Traducement: A Model for Record Security

TOM WALCOTT and MATT BISHOP
University of California at Davis

Security models generally incorporate elements of both confidentiality and integrity. We examine a case where confidentiality is irrelevant to the process being modeled. In this case, integrity includes not only the authentication of origin and the lack of unauthorized changes to a document, but also the acceptance of all parties that the document is complete, signed by all parties, and cannot be modified further. This is especially critical when the document is recorded, so that it is legally the agreement or statement of record, and any copies of the document have no legal force. We show that current security models do not capture the details of this process. We then present a new security model for this process. This model captures the recordation process, and augments, rather than supplants, existing models. Hence it can also be used with existing security models to describe other situations.

Categories and Subject Descriptors: D.2.0 [**Software Engineering**]: General—*Protection mechanisms*; D.4.6 [**Operating Systems**]: Security and Protection—*Access controls*; H.2.7 [**Database Management**]: Database Administration—*Security, integrity, and protection*; J.1 [**Computer Applications**]: Administrative Data Processing—*Government*; K.6.5 [**Management of Computing and Information Systems**]: Security and Protection—*Unauthorized access*

General Terms: Design, Management, Security, Theory

Additional Key Words and Phrases: Integrity, recordation, security policy, traducement

1. INTRODUCTION

Systems designed to provide some measure of security to government systems are the bread and butter of computer security. In fact, most of the models currently accepted by the security community are based around government or corporate needs, addressing criteria under which information flow is permissible. Some models also address changing information.

Government needs motivated the first security issues. Times of war necessitated private communications, and gave rise to the field of cryptography. Since then, people have used increasingly sophisticated techniques to ensure

This research was supported by an award from the Office of the Clerk-Recorder, Yolo County, California.

Authors' addresses: Tom Walcott and Matt Bishop, Department of Computer Science, University of California at Davis, 1 Shields Ave., Davis, CA 95616-8562. Tom Walcott is employed by RABA Technologies LLC; email: twalcott@raba.com.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or direct commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 1515 Broadway, New York, NY 10036 USA, fax: +1 (212) 869-0481, or permissions@acm.org.
© 2004 ACM 1094-9224/04/1100-0576 $5.00

the confidentiality and integrity of important information. As these techniques proved useful, a wider audience, including corporations and private individuals, has adopted them. Over time, these techniques have given rise to models that formalize the systems and permit a clearer examination of the component needs.

This paper addresses the problem of modeling the security involved in recordation, which is the registering of legal documents with some authority designated to hold the documents. Our instance is the Yolo County Recorder's Office, the entity in Yolo County, California, US, for recording legal documents, pursuant to California law.

The Yolo County Recorder's Office "records births, marriages, and deaths; issues marriage licenses; receives and records filings for fictitious business names, notary publics, and process servers; acts as a passport agency; and receives and records property transfer documents" [Oakley 2003]. These are documents of record that are also available to the public. The role of the recorder is to accept documents, verify they are complete, record them, and ultimately make them public. Recorded documents are dispositive in a court of law; unauthorized amending of records could be disastrous because (for example) the recorded instrument for property ownership is the document that is filed in the recorder's office.

The submission of a document may require one signatory, as in the case of a tax lien, or multiple signatories, as in the case of a marriage certificate. Until the recorder's office has recorded the document, any signatory to the document may revoke the transaction. A signed document may not be altered—even if the document has not yet been recorded. This ensures that all signatories approve of the document as it was signed. This is an unusual requirement because it prohibits alteration yet allows revocation.

The verification process for documents consists of determining that the document is in the correct form and is completely filled out, then adding some additional information such as a document number. The recorder's office at no point changes the information in the document; they have only the right to append specific information to a document. Once a document has been recorded, it is indelible. Not even the recorder's office may change it.

From this outline, we can identify the characteristics that make this problem interesting from a security standpoint. Enumerating them clearly permits us to refer back to specific points when assessing the suitability of other security models in addressing recordation. In temporal order, the requirements are:

(1) a signed document cannot be altered (although new signatures may be appended);
(2) a document may require multiple signatures;
(3) a document submitted to the recorder's office may be revoked by any signatory until the document is recorded, but is no longer eligible for additional signatures;
(4) the recorder may only append information to the document (i.e., sign it); and

(5) if the document is recorded, it becomes a public record immutable to all parties.

