Declassification rules.* These rules allow the normal execution of transactions even if an intermediate subject is not authorized to read or write an object; they can be used if and only if the initiator of this transaction is an effective reader or writer respectively.

Rules	Interpretation
Rule 10 (r_1, ε) ∈ EDR_q ∧ r_1 ∈ ER_q ⇒ add (ε) to ER_q	If (r_1, ε) is a component of the EDR_q set and r_1 is an effective reader, then we can add the reader set ε to the ER_q set
Rule 11 (r_1, ε) ∈ EDW_q ∧ r_1 ∈ EW_q ⇒ add (ε) to EW_q	If (r_1, ε) is a component of the EDW_q set and r_1 is an effective writer then we can add the writer set ε to the EW_q set

Table 4. *The rule of the safe information flow*.

Rule 12	An information flow is safe, if it does not lead to information disclosure, and the confidentiality of information contained in the source object is guaranteed. In other words, the number of authorized readers will not increase as a result of the information flow. non-authorized writing an object, and the integrity of the recipient object will be maintained. This means that the number of authorized writers will not increase as a result of the information flow.

Operations on Labels. When performing operations on one or more objects, a result object with a new, derived label is produced. The operations on the labels can be associated with any operation on the objects themselves. In this section, we describe operations for deriving a new label. We define *join* operations, and introduce the operator "_" to join labels.

For all label joins, let q_1, q_2, $q_3 \in Q$, $L(q_1)$, $L(q_2)$ and $L(q_3)$ be the labels of q_1, q_2 and q_3 respectively and $L(x)$ a new label. Let us suppose that there is an operation on q_1 and q_2 that produces q_3, i.e. $q_3 = q_1 \, Op \, q_2$.

Assigning join ($\_^a$). This operation is tailored to object assignment. Instead of inheriting all characteristics of one object, the result is a combination of both objects.

In the assignment operation the label of the destination object first takes over that of the destination object, and then the trusted owner, reader and read declassification sets are further enlarged, by adding effective owners, readers and read declassification sets of the source object. The write and declassification write sets are reduced in a similar fashion.

If $L(x) = L(q_1) \_^a L(q_2)$ then:

$O_x = O_{q1} \cup (O_{q2} \cap JR_{q1})$

$\forall o_i \in O_x \backslash O_{q1}$ $(o_i \in O_x$ and $o_i \notin O_{q1})$ and $\forall o_j \in O_{q1}$

$R_{x,oi} = ER_{q1} \cup R_{q2,oi}$; $R_{x,oj} = R_{q1,oj}$

$W_{x,oi} = W_{q2,oi}$; $W_{x,oj} = W_{q1,oj} \cap JW_{q2}$

$TO_{x,oi} = EO_{q1} \cup TO_{q2,oi}$; $TO_{x,oj} = TO_{q1,oj}$

$DR_{x,oi} = EDR_{q1} \cup DR_{q2,oi}$; $DR_{x,oj} = DR_{q1,oj}$

$DW_{x,oi} = DW_{q2,oi}$; $DW_{x,oj} = DW_{q1,oj} \cap EDW_{q2}$

$L(q_3) \leftarrow L(x)$

Restrictive join ($\_^r$). This operation is appropriate when dealing with sensitive data. The resulting label will first take the properties of the source label, then the owner, reader and writer sets are reduced by removing those that do not have a similar role in the destination label. If no common owners exist, permissions are preserved through a new owner that represents the system itself.

If $L(x) = L(q1) \_^r L(q2)$ then:

$O_x = O_{q1} \cap O_{q2}$

if $O_x \neq \varnothing$ then $\forall o_i \in O_x$

$R_{x,oi} = R_{q1,oi} \cap R_{q2,oi}$

$W_{x,oi} = W_{q1,oi} \cap W_{q2,oi}$

$TO_{x,oi} = TO_{q1,oi} \cap TO_{q2,oi}$

$DR_{x,oi} = DR_{q1,oi} \cap DR_{q2,oi}$

$DW_{x,oi} = DW_{q1,oi} \cap DW_{q2,oi}$

else $O_x \leftarrow System$

$R_x = JR_{q1} \cap JR_{q2}$

$W_x = JW_{q1} \cap JW_{q2}$

$DR_x = EDR_{q1} \cap EDR_{q2}$

$DW_x = EDW_{q1} \cap EDW_{q2}$

$L(q_3) \leftarrow L(x)$

Fusing join ($\_^f$). This join is similar to the restrictive join but is slightly less restrictive, because it keeps more owners in the new label. It calculates the label of the object containing the result of the operation as follows.