A proposed solution must address these requirements. A good model may have certain additional characteristics. For example, a signed document may be incompletely filled out. Requirement (1) stipulates that a signed document may not be altered. Yet this requirement could be fulfilled in a system where alteration of a document automatically revoked all signatures. Other desirable characteristics include an enumerated list of document authors (for reasons of accountability), and dates of document creation and document recordation.

We first examine current security models in light of these needs and argue that they do not adequately address the problem. We will then describe Traducement, a proposed model that satisfies the above requirements. We will demonstrate that it models the recordation process, and show other processes that it can model. We conclude with future directions for research.

2. EXISTING POLICY MODELS

To demonstrate the need for a new policy model, we review several major computer security models and show how each falls short of the above-mentioned recorder's goals. Although there are several discretionary access controls that could be applied to this problem, they are unsuitable by their very nature. Discretionary access controls rely upon users making appropriate security decisions. This assumes users are intimately familiar with the operation of the system, and will act in good faith. But history, and indeed the very existence of the recordation mechanism, provides ample evidence that neither of these assumptions is always true.

2.1 Bell–LaPadula Model and Biba Model

The Bell–LaPadula model [Bell and LaPadula 1975] was developed for an environment with rigidly defined security clearances and classification labels. The recordation process has a completely different environment. Consider requirement (1); once one party has signed the document, the contents must become immutable, yet readable by all. Therefore, it must be unclassified. Therefore, all other cosigners must operate at the unclassified level, in order to sign (write to) the document. But this elides the distinction between writing and appending! The problem persists once the document is submitted to the recorder, who must be able to append to the document without altering the existing content. Finally, the notion encapsulated by requirement (5) poses yet more problems. An immutable public record would have to exist at a clearance level below unclassified to be a clearance level so low that no one can write to it.

One way to make the Bell–LaPadula model applicable would be to make recorders declassification authorities. Yet this would impose a considerable burden upon the recorder as they adjusted the classification of documents at essentially every step of the process, and the overhead would be unacceptable.

The Biba model [Biba 1977], an integrity mode, is the mathematical dual of the Bell–LaPadula model. It suffers from the same problems.

2.2 Lipner's Integrity Matrix Model

Lipner's model [Lipner 1982] makes extensive use of separation of duty to assure confidentiality and integrity. Security compartments are defined to permit particular types of object manipulation. This is most often useful in software development environments, establishing a distinction between development, compilation, and testing of programs. Initially, this seems to be a very productive approach for the recorder problem. By providing separate compartments for recorders and signatories, we can limit the privileges of each party over a given document. The signing process then becomes a means to transfer a document from security environments.

Unfortunately, automated compartment shifts are a security hazard. The purpose of separation of duty is to complicate ready movement between compartments, and ensure that only data that is in some way trusted is permitted to move across. Furthermore, depending upon implementation, different users looking at the same set of objects may find themselves with different controls over those objects. In this case, we must alter the privileges of various users in an automated fashion based upon the stage of the document's completion. The resulting mechanism is excessively complex, and would require very sophisticated analysis for proofs of sufficiency.

2.3 Clark–Wilson Model

The Clark–Wilson model [Clark and Wilson 1987] defines a *valid* state of a system to be one conforming to predefined data integrity constraints. The model uses the notion of transformation procedures, which move the system from one valid state to another. The model applies the data integrity constraints to all data crucial to the operation of the organization. Yet we have no notion of what comprises a valid state in a document to be recorded. There is no stricture on what type of data can be submitted to the recorder. This is a decision that ultimately must be made by a human. Further, as forms change over time, this would require regular updates to the model.

Some aspects of the model, however, are useful. Defining a series of transformation procedures on the basis of signatures and approvals is a meaningful concept, but it does not depend upon the data so much as the humans submitting the data. Unfortunately, these transformations require maintaining a great deal of state regarding signatories, initial signatories, and recorders. While the Clark–Wilson model could be modified to support these abstract transformation procedures, it would require considerable effort in both design and implementation.

2.4 Chinese Wall Model

The Chinese Wall model [Brewer and Nash 1989] was developed to prevent access to data sets from different sources of similar types. The model creates multiple Conflict of Interest classes that contain all organizations competing in the area of the Conflict of Interest class. The model prohibits a subject from ever seeing data from two organizations in the same Conflict of Interest class. This prevents data proliferation between two organizations.

Both the Chinese Wall model and our model change the properties of entities over time, in response to the actions of various subjects. The Chinese Wall model changes properties of the subjects, in the sense that the set of objects that a subject can access changes over time. Our model changes properties of the objects, in the sense that whether the objects can be altered changes over time. However, the details of the models differ greatly, and unlike the Chinese Wall model, our model considers *all* data public.

The purpose of the recorder's office is to *encourage* data proliferation while providing some assurances of the quality of the data. So the constraints of the Chinese Wall model through which it prevents conflicts of interest contradict the *raison d'etre* for the recorder's office. Furthermore, defining conflict of interest classes is non-trivial in this environment, and would limit reviewing, revoking, and voicing approval of a candidate record. For these reasons, the Chinese Wall model is unsuitable for use in the recorder's venue.

2.5 Originator-Controlled Models

The ORCON (or ORGCON) set of models [Graubert 1989] control dissemination of information by identifying a set of subjects as the originators of the information. These originators control to whom the information can be disseminated. This poses two problems. First, we have a flexible originator set that shall always include the recorder. Second, our primary concern is not the dissemination of information, but rather control over the legally approved content. Information from the recorder's office is intended for broad dissemination without limitation on recipients.

2.6 Clinical Information Systems Security Model

Of all the models surveyed, the Clinical Information Systems Security (CISS) model [Anderson 1996] is closest to meeting the spirit of the requirements. The CISS model protects medical data from unauthorized disclosure and modification. In that model, a patient's medical data is specifically attributed to that patient, and all those who have the ability to read or modify that data are closely monitored. An adaptation of this model to the environment of the recorder's office would ensure that all transformations enacted upon records were logged and available to some auditor. This auditor could then determine whether a particular record satisfied the requirements by assessing modifications after the fact.

However, the protections for patient records are too restrictive to use in this context. Regular modifications of the lists of entities permitted access to an object would be necessary during the initial phases prior to recordation. Further, the goal of a recorded document is to disseminate information freely. This goal is antithetical to the goals of the CISS model.

Finally and most importantly, the auditing mechanism to detect problems could only operate after the problem has occurred. Once a document has been recorded and cannot be modified by any party, it is too late to review the audit logs and determine whether there has been an unauthorized modification. That

document, once recorded, is a legal record; it is too late to cry foul. If, however, the document history is audited prior to recordation, there exists a race condition whereby the document could be altered between auditing and recording. Both of these alternatives are unacceptable.

3. STATEMENT OF THE MODEL

The models above concern themselves with the goals of confidentiality and separation of duty. In their conventional form, these goals are secondary to the recorder's office. Our model, Traducement, deliberately avoids most conventional notions of access controls. Any implementation of this model would need to consider access controls for confidentiality and separation of duty independently. While this demands a greater investment of time initially, Traducement does not interfere with traditional access controls. As a result, Traducement can interact seamlessly with all of the aforementioned models.

Traducement emphasizes two relatively unusual aspects of legal documentation. These aspects operate in a manner distinct from traditional confidentiality and integrity mechanisms. The notion of publishing a document, or relinquishing the right to further modify it (and possibly delivering it to a larger community), is fundamental to the legal environment. Equally important is the notion of associating the authors of a document with the document.

We distinguish between two types of signatories for documents. There are authors, who have contributed in some way to the document to be filed (even if only by signing the document). The recorder's office does not contribute to the document contents per se, but attests to the completion of the form. The second form of signature, then, converts the document into an official record. This leads to three goals.

Goal 1. The set of authors (contributors) remains associated with a document throughout the lifetime of that document. Any alteration of the document must void all existing signatures.

Goal 2. Subjects must be able to sign documents, and the act of signing a document must in no way invalidate existing signatures.

Goal 3. The signature of the recorder's office (in the capacity of recorder) must serve to publish the document.

The model recognizes two types of entities: objects and users. The *administrative authority* is a distinguished set of users with special rights as defined below (similar to the UNIX superuser or the Windows Administrator). A set of *rules* governs the manipulation of data.

First, we define terms to capture the notion of authorship and review. In the following definitions, u is a user, o is an object, and $o(x)$ is attribute x of object o.