If $L(x) = L(q_1) \_^f L(q_2)$ then:

$O_x = (O_{q1} \cup O_{q2}) \cap (JR_{q1} \cap JR_{q2})$

$\forall o_i \in O_x$ and $o_i \in O_{q1} \cap O_{q2}$

$R_{x,oi} = R_{q1,oi} \cap R_{q2,oi}$

$W_{x,oi} = W_{q1,oi} \cap W_{q2,oi}$

$TO_{x,oi} = TO_{q1,oi} \cap TO_{q2,oi}$

$DR_{x,oi} = DR_{q1,oi} \cap DR_{q2,oi}$

$DW_{x,oi} = DW_{q1,oi} \cap DW_{q2,oi}$

$L(q_3) \leftarrow L(x)$

Declassification join ($\_^d$). During the execution of a transaction, if a step cannot be carried out because a participant does not have access to an object, the whole transaction will be aborted. Allowing the operation to proceed by temporarily declassifying the object can solve this problem, but strict rules have to be introduced to avoid total loss of security.

The proposed solution is adding an intermediate subject to the effective reader set of the requested object, if the initiator of this transaction is already an effective reader of the object. For writing, we add the intermediate subject to the effective writers, if the originator is already an effective writer. Declassification can proceed only if all effective owners agree to it by nominating subjects in their declassification sets.

When reading is performed, the declassified information must not be saved or used later. That can be ensured by setting the label of the object receiving the declassified information to empty ($\varnothing$). In case of writing, the label of the written object should not become less restrictive as a result. When the declassification read or write is finished, the intermediate principal will be removed from the effective reader or writer set respectively.

4 Implementation

4.1 Dynamic label checking

The labels are evaluated run-time, i.e. flow control is implemented dynamically. Object access is via special information flow control modules that check the labels of objects against the requester's credentials; no direct access is allowed. The IFC

modules implement the rules and operations described above, and perform the necessary actions. The basic structure of the system is shown in Figure 1.

The system operates in conjunction with local policies that may be implemented statically. The integrated system provides uniform access to all objects. A web-based environment facilitates the deployment of IFC modules, they can be inserted between requesters and objects transparently.

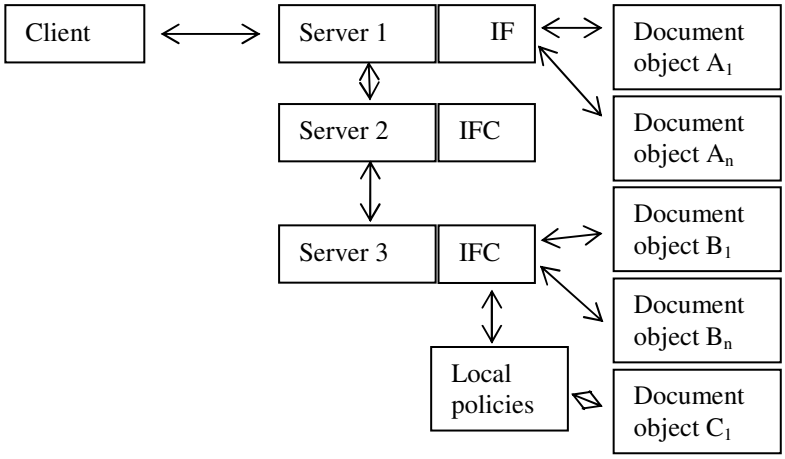

Fig. 1 - System Architecture

5 Related work

Several attempts have been made to secure documents in virtual enterprise environments. An early model suggested a federated process framework to coordinate access to shared data [1]. Cooperation on the process level enabled multiple acts, including iteration and re-attempting operations, to be conveniently performed on one hand, but resulted in a very complex system on the other.