Definition 1. The *author_set* attribute of an object specifies the set of users who have written to that object. No author can ever be removed from an object's

author set. All users in an object's author set are considered to have "creative rights" over that object.

Definition 2. The *signer_set* attribute of an object specifies the set of users who have approved of that object and its contents. Any user who can read an object can voluntarily add herself to this set. Individual users can be removed from the signer set only by the administrative authority.

A set of rules defines how these attributes are initialized and changed. Essentially, the above are initialized on creation, and the author set is modified on alteration. In the following statements, x is an entity before the rule is applied, and x' is that same entity after the rule is applied.

Creation Rule. When a user u creates an object o, the object is indelibly stamped with its creation time. The *author_set* contains one member, the creator. The *signer_set* is empty. In symbols:

$$o'(author_set) = \{u\}$$
$$o'(signer_set) = \oslash$$

Creating a file does not express approval of its contents. For example, a lawyer might be hired to write out a lien. In the process of writing out that lien, some fields may be incomplete and others may contain approximated values. The lawyer did not approve of the contents of this file; she merely generated it as a basis for her work. This is the rationale behind leaving the *signer_set* empty.

Alteration Rule. When a user u alters an object o, the user is added to the *author_set*. The *signer_set* is cleared. In symbols:

$$o'(author_set) = \{u\} \cup o(author_set)$$
$$o'(signer_set) = \oslash$$

Clearing the *signer_set* seems antithetical, because a user who changes a file might be expected to approve of the altered file. This is untrue for two reasons. First, other signers may not have had an opportunity to approve the modification. Second, if the user uses a data gathering program to obtain the data, she may not know what data is entered. In this case, she cannot approve of what she does not know. Hence the signing is distinct from modification. If the user wants to add herself to the *signer_set* after the modification, she can apply the next rule to do so.

Signature Rule. When a user u signs an object o, the user is added to the *signer_set*. The *author_set* is unchanged. In symbols:

$$o'(author_set) = o(author_set)$$
$$o'(signer_set) = \{u\} \cup o(signer_set)$$

As an example, suppose Peter writes a draft of a document. His draft has the *author_set* {Peter}, and an empty *signer_set*.

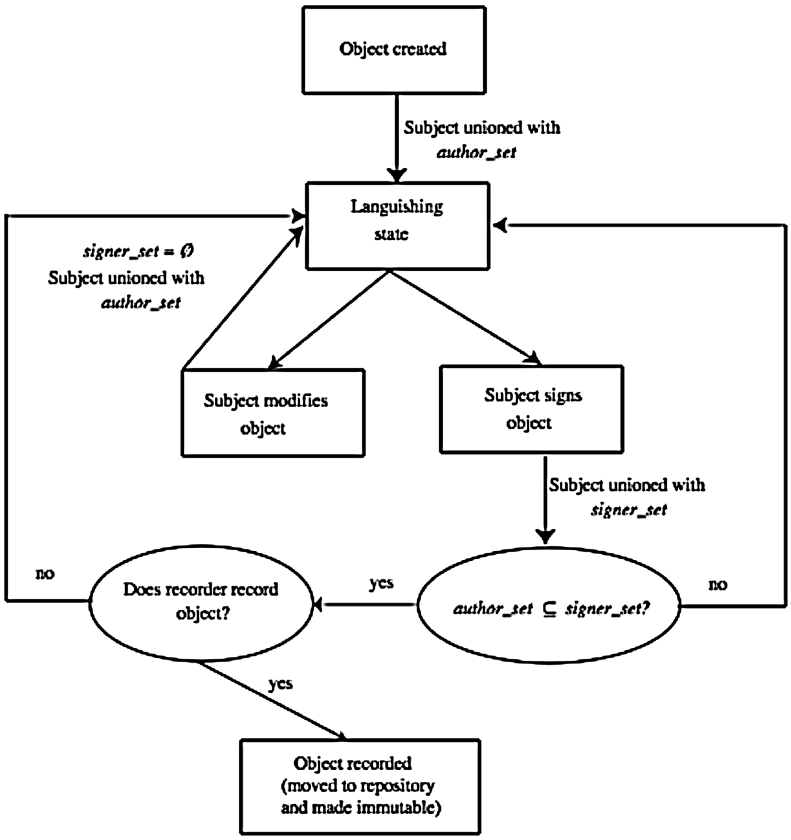

Fig. 1. State flow diagram of the model. There are two possible paths from the Languishing state, depending on the subject's action. The results of the action are shown along the connecting arrows. The ellipses represent decision points; the arrow to be followed is shown by the answer to the question.