Extending basic access control mechanisms across enterprises offered a simpler approach. An example of this is securing workflows with centrally controlled read access to documents [4]. The solution defines trusted virtual domains for individual workflows, which are under the control of a security kernel. The underlying security mechanism is encryption that provides durable protection, even in case of off-line access. A cornerstone of the method is encryption key management, but no solution is presented for that in the paper, although some notes refer to its complexity. The paper mentions that simple solutions such as key revocation or re-encryption of documents are unwieldy. More complex solutions, such a time-dependent keys or group keys are not considered in the paper, perhaps because of the computational load involved. The aims of the method are similar to ours in protecting objects (workflows) across different domains. However, our approach is more comprehensive because of introducing and handling object ownership and rights management as opposed to simple right enforcement, while we also deal with write operations not only with read.

A formal approach to the problem of authorizations in workflows for virtual enterprises was proposed in [8]. Their solution is more theoretical than that of [4], but still has the same constraints.

Object management on the virtual enterprise level requires a global approach; as an extension of local rights management may not be sufficient to address the complexities of the task. The web and its service-oriented paradigm offer a number of advantages, among them an established infrastructure with appropriate support. Such a solution utilizing role-based access control is described in [10]. It defines workspaces as online web environments shared by a cohesive set of actors, where access to information/data is controlled via services that support the business processes. Workspace membership helps the formation of communities of geographically dispersed actors, while their collaboration is controlled by the services that enforce security / confidentiality. The method complements our solution, as it provides a model for upper-level management of data sharing in a virtual enterprise, while it relies on the underlying mechanisms provided by the implementation platform to describe and enforce the security requirements.

Other high-level approaches include ontology-based knowledge sharing [2,3] that look at the assignment of user rights, and can be built on top of our model.

6 Conclusion

An information flow control model was proposed in this paper, to facilitate operations on and preserve confidentiality and integrity of objects. The model is tailored to environments where users at different sites with different access models need to work with shared, common objects.

The solution is based on security labels attached to objects. Owners of objects can nominate readers, writers of their objects, as well as express their consent to declassify the objects to specific subjects. Each owner also has a list of other owners of the object whom they trust.

We defined operations on labels, which can be associated with different procedures on the objects themselves, for example assignment and merging. Specific label operations also permit limited declassification of objects under controlled circumstances. The operations have been proven to preserve object security; however, space limitations do not allow the inclusion of proofs here.

References

1. Bae K, Kim J, Huh S. Federated process framework in a virtual enterprise using an object-oriented database and extensible markup language, Journal of Database Management, Jan 2003, http://www.accessmylibrary.com/coms2/summary_0286-22108914_ITM
2. Chen TY, Chen YM, Chu HC, Chen CC. Knowledge access control policy language model for virtual enterprises, IEEE International Conference on Industrial Engineering and Engineering Management, 2-4 Dec. 2007, 1903 – 1907
3. Chen TY. Knowledge sharing in virtual enterprises via an ontology-based access control approach, Computers in Industry, 2008, 59 502-519

4. Gasmi Y, Sadeghi A, Stewin P, Unger M, Winandy M, Husseiki R, Stüble C. Flexible and Secure Enterprise Rights Management Based on Trusted Virtual Domains, ACM STC'08, Fairfax, VA, 2008
5. Gray JW III. Toward a Mathematical Foundation for Information Flow Security. IEEE Symposium on Security and Privacy 1991: 21-35
6. McLean J. Security Models and Information Flow. IEEE Symposium on Security and Privacy. 1990; 180-189
7. Millen JK. Covert channels capacity. In Proceedings of the 1987 IEEE Computer Society Symposium on Security and Privacy, Oakland, CA, 1987.
8. Qing L, Huan Z. Research on Dynamic Authorization in Workflow of Virtual Enterprise, International Conference on Wireless Communications, Networking and Mobile Computing, WiCom 2007, 21-25 Sept. 2007, 6068 - 6070
9. Osborn S, Sandhu R, Munawer Q. "Configuring role-based access control to enforce mandatory and discretionary access control policies". ACM Transactions on Information and System Security (TISSEC). 2000; 3(2) 85-106
10. Rezgui Y. Role-based Service Oriented implementation of a Virtual Enterprise: a Case Study in the Construction Sector, Computers in Industry, 2007, 58 74-86
11. Tari Z, Bertok P, Simic D. A Dynamic Label Checking Approach for Information Flow Control in Web Services. International Journal of Web Services. 2006; 1:1-28
12. Wittbold JT, Johnson DM. Information flow in nondeterministic systems. In Proceedings of the 1990 IEEE Computer Society Symposium on Security and Privacy, Oakland, CA, 1990.