Peter asks Paul, his lawyer, to review his work. Paul reads it and finds it appropriate, so he signs it. Peter's paper now has *author_set* {Peter} and *signer_set* {Paul}.

Peter then asks Mary for comments. Mary makes several changes to the document. After she does so, the document has the *author_set* {Peter, Mary} and an empty *signer_set*. The *signer_set* must be cleared because Paul has not agreed to any of Mary's changes. Paul can reread the document and sign it again, if he wishes. This reflects the association of a signer with the document as seen. If an author makes additional modifications after the signing, the signatories need to review the document again to verify the changes are unobjectionable.

Figure 1 is a state flow diagram of the model that captures the above rules.

It is important to note that this document has not yet been published. It is still pending publication, and has not at any time been approved by the recorder. An example of publication will follow a few more definitions.

Kate now copies the document for her own use. As she does not alter the document, none of the attributes will change. The copy rule captures this:

Copy Rule. When a user u creates a copy O of an object o, the *author_set* and *signer_set* are copied. In symbols:

$$O'(author_set) = o(author_set)$$
$$O'(signer_set) = o(signer_set)$$

We now establish a basic result.

PROPOSITION 1. *A user is in the* signer_set *of an object if and only if the document has not been modified since the user was added to the* signer_set.

PROOF. ($\Rightarrow$) Let o be an object with a user $u \in o(signer_set)$. Consider which rules could alter o. The creation and alteration rules set $o(signer_set) = \emptyset$. The signature and copy rules do not alter the object o. A straightforward induction shows that, if $u \in o(signer_set)$, neither the creation nor the alteration rules were used.

($\Leftarrow$) Now assume the object o has not been modified since u was added to $o(signer_set)$. Then the signature or copy rule was applied. The signature rule adds elements to $o(signer_set)$. The copy rule copies $o(signer_set)$ from the original element. Neither deletes elements from the *signer_set*. Again, a straightforward induction shows that, if the document has not been modified since the user was added to the *signer_set*, then that user is still in the *signer_set*. □

This result means that the signers of the document have access to all changes in that instance of the document. No signer can credibly claim that the document was altered after he signs it.

Assume that the current state of a system satisfies:

Precondition 1. Each document in the system has an *author_set* list identifying all users who created or modified that document.

Precondition 2. Each document in the system has a *signer_set* list identifying all users who approve that document.

We now show that systems that satisfy these conditions and upon which the rules are applied continue to satisfy those conditions.

THEOREM 1. *If a system satisfies Preconditions* 1 *and* 2*, then the system satisfies preconditions* 1 *and* 2 *after any sequence of applications of the creation, alteration, signature, and copy rules.*

PROOF. Let a system meet Preconditions 1 and 2. Call this state s_0. Consider the state s_1 reached by applying one of the rules.

1. If the create rule is applied, a new document is created. The associated *author_set* is the singleton containing the creator only (satisfying Precondition 1), and the associated *signer_set* is empty (satisfying Precondition 2).

2. If the alteration rule is applied, the document is altered. The identifier of the user making the alteration is added to the *author_set*, so the *author_set* contains the new alterer, the previous alterers (by assumption), and no one else. This satisfies Precondition 1. The *signer_set* is cleared. This means that those who have approved of the prior content are not held to have approved of the altered document. This meets Precondition 2.
3. If the signature rule is applied, the document is not altered. Hence the *author_set* is not changed, and Proposition 1 is satisfied. The user applying the signature rule approves of the document, and the signature rule adds her signature to the *signer_set*. The document remains unchanged from its last modification. This satisfies Precondition 2.
4. If the copy rule is applied, a new instance of the document is created. As the copy did not alter the contents of the file (merely duplicated it), the user is not added to the *author_set*. This meets Proposition 1. As the signers of the document approved of the *content* and the content is unchanged, they remain associated with the contents of the document. Hence the *signer_set* is copied, meeting Precondition 2.

Both Preconditions hold in state s_1. A straightforward induction argument completes the proof. □

If one defines a system meeting Preconditions 1 and 2 as "secure," this theorem is analogous to the Basic Security Theorem of the Bell–LaPadula model, in that it states systems beginning in a secure state and using these transition rules will always remain in a secure state.

Our next theorem characterizes systems in terms of transitions.

THEOREM 2. *Let R be a rule, s be a state of a system, and s' be the state obtained by applying R to s. Let the system in state s satisfy Preconditions* 1 *and* 2*, and let O and O' be the set of objects in states s and s', respectively. Then:*

1. *If there is an object o' such that $o' \notin O$, $o' \in O'$, $O' = O \cup \{o'\}$, $o'(author_set) = \{u\}$ for some subject u, and $o'(signer_set) = \emptyset$, then s' satisfies Preconditions* 1 *and* 2.
2. *If there is an object $o \in O$ such that $o'(author_set) = \{u\} \cup o(author_set)$, and $o'(signer_set) = \emptyset$, then s' satisfies Preconditions* 1 *and* 2.
3. *If there is an object $o \in O$ such that $o'(author_set) = o(author_set)$ and $o'(signer_set) = \{u\} \cup o(signer_set)$, then s' satisfies Preconditions* 1 *and* 2.
4. *If there is an object $x' \notin O$ but $x' \in O'$, and there is an object $o \in O$ such that $x'(author_set) = o(author_set)$ and $x'(signer_set) = o(signer_set)$, then s' satisfies Preconditions* 1 *and* 2.

PROOF. Consider the first claim. As s satisfies Preconditions 1 and 2, for each $o \in O$, $o(author_set)$ identifies all users who created or modified o, and $o(signer_set)$ identifies all users who approve that object. As $o' \notin O$, but $o' \in O', o'$ is created. Let u be the subject that created it. As $o'(author_set) = \{u\}$, $o'(author_set)$ contains the user who created o'. Thus, for all $x' \in O'$, $x'(author_set)$ identifies all users who created or modified x'. Further, as o'

has just been created, no one has yet approved it. So $o'(signer_set) = \oslash$, for all $x' \in O'$, $x'(signer_set)$ identifies all users who approved it. But this means that s' satisfies Preconditions 1 and 2, as claimed.

The other three cases are proven similarly. □

The model raises several issues about naming. In particular, individuals from different counties often collaborate. The model must allow this, while preserving whatever basic level of confidentiality the collaborators desired. The different recording offices may have different security policies in place. The collaborators therefore must enforce the most conservative elements of some set of policies.

The government has internal hierarchies. The County of Yolo is an organizational unit of the State of California, which in turn is an organizational unit of the United States of America. Anna Smith, a resident of Yolo County, must be distinguished from her collaborator Anna Smith, a resident of Orange County. These two must be identified unambiguously. One approach is to use the X.509 [ITU 1993] Distinguished Names. Then a document can have authors from multiple hierarchies and administrative domains.

This leads to the following rule for naming users.

Domain Rule. The authors contained in the author group shall be given unique names.

Under most circumstances, the "unique name" should be a scoped name that reflects the administrative domain in which the user resides. This is one advantage of the X.509 naming scheme. The semantics of Distinguished Names can reflect hierarchical arrangements easily.

Externally, different domains may have hierarchical relationships with other domains. For example, Yolo County may be recording a tax lien on property in Orange County. In that case, the higher-level entity is the State of California. If such relationships are important, the authors can be named appropriately as described above. Note that a single user may have multiple names reflecting different roles, as discussed in RFC 1422 [Kent 1993].

Interacting counties logically leads to the notion of multiple domains. These domains need to interact with each other in order to move data. This requires the different domains to establish trust relationships with each other. While new counties and townships are rarely introduced, that possibility cannot be discounted. Hence, this notion of trust must account for expanding networks. While the X.509 "web of trust" lacks the sort of hierarchical accountability favored in other models, that web is perfect for a model emphasizing individual accountability.

Definition 3. A *domain* is a collection of systems. Domains may host any number of inferior domains, called *subdomains*. The domain that contains a subdomain is called the subdomain's *parent domain*. Objects on a subdomain are referred to as a subdomain's object. Each domain has its own administrative authority.

Further, Theorems 1 and 2 hold as long as fully qualified signatures are used.

4. AUTHORSHIP INTEGRITY

A primary interest in the recorder's office is tracing accountability for particular documents. Falsifying records is illegal, yet both intentional and unintentional instances of it occur regularly. With documents of court record, any low-cost assurances that can be provided offer great potential benefits. Traducement supports this by tying authorship to documents and their contents.

Definition 4. An object is *recorded* when the object's author set is a subset of the signer set and the recorder's office executes a recordation transformation on the object. The recordation transformation affixes the signature of the recorder to the object. Each domain has a designated repository that stores a copy of every recorded object in that domain. Only the administrative authority of a domain, or of its ancestor domains, may add to the contents of repository.

Returning to our previous example of an object with *author_set* {Peter, Mary} and *signer_set* {Paul}, as soon as Peter and Mary have both signed the object, it is eligible for recordation. The recorder will have to review the object for completeness (a notion outside the scope of this model) and then execute the recordation transformation upon the object.

The administrative authority may *not* alter objects in the repository. It may only add objects to the repository. Further, multiple domains may share a single repository. In this case, the administrative authority may add only those objects in its domain (or any subdomain).

This permits easy revocation of objects that have not yet been signed by the recorder's office; the object need only be modified. All signatures on the object will then be removed, and the object cannot be recorded. This highlights the importance of having some additional confidentiality enforcement mechanism to limit access to the object prior to recording, additional integrity enforcement mechanisms to protect recorded documents in the repository, and otherwise prevent denial-of-service attacks that could preclude recordation of a document. Traducement does not supply these mechanisms, but does not preclude other traditional security models from being employed upon the objects.

5. EXAMPLE APPLICATION OF THE MODEL

An example walking through the life cycle of a document will illustrate how the requirements for recordation are satisfied. Recall the requirements of a solution to the recordation problem are:

1. A signed document cannot be altered (although new signatures may be appended);
2. A document may require multiple signatures;
3. A document submitted to the recorder's office may be revoked by any signatory until the document is recorded, but is no longer eligible for additional signatures;
4. The recorder may only append information to the document (i.e., sign it);
5. If the document is recorded, it becomes a public record immutable to all parties.

Consider a document created by Alice that must be signed by Alice and Bob. But Eve would prefer not to have this document recorded. By assumption, Eve is not the recorder. The role of recorder is a trusted position, and addressing a failure of trust at the recorder's level is outside the scope of this model. In the initial state, the document will have the following characteristics when created by Alice:

$$o'(author_set) = \{Alice\}$$
$$o'(signer_set) = \oslash$$

If Eve attempts to modify the document at this point, Eve would be added to the author set. This would attest to the fact that she has in some way modified the contents of the document.

Let us therefore assume that Alice signs the document. The signing transformation leaves the document in this state:

$$o'(author_set) = \{Alice\}$$
$$o'(signer_set) = \{Alice\}$$

Should Eve modify the document now, not only would Eve be added to the author set, but Alice's signature would be revoked.

Alice now passes the document to Bob. As this document requires two signatures, and Bob wishes to make certain that he is one of the signatories, he alters the document. This leaves the document in the state:

$$o'(author_set) = \{Alice, Bob\}$$
$$o'(signer_set) = \oslash$$

Again, if Eve modifies the document, Eve would be added to the author set.

Bob signs the document and sends it back to Alice. Possibly after some additional iteration, Alice and Bob are both content with the document, and have both signed it. It is in the following state:

$$o'(author_set) = \{Alice, Bob\}$$
$$o'(signer_set) = \{Alice, Bob\}$$

At any point in this process, up to the next step, either Alice or Bob may revoke the document simply by altering it. This would empty the signer set.

The document is now passed to the recorder. As the document meets the requirements of Definition 4 (the author set is a subset of the signer set), the recorder's office may execute a recordation transformation, making the document immutable. Eve cannot tamper with the document at that point.

It should be noted that Eve could at any time before recordation launch a denial-of-service attack. While this is indeed a problem, it is a problem that can be addressed through the use of traditional mechanisms (such as access control lists), and through retaining copies of documents before submission. Traducement allows for the use of these techniques.

Each of the recorder's requirements has been addressed. A signed document cannot be altered, because alteration revokes the signatures. The document in

this example demonstrated a requirement for multiple signatures. Any signer could have revoked the document by modifying it. The recorder appended additional information in the form of a signature; the signature could include a timestamp, name, date of submission, and cryptographic checksum, for example. Finally, once the document is recorded, it is added to a repository and made immutable.

6. CONCLUSION

Important discrepancies exist between current security policy models and the security needs of some government documents. This paper presents an additional security policy model that emphasizes those features and highlights the unusual characteristics. It captures the notion of publication and accountability for release approval as well as co-signatory accountability. Further, this model can be combined with other access control policy models. This would reflect the frequent interaction of documents of record with both corporate and government organizations. As both groups have specific needs regarding the security of their information, the collaborating organizations can combine both policies to satisfy requirements wherever possible.

The ability to conjoin this model with other, existing models is especially valuable in this context. Meeting this goal in a consistent, documented fashion raises the level of confidence placed in the official records. The properties and unique features of this model appear to make this goal eminently reachable.

The model does not discuss availability. While vital to any organization (not just public offices), a reasonable definition of what availability means in this model is elusive. The definition that seems most applicable is "quality of service," which has some clear security implications. However, different organizations may define an acceptable level of "quality of service" very differently. The recorder's office occasionally needs to transmit batches of data (e.g., tax liens) to companies with appropriate justifications and approvals. The need of those companies to gain timely information may be different from those of an individual. Rather than add a vague requirement similar to "the system shall meet those availability requirements defined by the administrative authority," we elected to remain silent while the community refines its understanding of availability.

Despite this silence, the model makes one implicit statement about the need for availability. Built into the model is the notion of distributing recorded information; once data is recorded, it should be available to everyone that policy dictates should be permitted access. In other words, denial of access to recorded information constitutes a breach of availability. The model therefore requires, implicitly, that access should be assured to all information for which a requestor should have access.

Traducement addresses primary recording needs in a way that other security models do not, while preserving the desirable aspects of those other models. Traducement also addresses the potential for collaboration between entities, of automating the accountability process, and of publishing a document for dissemination.

ACKNOWLEDGMENTS

The authors thank Homer Briggs and Eli Chandler, both of whom contributed immensely to the original ideas that grew into the system outlined here. Without them, this paper would not have been possible. In addition, the authors thank Tony Bernhard, the former clerk-recorder of Yolo County, Freddie Oakley, the current clerk-recorder of Yolo County, and their staff, who provided valuable insight into the legalities and procedures of recordation.

REFERENCES

ANDERSON, R. 1996. A security policy model for clinical information systems. In *Proceedings of the 1996 IEEE Symposium on Security and Privacy*, Oakland, CA. 34–48.

BELL, D. AND LAPADULA, L. 1975. *Secure Computer System: Unified Exposition and Multics Interpretation*. Tech. Rep. MTR-2997 Rev. 1, The MITRE Corporation, Bedford, MA.

BIBA, K. 1977. *Integrity Considerations for Secure Computer Systems*. Tech. Rep. MTR-3153, The MITRE Corporation, Bedford, MA.

BREWER, D. AND NASH, M. 1989. The Chinese Wall security policy. In *Proceedings of the 1989 IEEE Symposium on Security and Privacy*, Oakland, CA. 206–214.

CLARK, D. AND WILSON, D. 1987. A comparison of commercial and military security policies. In *Proceedings of the 1987 IEEE Symposium on Security and Privacy*, Oakland, CA. 184–194.

GRAUBERT, R. 1989. On the need for a third form of access control. In *Proceedings of the Twelfth National Computer Security Conference*, Baltimore, MD. 296–304.

ITU. 1993. *Recommendation X.509—the Directory Authentication Framework*. International Telecommunications Union, Geneva, Switzerland.

KENT, S. 1993. *Privacy Enhancement for Internet Electronic Mail: Part II: Certificate-Based Key Management*. RFC 1422. Available from `ftp://ftp.rfc-editor.org/in-notes/rfc1422.txt`.

LIPNER, S. 1982. Non-discretionary controls for commercial applications. In *Proceedings of the 1982 IEEE Symposium on Privacy and Security*, Oakland, CA. 2–10.

OAKLEY. 2003. *Yolo County Clerk-Recorder*. Available at `http://www.yolorecorder.org`.

Received June 2003; revised June 2004; accepted August 2004